sbu

Spartacus sotto il cofano

Come funziona davvero il meta-framework Angular per storefront su SAP Commerce Cloud. Ogni affermazione rimanda a file e righe del codice sorgente.

SPARTACUS 221121.22.0 · ANGULAR 21.2 · SHA 5B7A1C2B
51 CAPITOLI · 12 APPENDICI · MATTEO GIUDITTA · NOV 2026

COLOPHON

Spartacus sotto il cofano

AUTORE	Matteo Giuditta · sbu.dev
EDIZIONE	Novembre 2026
CODICE ANALIZZATO	Spartacus 221121.22.0 allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1`; Angular 21.2.23, NgRx 21, RxJS 7.8.
RIFERIMENTI	Ogni riferimento `percorso:Lx-Ly` è un link: apre il file su GitHub, alle righe citate e allo SHA analizzato. I riferimenti con prefisso `angular:` aprono il repository di Angular al tag v21.2.23.
DIAGRAMMI	I diagrammi troppo grandi per la colonna sono tavole a tutta pagina in fondo al capitolo; nel testo resta il rimando.
IMPAGINAZIONE	Direzione Quaderno di sbu.dev: carta, inchiostro, reticolo e un solo quadretto colorato. Caratteri Bricolage Grotesque, Inter e IBM Plex Mono.
SORGENTE	github.com/mgiuditta/spartacus, branch `claude/spartacus-book`.

SBU · SPARTACUS SOTTO IL COFANO · 63 CAPITOLI

Indice

Introduzione 7

Parte I · Angular sotto il cofano 11

01 **Dependency Injection in Angular e in Spartacus** 12
Token, injector e provider. Perché un provideConfig vince sui default, come si sostituisce un adapter, perché i multi provider dei moduli lazy richiedono UnifiedInjector.

02 **NgModule, standalone e bootstrap dell'applicazione** 44
bootstrapApplication contro bootstrapModule: la funzione interna comune e l'appiattimento dei provider dei moduli.

03 **Lazy loading: router e moduli caricati a runtime** 82
Code splitting e nuovo injector a runtime. loadChildren/loadComponent di Angular, e perché Spartacus non usa il lazy loading del router.

04 **Change detection, Zone, zoneless e signals** 113
Come Angular riallinea il DOM allo stato, il ruolo di Zone e del modo zoneless, la stabilità per SSR, i signals.

05 **Il Router di Angular** 147
La pipeline URL → UrlTree → recognize → guard → resolver → attivazione, e i suoi eventi.

06 **HttpClient e interceptor** 179
La catena di interceptor tra l'adapter e il fetch(), su cui Spartacus appoggia auth, site context ed errori.

07 **SSR, hydration e TransferState** 214
Render su Node, hydration nel browser, TransferState. Le scelte di Spartacus: CxCommonEngine e lo state NgRx trasferito.

08 **Schematics e builder** 247
Trasformazioni su Tree per ng add/ng update e builder Architect eseguiti da Nx.

Parte II · Spartacus sotto il cofano 279

10 **Bootstrap e ordine di inizializzazione** 280
Cosa deve sapere lo storefront prima di renderizzare (config, site context, rotte, auth) e come lo attende.

11 **Configurazione: provideConfig, default, merge, validator, initializer** 314
Chunk di config, deep merge, CONFIG_INITIALIZER, config unificata dei moduli lazy, validator.

12 **Feature module e lazy loading in Spartacus** 341
Il lazy loading guidato dal CMS e non dalle route: featureModules, chunk, facade.

13 **Site context e site theme** 369
Base site, lingua e valuta: da dove arrivano e come si propagano a router, HTTP e UI.

14 **Routing: route semantiche, URL, page guard, contesto** 401
Route semantiche configurabili e route wildcard CMS protetta da CmsPageGuard. Protected routes.

15 **CMS (core): pagine, slot, componenti, caricamento** 433
Struttura di pagina da PageContext, separazione struttura/dati, caricamento accorpato dei componenti, CmsService.

16 **CMS (storefront): rendering dinamico delle pagine** 465
Layout degli slot, componenti differiti, mapping cmsComponents, injector per componente, guard CMS.

17 **Outlet** 505
Punti nominati nei template per inserire o sostituire UI senza fork.

18 **Layout, breakpoint, launch dialog e accessibilità** 533
Il "guscio" dello storefront in core-libs/storefront/layout/: breakpoint, dialog, a11y, direction, header.

19 **State management: NgRx, loader/entity/process state** 571
Higher-order reducer, azioni base e selector per dati in caricamento, caricati o falliti, indicizzati per id.

20 **Facade, command & query, eventi** 598
Facade proxy lazy, command con strategie di concorrenza, query con invalidazione a eventi, EventService.

21 **Livello OCC: adapter, connector, converter, endpoint** 629
Adapter astratti e connector, endpoint configurabili, converter impilabili, scope di caricamento.

22 **Autenticazione: OAuth, token, interceptor, refresh** 662
Flow OAuth, conservazione del token, 401 → refresh → retry, token client, guard di auth.

23 **Utente, profilo, consensi** 697
Un dominio diviso in tre strati di età e stile architetturale diversi.

24 **Prodotto e ricerca** 733
Caricamento "a pezzi" del prodotto con gli scope, ricerca PLP e search box.

25 **Carrello** 772
@spartacus/cart, esposta solo tramite secondary entry point: active cart, saved cart, wish list, quick order, import/export.

26 **Checkout** 816
Dal carrello all'ordine: indirizzo, delivery mode, pagamento, revisione. Le tre sotto-feature della libreria.

27 **Ordini** 854
Place order, storia e dettaglio ordini, resi, ordini ricorrenti, reorder, allegati.

28 **B2B: organization** 894
Amministrazione aziendale, approvazioni, ordini dell'unità, estratto conto, registrazione B2B.

29 **ASM (Assisted Service Module)** 934
Il customer support agent che emula un cliente: ricerca, sessione, carrello, Customer 360.

30 **Product configurator** 972
Prodotti configurabili: pagina di configurazione, conflitti, prezzi, overview.

31 **Quote, customer ticketing, subscription billing** 1011
Tre feature lib di account con la stessa architettura a quattro strati, usata in modi diversi.

32 **Pickup in store, store finder, delivery date** 1051
"Dove e quando arriva la merce": data di consegna richiesta e stimata, store finder, ritiro in negozio.

33 **Tracking, SmartEdit, Qualtrics** 1090
Librerie che collegano lo storefront a sistemi esterni (GTM, AEP, SmartEdit, Qualtrics).

34 **Integration libs: CDC, CDP, CDS, segment-refs** 1135
Integrazioni con Customer Data Cloud, CDP, Context-Driven Services, segmenti.

35 **Integration libs: OPF, digital payments, OPPS** 1182
Open Payment Framework, Digital payments, Omnichannel Promotion Pricing.

36 **Integration libs: S/4, OMF, CPQ quote, punchout, EPD** 1223
Librerie per backend Commerce arricchiti da addon o sistemi SAP esterni.

37 **Internazionalizzazione (i18n)** 1273
Chiavi di traduzione a runtime su i18next, chunk, pipe di formattazione, istanza per richiesta in SSR.

38 **Stili e theming** 1301
@spartacus/styles, una libreria Sass distribuita come sorgente, e il theming.

39 **SSR ottimizzato** 1333
Cosa succede attorno al render SSR sotto traffico reale.

40 **SEO e PWA** 1367
Title, meta, canonical e JSON-LD anche in SSR. Supporto PWA.

41 **Errori HTTP, error handling, logger e global message** 1398
Due sottosistemi distinti: global message (UX) e error handling (osservabilità/SSR).

42 **Feature toggle e deprecazioni** 1432
Correzioni protette da flag con nome, per non rompere le estensioni a valle. Gestione delle deprecazioni.

43 **Schematics e migrazioni di Spartacus** 1460
@spartacus/schematics: installazione in un'app ng new e migrazioni ai breaking change.

44 **Strategia di test e tooling del monorepo** 1498
Tre runner di unit test, utility di test, E2E SSR e Cypress, validazioni statiche, tool interni.

45 **Componenti CMS e shared components di storefront** 1536
Componenti per i tipi CMS di base (dati, normalizzazione, link, a11y) e building block shared/.

46 **Utility, window e modelli di dominio** 1579
Gli "attrezzi" di @spartacus/core: ConverterService, resolveApplicable, backOff, normalizzazione degli errori.

47 **L'app demo storefrontapp** 1615
Il banco di prova che accende qualunque combinazione di librerie, guidato dai test e2e.

Parte III · Ricette 1646

50 **Ricette: configurazione, DI, interceptor, eventi** 1647
Override di config, sostituzione di servizi e facade, interceptor ed error handler, eventi custom.

51 **Ricette: componenti CMS, outlet, route e layout** 1689
Sostituire o creare componenti CMS, provideOutlet, route e layout.

52 **Ricette: OCC, adapter, converter, endpoint, meta resolver** 1730
Adapter e connector, normalizer e serializer, fields ed endpoint, PageMetaResolver.

53 **Ricette: feature lib lazy, carrello/checkout, stili, traduzioni** 1776
Una feature lazy propria, estensioni di carrello e checkout, stili senza fork, traduzioni.

54 **API pubblica vs interna, upgrade, anti-pattern, checklist upgrade-safe** 1830
La mia estensione sopravvivrà al prossimo ng update?

Appendici 1859

A01 **InjectionToken** 1860

A02 **Configurazione (*Config)** 1878

A03 **Componenti CMS (cmsComponents)** 1936

A04 **Endpoint OCC** 1972

A05 **Guard** 2017

A06 **HTTP interceptor** 2022

A07 **Eventi (CxEvent)** 2028

A08 **Outlet** 2050

A09 **Feature toggle** 2065

A10 **API deprecate (@deprecated)** 2085

A11 **Glossario** 2098

A12 **Indice dei simboli** 2121

SBU · SPARTACUS SOTTO IL COFANO · PRIMA DI COMINCIARE

Introduzione

Un libro tecnico, in italiano, che spiega **come funziona davvero** Spartacus, il meta-framework Angular per storefront e-commerce su SAP Commerce Cloud (backend OCC). Ogni affermazione rimanda a file e righe precise del codice sorgente.

Di cosa parla

Le sintesi qui sotto riassumono le sezioni "1. A cosa serve e perché esiste" dei capitoli.

- **Parte I — Angular sotto il cofano.** I meccanismi di Angular 21 su cui Spartacus si appoggia, letti nei sorgenti di Angular: dependency injection (token, injector, multi provider), bootstrap standalone e per NgModule, lazy loading (code splitting più un nuovo injector a runtime), change detection, Zone, zoneless e signals, la pipeline del Router (recognize, guard, resolver, attivazione), la catena di interceptor di `HttpClient`, SSR con hydration e TransferState, schematics e builder.
- **Parte II — Spartacus sotto il cofano.** Si parte dall'avvio: configurazione definitiva, site context, rotte completate a runtime, stato di autenticazione. Poi la configurazione a chunk fusa con deep merge e i feature module lazy guidati dal CMS. Seguono site context e routing semantico con la route wildcard protetta da `CmsPageGuard`, e il CMS, sia core (pagine, slot, componenti) sia rendering dinamico. Poi outlet, layout, NgRx con i suoi higher-order reducer, facade proxy, command/query ed eventi, il livello OCC (adapter, connector, converter, endpoint) e l'autenticazione OAuth.
- Un capitolo per ogni dominio: utente, prodotto e ricerca, carrello, checkout, ordini, B2B, ASM, configuratore, quote, ticketing e subscription, pickup e store finder, tracking e SmartEdit, e le integration lib (CDC, CDS, OPF, S/4...). Poi gli aspetti trasversali: i18n su i18next, stili Sass, SSR ottimizzato, SEO e PWA, errori e global message, feature toggle e deprecazioni, schematics e migrazioni, tooling di test, componenti CMS di storefront, utility e app demo.
- **Parte III — Ricette.** Personalizzazioni che importano solo simboli esportati dai `public_api.ts` dei pacchetti `@spartacus/*`: configurazione, DI, interceptor ed eventi. Poi componenti CMS, outlet, route e layout, e i dati OCC (adapter, converter, endpoint, meta resolver). Infine feature lazy, carrello e checkout, stili e traduzioni, e una guida per far sopravvivere le estensioni al prossimo `ng update`.
- **Appendici A01–A12.** Tabelle generate da script a partire dal codice: token, config, componenti CMS, endpoint OCC, guard, interceptor, eventi, outlet, feature toggle, API deprecate, glossario e indice dei simboli.

Codice e versioni analizzati

Fonte: 00-meta.md.

VOCE	VALORE
SHA analizzato	`5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (detached da `develop`)
Versione Spartacus	librerie `221121.22.0`, workspace root `2612.0.0` (vedi DECISIONS.md, D5/D7)
Angular / @angular/ssr	~21.2.23 (sorgenti: tag `v21.2.23`)

VOCE	VALORE
NgRx	~21.0.1
RxJS	~7.8.0
zone.js	^0.16.0
TypeScript	~5.9.3
Node	engines `^22.22.0`
Nx	^22.7.6
Test	Jasmine ~6.3 + Karma ~6.4.4; Vitest ^4.1.10

Tutti i riferimenti `file:Lx-Ly` valgono per lo SHA indicato sopra. I riferimenti con prefisso `angular:` puntano al repo Angular al tag `v21.2.23`. Quelli con prefisso `deps:` puntano ai pacchetti npm pubblicati (`@angular/ssr`, NgRx, angular-oauth2-oidc, rxjs, i18next), estratti in sola lettura.

Metodo

Il libro è stato prodotto da agenti, in fasi distinte:

1. **Inventario**: mappa del monorepo (INVENTORY.md). Ogni file sorgente è assegnato a un capitolo in COVERAGE.csv.
2. **Outline**: piano dei capitoli, con focus, file chiave e file assegnati (OUTLINE.md).
3. **Dossier → capitolo**: per ogni capitolo, un analista raccoglie i fatti verificati in un dossier (cartella `dossier/`). Uno scrittore ne ricava il capitolo seguendo un template fisso di 11 sezioni, dalla "1. A cosa serve" alla "11. Domande di autoverifica".
4. **Review avversariale**: un verificatore ricontrolla i riferimenti e le affermazioni sul codice (cartella `review/`).
5. **Correzione**: un correttore applica i rilievi. Poi gli script di controllo e l'assemblaggio finale.

Regole di verità (testo completo in tools/AGENT_RULES.md):

- ogni affermazione sul funzionamento porta un riferimento `percorso:Lx-Ly`, con numeri di riga verificati allo SHA;
- niente simboli, file, chiavi di config o comportamenti inventati: ciò che non è stato letto nel codice non viene scritto;
- la documentazione ufficiale e la memoria valgono solo come indizi, da confermare sul codice;
- ciò che non si può verificare è marcato ⚠️ `DA VERIFICARE`, con il motivo;
- gli snippet di codice "reale" sono copiati alla lettera, con path e righe.

Documenti di lavoro:

- DECISIONS.md: decisioni prese durante il lavoro;
- REPORT.md: rapporto finale (prodotto alla chiusura del lavoro);
- INVENTORY.md: inventario del monorepo;
- OUTLINE.md: piano dei capitoli;
- COVERAGE.csv: stato di copertura di ogni file sorgente.

Percorsi di lettura

1. "Sono nuovo sul progetto"

Obiettivo: capire come si avvia uno storefront e come una pagina arriva sullo schermo, dall'URL al componente.

1. 02 — NgModule, standalone e bootstrap: come nasce l'app.
2. 01 — Dependency Injection: il meccanismo su cui si regge tutto Spartacus.
3. 10 — Bootstrap e ordine di inizializzazione: cosa attende lo storefront prima del primo render.
4. 11 — Configurazione: da dove viene quasi ogni comportamento.
5. 12 — Feature module e lazy loading: perché le feature si caricano dal CMS e non dalle route.
6. 14 — Routing: route semantiche e route wildcard CMS.
7. 15 — CMS core e 16 — CMS rendering: dalla struttura di pagina ai componenti.
8. 19 — State management: NgRx e i suoi reducer generici.
9. 20 — Facade, command & query, eventi: l'API che i componenti usano davvero.
10. 21 — Livello OCC: come i dati arrivano dal backend.
11. Poi un dominio a scelta, per vedere i pattern applicati: 25 — Carrello, 24 — Prodotto e ricerca o 22 — Auth. Infine 47 — L'app demo, per vedere tutto montato insieme.

2. "Devo estendere Spartacus"

Obiettivo: conoscere gli extension point e usarli senza rompere gli upgrade.

1. I meccanismi: 11 — Configurazione, 12 — Feature module, 20 — Facade ed eventi e 21 — OCC. In ciascuno conviene leggere almeno la sezione "8. Extension point".
2. La UI: 16 — CMS rendering per i mapping dei componenti, 17 — Outlet per inserire UI nei punti nominati.
3. Le ricette, in ordine: 50 — config, DI, interceptor, eventi → 51 — CMS, outlet, layout → 52 — OCC e dati → 53 — feature lazy, carrello/checkout, stili, i18n → 54 — upgrade-safe. Il capitolo 54 è da leggere prima di consegnare.
4. Le appendici come catalogo degli extension point: A02 — Config, A03 — Componenti CMS, A04 — Endpoint OCC, A08 — Outlet. Prima di agganciarsi a un simbolo vale la pena controllare anche A10 — API deprecate.

3. "Devo fare debug"

Obiettivo: arrivare in fretta al punto del codice in cui un comportamento nasce. Ogni capitolo ha una sezione "10. Come fare debug": i link qui sotto portano direttamente lì.

1. 41 — Errori e global message: da dove viene un messaggio d'errore, e perché un render SSR fallisce.
2. 07 — SSR, hydration e TransferState e 39 — SSR ottimizzato: differenze server/browser, hydration, fallback CSR, timeout.
3. 05 — Router e 14 — Routing: perché un URL porta alla pagina sbagliata o a una 404.
4. 16 — CMS rendering e 15 — CMS core: un componente che non compare o compare vuoto.
5. 10 — Bootstrap e 11 — Configurazione: ordine di inizializzazione e valore effettivo della config.
6. 21 — OCC, 06 — HttpClient e interceptor e 22 — Auth: URL, header, converter, token e refresh.
7. 19 — State management: ispezionare lo stato NgRx di loader ed entity.

8. Le appendici come mappe di ricerca: A06 — HTTP interceptor per sapere chi tocca una richiesta, A07 — Eventi per sapere chi emette o ascolta un evento, A09 — Feature toggle per sapere quale flag cambia un comportamento.

PARTE I

Angular sotto il cofano



01 Dependency Injection in Angular e in Spartacus

02 NgModule, standalone e bootstrap dell'applicazione

03 Lazy loading: router e moduli caricati a runtime

04 Change detection, Zone, zoneless e signals

05 Il Router di Angular

06 HttpClient e interceptor

07 SSR, hydration e TransferState

08 Schematics e builder

01

Dependency Injection in Angular e in Spartacus

Token, injector e provider. Perché un `provideConfig` vince sui default, come si sostituisce un adapter, perché i multi provider dei moduli lazy richiedono `UnifiedInjector`.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Sorgenti Angular: `/tmp/angular` @ `v21.2.23` (prefisso `angular:`).

1. A cosa serve e perché esiste

La Dependency Injection (DI) è il meccanismo con cui Angular costruisce e distribuisce le istanze di servizi, valori e funzioni: chi ha bisogno di qualcosa chiede un **token**, e un **injector** decide quale **provider** usare per produrre il valore. Spartacus è costruito quasi interamente su questo meccanismo: la configurazione globale è un token DI, i normalizer OCC sono multi provider, le facade sono provider con factory che caricano moduli lazy, gli adapter verso il backend sono classi astratte sostituibili via DI.

Capire il DI "sotto il cofano" serve, in un progetto Spartacus, a rispondere a domande molto concrete:

- perché un mio `provideConfig(...)` vince sulla config di default di una libreria? (risposta: `configFactory` fonde `RootConfig` dopo `DefaultConfig`, core-libs/core/src/config/config-tokens.ts:L10-L12);
- perché un mio provider `{provide: ProductAdapter, useClass: ...}` sostituisce quello OCC? (risposta: per i provider non-multi l'ultimo `records.set` vince, angular:packages/core/src/di/r3_injector.ts:L513-L521);
- perché un normalizer registrato in un modulo lazy non compare con `inject(PRODUCT_NORMALIZER)` nel root, eppure il `ConverterService` lo usa? (risposta: un multi token si ferma al primo injector che ha il record, angular:packages/core/src/di/r3_injector.ts:L347-L371; Spartacus raccoglie i multi provider di tutti gli injector con `UnifiedInjector.getMulti`, core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93);
- perché `inject()` lancia `MISSING_INJECTION_CONTEXT` dentro un callback asincrono? (angular:packages/core/src/di/contextual.ts:L25-L27, angular:packages/core/src/di/injector_compatibility.ts:L97-L103).

Il capitolo è diviso in due metà: prima il motore di Angular 21 (injector, provider, token, contesto di iniezione, initializer), poi come Spartacus lo usa (config, `CONFIG_INITIALIZER`, converter, interceptor, `MODULE_INITIALIZER`, `UnifiedInjector`, override via DI). I dettagli completi della configurazione sono in 11 — Configurazione e quelli del lazy loading in 12 — Feature module e lazy loading.

2. Mappa dei file

PATH	RUOLO
angular:packages/core/src/di/injector.ts	`Injector` astratto, `Injector.NULL`, `Injector.create`
angular:packages/core/src/di/null_injector.ts	`NullInjector`: fine della catena, lancia "No provider found"
angular:packages/core/src/di/r3_injector.ts	`EnvironmentInjector`, `R3Injector` (record, `get`, `hydrate`, multi, scope, initializer), `providerToFactory`
angular:packages/core/src/di/create_injector.ts	`createInjector`, `createInjectorWithoutInjectorInstances`
angular:packages/core/src/di/scope.ts	`InjectorScope`, `INJECTOR_SCOPE`
angular:packages/core/src/di/injection_token.ts	`InjectionToken` con `providedIn`/`factory`
angular:packages/core/src/di/interface/provider.ts	tipi `Provider`, `StaticProvider`, `EnvironmentProviders`, `ModuleWithProviders`

PATH	RUOLO
angular:packages/core/src/di/interface/defs.ts	ɵɵdefineInjectable (forma di ɵprov)
angular:packages/core/src/di/provider_collection.ts	makeEnvironmentProviders, provideEnvironmentInitializer, importProvidersFrom, walkProviderTree, type guard
angular:packages/core/src/di/injector_compatibility.ts	inject, ɵɵinject, injectInjectorOnly, injectArgs, RetrievingInjector
angular:packages/core/src/di/contextual.ts	runInInjectionContext, isInInjectionContext, assertInInjectionContext
angular:packages/core/src/di/initializer_token.ts	ENVIRONMENT_INITIALIZER (deprecato)
angular:packages/core/src/render3/di.ts	NodeInjector, getOrCreateInjectable, bloom filter, fallback al module injector
angular:packages/core/src/render3/ng_module_ref.ts	NgModuleRef, NgModuleFactory, EnvironmentNgModuleRefAdapter, createEnvironmentInjector
angular:packages/core/src/application/application_init.ts	APP_INITIALIZER, provideAppInitializer, ApplicationInitStatus
angular:packages/core/src/application/create_application.ts	creazione dell'env injector in bootstrapApplication
angular:packages/core/src/platform/bootstrap.ts	ordine environment initializer → app initializer → root component
core-libs/core/src/config/config-tokens.ts	Config, DefaultConfig, RootConfig, ConfigChunk, DefaultConfigChunk
core-libs/core/src/config/config-providers.ts	provideConfig, provideConfigFactory, provideDefaultConfig, provideDefaultConfigFactory
core-libs/core/src/config/utils/deep-merge.ts	deepMerge, semantica del merge della config
core-libs/core/src/config/config.module.ts	ConfigModule (forRoot, withConfig, withConfigFactory)
core-libs/core/src/config/services/configuration.service.ts	ConfigurationService: config "ambientale" dei moduli lazy
core-libs/core/src/config/config-initializer/config-initializer.ts	ConfigInitializer, CONFIG_INITIALIZER, helper
core-libs/core/src/config/config-initializer/config-initializer.module.ts	APP_INITIALIZER + location initializer per la config asincrona
core-libs/core/src/config/config-initializer/config-initializer.service.ts	ConfigInitializerService (initialize, getStable)
core-libs/core/src/util/converter.service.ts	Converter, ConverterService
core-libs/core/src/util/combined-injector.ts	CombinedInjector (parent multiplo per moduli lazy)
core-libs/core/src/lazy-loading/tokens.ts	MODULE_INITIALIZER
core-libs/core/src/lazy-loading/lazy-loading.module.ts	MODULE_INITIALIZER eseguiti in fase eager via APP_INITIALIZER
core-libs/core/src/lazy-loading/lazy-modules.service.ts	creazione degli injector dei moduli lazy
core-libs/core/src/lazy-loading/unified-injector.ts	UnifiedInjector (root + injector lazy)

PATH	RUOLO
core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts	proxy delle facade lazy
core-libs/core/src/routing/routing.module.ts	override di LOCATION_INITIALIZED, provideAppInitializer
core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts	LOCATION_INITIALIZED_MULTI e helper
core-libs/core/src/base-core.module.ts	composizione dei forRoot() del core (ordine degli import)
projects/storefrontapp/src/app/app.config.ts	appConfig dell'app demo (importProvidersFrom(AppModule))

3. Come funziona passo per passo

3.1 La gerarchia degli injector

Angular 21 ha due alberi di injector che collaborano.

Albero "environment" (R3Injector).

1. In cima c'è il NullInjector, che non risolve nulla: se gli si chiede un token con THROW_IF_NOT_FOUND lancia un errore PROVIDER_NOT_FOUND ("No provider found for …", con error.name = 'ɵNotFound'), altrimenti restituisce il notFoundValue — angular:packages/core/src/di/null_injector.ts:L16-L28. Un singleton è creato pigramente da getNullInjector() — angular:packages/core/src/di/r3_injector.ts:L103-L110 — ed è il parent di default quando non ne viene passato uno — angular:packages/core/src/di/create_injector.ts:L55-L55, angular:packages/core/src/render3/ng_module_ref.ts:L164-L164.
2. Il **platform injector** è creato con Injector.create e il provider {provide: INJECTOR_SCOPE, useValue: 'platform'} — angular:packages/core/src/platform/platform.ts:L96-L105. Injector.create delega a createInjector, che costruisce un R3Injector e ne risolve gli initializer — angular:packages/core/src/di/injector.ts:L83-L118, angular:packages/core/src/di/create_injector.ts:L21-L35, angular:packages/core/src/di/create_injector.ts:L42-L55.
3. L'**environment injector applicativo** (il "root injector") in un avvio standalone è un EnvironmentNgModuleRefAdapter con parent il platform injector e runEnvironmentInitializers: false — angular:packages/core/src/application/create_application.ts:L64-L79. L'adapter crea un R3Injector con scope new Set(['environment']) — angular:packages/core/src/render3/ng_module_ref.ts:L151-L172. Lo scope 'root' arriva dal provider {provide: INJECTOR_SCOPE, useValue: 'root'} dentro BROWSER_MODULE_PROVIDERS, che bootstrapApplication antepone ai provider dell'utente — angular:packages/platform-browser/src/browser.ts:L259-L260, angular:packages/platform-browser/src/browser.ts:L168-L171. Il costruttore di R3Injector legge INJECTOR_SCOPE e aggiunge la stringa ai suoi scopes — angular:packages/core/src/di/r3_injector.ts:L236-L241.
4. Gli **injector dei moduli lazy** (Router loadChildren, oppure in Spartacus LazyModulesService) sono NgModuleRef il cui R3Injector ha scope solo 'environment' — angular:packages/core/src/render3/ng_module_ref.ts:L82-L95. NgModuleFactory.create(parentInjector) è semplicemente new NgModuleRef(this.moduleType, parentInjector, []) — angular:packages/core/src/render3/ng_module_ref.ts:L132-L134.
5. createEnvironmentInjector(providers, parent, debugName) è l'API pubblica per creare a mano un environment injector figlio; a differenza del bootstrap usa runEnvironmentInitializers: true — angular:packages/core/src/render3/ng_module_ref.ts:L193-L205.

`InjectorScope` è il tipo `'root' | 'platform' | 'environment'` e `INJECTOR_SCOPE` è il token (interno) che lo imposta — angular:packages/core/src/di/scope.ts:L11-L20. Il costruttore registra sempre `INJECTOR` → se stesso e, se lo scope contiene `'environment'`, anche `EnvironmentInjector` → se stesso — angular:packages/core/src/di/r3_injector.ts:L216-L234.

Albero degli elementi (NodeInjector). Ogni componente/direttiva ha un `NodeInjector` legato a `(tNode, lView)`, il cui `get` delega a `getOrCreateInjectable` — angular:packages/core/src/render3/di.ts:L893-L908. La ricerca:

1. consulta prima gli eventuali embedded view injector (se presenti e senza `Self`) — angular:packages/core/src/render3/di.ts:L465-L483;
2. risale i node injector usando un bloom filter a 256 slot per scartare in fretta i nodi che non possono contenere il token — angular:packages/core/src/render3/di.ts:L124-L136, angular:packages/core/src/render3/di.ts:L576-L621; i `viewProviders` sono visibili solo dal componente o risalendo dalla sua view — angular:packages/core/src/render3/di.ts:L637-L655; `Self` interrompe la risalita, `Host` la interrompe all'host element — angular:packages/core/src/render3/di.ts:L870-L877;
3. infine ricade sul "module injector" `lView[INJECTOR]` (l'environment injector) con `setInjectImplementation(undefined)`, così i servizi dell'environment non vedono lo scope dei componenti; `Self`/`Host` impediscono questo fallback — angular:packages/core/src/render3/di.ts:L407-L439, angular:packages/core/src/render3/di.ts:L491-L493.

I `providers` e `viewProviders` di `@Component` arrivano sull'element injector attraverso ɵɵProvidersFeature → `providersResolver` — angular:packages/core/src/render3/features/providers_feature.ts:L44-L56.

3.2 Come un `R3Injector` risolve un token

`R3Injector extends EnvironmentInjector` e tiene una `Map<ProviderToken, Record | null>`; il valore `null` significa "so già che qui non c'è" e evita ricerche ripetute — angular:packages/core/src/di/r3_injector.ts:L191-L197. Un `Record` contiene `factory`, `value` e `multi` — angular:packages/core/src/di/r3_injector.ts:L116-L120.

Passi di `R3Injector.get(token, notFoundValue, options)` — angular:packages/core/src/di/r3_injector.ts:L323-L422:

1. verifica che l'injector non sia distrutto (`assertNotDestroyed`, che lancia `INJECTOR_ALREADY_DESTROYED`) — angular:packages/core/src/di/r3_injector.ts:L328-L328, angular:packages/core/src/di/r3_injector.ts:L683-L690;
2. imposta l'**injection context**: `setCurrentInjector(this)` e `setInjectImplementation(undefined)` — angular:packages/core/src/di/r3_injector.ts:L341-L342. È questo che rende valido `inject()` dentro le factory;
3. se non c'è `SkipSelf`, cerca il record locale — angular:packages/core/src/di/r3_injector.ts:L344-L347;
4. se il record manca ma il token ha un `ɵprov` il cui `providedIn` è "in scope", crea il record al volo ("Pretend as if it was here all along"); altrimenti memorizza `null` — angular:packages/core/src/di/r3_injector.ts:L348-L367. La regola di scope: `'any'` è sempre in scope, una stringa deve essere fra gli `scopes`, un tipo NgModule deve essere fra gli `injectorDefTypes` — angular:packages/core/src/di/r3_injector.ts:L551-L561, angular:packages/core/src/di/r3_injector.ts:L243-L243;
5. se il record c'è, `hydrate` crea l'istanza al primo accesso: imposta il marker `CIRCULAR`, invoca la factory, lancia `cyclicDependencyError` se rientra sullo stesso token, e registra l'istanza per `ngOnDestroy` se ha l'hook — angular:packages/core/src/di/r3_injector.ts:L524-L549, angular:packages/core/src/di/r3_injector.ts:L86-L98;

6. altrimenti passa al `parent` (o al `NullInjector` se c'è `Self`); con `Optional` e `notFoundValue` di default usa `null` — angular:packages/core/src/di/r3_injector.ts:L374-L383;
7. in caso di `CYCLIC_DI_DEPENDENCY`/`PROVIDER_NOT_FOUND`, in dev mode ogni livello antepone il token al path di dipendenza e l'ultimo formatta il messaggio — angular:packages/core/src/di/r3_injector.ts:L384-L410; in produzione l'errore è rilanciato senza messaggio (`new RuntimeError(errorCode, null)`) — angular:packages/core/src/di/r3_injector.ts:L411-L413.

Conseguenza fondamentale: **un'istanza è un singleton per injector**. Lo stesso token `providedIn: 'root'` ha un'unica istanza nell'environment injector root; un modulo lazy che ri-fornisce la classe nei propri `providers` ne crea un'altra.

`destroy()` marca l'injector come distrutto, chiama gli `ngOnDestroy` delle istanze e gli hook `onDestroy`, poi svuota i record — angular:packages/core/src/di/r3_injector.ts:L270-L295.

3.3 I provider e la loro trasformazione in factory

`Provider` è l'unione `TypeProvider | ValueProvider | ClassProvider | ConstructorProvider | ExistingProvider | FactoryProvider | any[]` — angular:packages/core/src/di/interface/provider.ts:L336-L343. La firma deprecata `Injector.create(providers, parent)` accetta solo `StaticProvider`, che esclude `TypeProvider`/`ClassProvider` e richiede `deps` espliciti per le classi (`StaticClassProvider`) — angular:packages/core/src/di/injector.ts:L83-L86, angular:packages/core/src/di/interface/provider.ts:L258-L264, angular:packages/core/src/di/interface/provider.ts:L59-L71; la firma attuale `Injector.create({providers, parent, name})` accetta invece `Array<Provider | StaticProvider>`, quindi anche classi senza `deps` — angular:packages/core/src/di/injector.ts:L100-L104. Tutte le forme oggetto (`ValueProvider`, `ClassProvider`, `ExistingProvider`, `FactoryProvider`, ...) hanno il flag opzionale `multi?: boolean` ("injector returns an array of instances") — angular:packages/core/src/di/interface/provider.ts:L40-L51; un `TypeProvider` (classe nuda) non può essere multi — angular:packages/core/src/di/r3_injector.ts:L497-L497.

Come vengono riconosciuti (type guard) — angular:packages/core/src/di/provider_collection.ts:L404-L427:

- `useValue`: per **presenza della chiave** (`USE_VALUE in value`), quindi anche `useValue: undefined` è un `ValueProvider`;
- `useExisting`, `useFactory`, `useClass`: per truthiness;
- `TypeProvider`: se il provider è una funzione (una classe).

Come diventano record — angular:packages/core/src/di/r3_injector.ts:L624-L632, angular:packages/core/src/di/r3_injector.ts:L639-L681:

FORMA	RECORD/FACTORY
`useValue`	record con valore già pronto, nessuna factory (`makeRecord(undefined, provider.useValue)`)
`useFactory` + `deps`	`provider.useFactory(...injectArgs(provider.deps || []))`
`useExisting`	`ɵɵinject(useExisting)` propagando solo il flag `Optional`: è un **alias**, non una nuova istanza
`useClass` con `deps`	`new classRef(...injectArgs(deps))`
`useClass` senza `deps` / `TypeProvider`	factory generata dal compilatore (`getFactoryDef`) o da `ɵprov`

`injectArgs` risolve ogni elemento di `deps`; se un elemento è un array, i suoi membri sono letti come flag (`new Optional()`, `Self`, ...) o `@Inject(token)` più il token — angular:packages/core/src/di/injector_compatibility.ts:L327-L362. È la forma che Spartacus usa in `deps: [ConfigInitializerService, [new Optional(), CONFIG_INITIALIZER]]` — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L51-L58.

Due errori tipici nascono qui: una classe senza decoratore con argomenti nel costruttore produce "Can't resolve all parameters" — angular:packages/core/src/di/r3_injector.ts:L598-L622; un `InjectionToken` senza `ɵprov` e senza provider produce "Token ... is missing a ɵprov definition." — angular:packages/core/src/di/r3_injector.ts:L580-L587.

3.4 Multi provider e precedenza

`processProvider` è il punto in cui si decide "chi vince" — angular:packages/core/src/di/r3_injector.ts:L473-L522:

1. il token è il provider stesso (per un `TypeProvider`) o `provider.provide` — angular:packages/core/src/di/r3_injector.ts:L477-L479;
2. se `multi: true`, crea (una sola volta per token) un record multi la cui factory è `injectArgs(multiRecord.multi)`; poi aggiunge il provider all'array `multi` e lo registra nei `records` con chiave **il provider stesso** — angular:packages/core/src/di/r3_injector.ts:L497-L512, angular:packages/core/src/di/r3_injector.ts:L521-L521. L'ordine di registrazione è quindi l'ordine dell'array finale;
3. se non è multi, `this.records.set(token, record)` **sovrascrive** il record precedente: l'ultimo provider registrato vince — angular:packages/core/src/di/r3_injector.ts:L513-L521;
4. mescolare multi e non-multi sullo stesso token lancia `throwMixedMultiProviderError()` solo in dev mode — angular:packages/core/src/di/r3_injector.ts:L501-L505, angular:packages/core/src/di/r3_injector.ts:L514-L519. In produzione un non-multi dopo un multi sovrascrive silenziosamente il record; un multi dopo un non-multi fa invece fallire la creazione dell'injector con un `TypeError`, perché `multiRecord.multi` è `undefined` — angular:packages/core/src/di/r3_injector.ts:L500-L512, angular:packages/core/src/di/r3_injector.ts:L692-L702.

L'ordine in cui i provider arrivano a `processProvider` dipende dall'appiattimento: `forEachSingleProvider` visita ricorsivamente array ed `EnvironmentProviders` preservando l'ordine — angular:packages/core/src/di/r3_injector.ts:L725-L738; per gli NgModule, `walkProviderTree` processa **prima gli** `imports` (ricorsivamente e con dedup), poi i provider dei `ModuleWithProviders` importati, poi il modulo stesso, infine i suoi `providers` — angular:packages/core/src/di/provider_collection.ts:L292-L354. La definizione di un NgModule importato più volte (imports, provider propri, istanziazione eager) è processata una sola volta — angular:packages/core/src/di/provider_collection.ts:L275-L276. I `providers` di ogni `ModuleWithProviders` vengono invece aggiunti a ogni import, anche se il modulo è già stato visto: `walkProviderTree` restituisce `true` per ogni `ModuleWithProviders` con `providers` e `processInjectorTypesWithProviders` li visita tutti — angular:packages/core/src/di/provider_collection.ts:L360-L362, angular:packages/core/src/di/provider_collection.ts:L195-L208. Più `ConfigModule.withConfig(...)` contribuiscono quindi tutti (è voluto), ma anche un `XModule.forRoot()` importato due volte registra due volte i suoi provider (per esempio un `APP_INITIALIZER` multi duplicato).

Infine il valore di un multi token è l'array dei soli provider registrati **in quell'injector**: la ricerca si ferma al primo injector che ha il record e non c'è unione con i parent — angular:packages/core/src/di/r3_injector.ts:L347-L371.

3.5 `InjectionToken` e `@Injectable`

`InjectionToken<T>` ha una descrizione usata solo per debug e un `ɵprov` opzionale — angular:packages/core/src/di/injection_token.ts:L65-L69, angular:packages/core/src/di/injection_token.ts:L82-L86. Se al costruttore si passano `options`, viene definito `ɵprov` con `providedIn: options.providedIn || 'root'`: `'root'` **è il default** — angular:packages/core/src/di/injection_token.ts:L110-L116. Le opzioni `providedIn: NgModule` e `'any'` sono deprecate — angular:packages/core/src/di/injection_token.ts:L47-L47, angular:packages/core/src/di/injection_token.ts:L71-L80. Due istanze diverse con la stessa descrizione sono token diversi — angular:packages/core/src/di/injection_token.ts:L26-L29.

`ɵɵdefineInjectable` produce `{token, providedIn, factory, value}` con `providedIn` fra `Type | 'root' | 'platform' | 'any' | 'environment' | null` — angular:packages/core/src/di/interface/defs.ts:L167-L178. Un token con `ɵprov` in scope viene "materializzato" alla prima richiesta (passo 4 di §3.2): non serve alcun provider esplicito.

3.6 `inject()`, injection context e `runInInjectionContext`

- `inject(token, options)` è `ɵɵinject(token, convertToBitFlags(options))` — angular:packages/core/src/di/injector_compatibility.ts:L293-L297; `convertToBitFlags` traduce `optional`/`host`/`self`/`skipSelf` nei flag interni — angular:packages/core/src/di/injector_compatibility.ts:L300-L315.
- `ɵɵinject` usa l'implementazione corrente (impostata da render3 per i NodeInjector) o `injectInjectorOnly` — angular:packages/core/src/di/injector_compatibility.ts:L140-L148.
- `injectInjectorOnly` lancia `MISSING_INJECTION_CONTEXT` se non c'è un injector corrente; altrimenti chiama `currentInjector.retrieve(...)` e, se opzionale e non trovato, restituisce `null` — angular:packages/core/src/di/injector_compatibility.ts:L93-L120.
- La JSDoc elenca i contesti validi: costruttore di una classe creata dal DI, field initializer, `useFactory`, `factory` di `InjectionToken`, frame dentro `runInInjectionContext`; chiamarlo in `ngOnInit` è un errore — angular:packages/core/src/di/injector_compatibility.ts:L226-L287.
- `runInInjectionContext(injector, fn)` imposta come injector corrente l'`R3Injector` (verificando che non sia distrutto) o un `RetrievingInjector` che adatta un `Injector` qualsiasi, esegue `fn` in modo **sincrono** e ripristina il contesto nel `finally` — angular:packages/core/src/di/contextual.ts:L37-L59, angular:packages/core/src/di/injector_compatibility.ts:L49-L68. La JSDoc avverte: niente `inject` dopo un `await` o in callback asincrone — angular:packages/core/src/di/contextual.ts:L25-L27.
- `isInInjectionContext()` / `assertInInjectionContext(fn)` permettono di verificarlo — angular:packages/core/src/di/contextual.ts:L64-L88.

3.7 `EnvironmentProviders`, `importProvidersFrom`, initializer

- `EnvironmentProviders` è un tipo "brandizzato" (`ɵbrand: 'EnvironmentProviders'`) accettato solo nella creazione di un `EnvironmentInjector` (e quindi anche negli NgModule), per impedirne l'uso in `@Component.providers` — angular:packages/core/src/di/interface/provider.ts:L345-L362. `makeEnvironmentProviders(providers)` restituisce solo `{ɵproviders: providers}` — angular:packages/core/src/di/provider_collection.ts:L40-L52.
- `importProvidersFrom(...sources)` restituisce `{ɵproviders: internalImportProvidersFrom(true, sources), ɵfromNgModule: true}` e in dev mode rifiuta i componenti standalone — angular:packages/core/src/di/provider_collection.ts:L144-L174.
- Per ogni NgModule incontrato `walkProviderTree` aggiunge **tre** provider: il modulo stesso (factory), `INJECTOR_DEF_TYPES` (multi) e un `ENVIRONMENT_INITIALIZER` `() => inject(defType)` che istanzia il modulo in modo **eager**; il commento precisa che ciò avviene dopo gli imports — angular:packages/core/src/di/provider_collection.ts:L324-L343.

- `ENVIRONMENT_INITIALIZER` è un multi token `ReadonlyArray<() => void>`, deprecato dalla v19 a favore di `provideEnvironmentInitializer`; le funzioni non sono attese — angular:packages/core/src/di/initializer_token.ts:L11-L26, angular:packages/core/src/di/provider_collection.ts:L81-L89. `resolveInjectorInitializers()` le legge con `{self: true}` e le esegue nell'injection context dell'injector — angular:packages/core/src/di/r3_injector.ts:L425-L455.
- `APP_INITIALIZER` è un `InjectionToken<ReadonlyArray<() => Observable | Promise | void>>` — angular:packages/core/src/application/application_init.ts:L150-L152; `provideAppInitializer(fn)` lo avvolge in `makeEnvironmentProviders` — angular:packages/core/src/application/application_init.ts:L165-L202; sia la JSDoc del token (`@deprecated from v19.0.0, use provideAppInitializer instead`, angular:packages/core/src/application/application_init.ts:L38-L38) sia quella della funzione lo dichiarano deprecato.
- `ApplicationInitStatus` (`providedIn: 'root'`) inietta `APP_INITIALIZER` opzionale (default `[]`) — angular:packages/core/src/application/application_init.ts:L209-L236. `runInitializers()` esegue **ogni** initializer con `runInInjectionContext`, raccoglie Promise e Observable (l'Observable deve **completare**) e risolve `donePromise` con `Promise.all` — angular:packages/core/src/application/application_init.ts:L239-L275.
- Nel `bootstrap`, dentro `ngZone.run`, prima si risolvono gli environment initializer e poi gli app initializer; il componente root parte solo a `donePromise` risolta — angular:packages/core/src/platform/bootstrap.ts:L87-L93, angular:packages/core/src/platform/bootstrap.ts:L137-L144.
- Con `initialNavigation: 'enabledBlocking'` il router registra un `provideAppInitializer` che attende `LOCATION_INITIALIZED` (default `Promise.resolve()`) prima della navigazione iniziale — angular:packages/router/src/provide_router.ts:L441-L469, angular:packages/router/src/router_module.ts:L240-L241.

3.8 Spartacus: la configurazione come grafo DI

Il cuore della config di Spartacus è in `config-tokens.ts`:

1. `Config` è una classe astratta vuota con `@Injectable({providedIn: 'root', useFactory: configFactory})` — core-libs/core/src/config/config-tokens.ts:L14-L21. `configFactory()` restituisce `deepMerge({}, inject(DefaultConfig), inject(RootConfig))`: **la config dell'app vince su quella di default** — core-libs/core/src/config/config-tokens.ts:L10-L12.
2. `DefaultConfig` e `RootConfig` sono `InjectionToken` `providedIn: 'root'` con factory che fondono rispettivamente tutti i `DefaultConfigChunk` e tutti i `ConfigChunk` (iniettati con `{optional: true}`, default `[]`) — core-libs/core/src/config/config-tokens.ts:L23-L47.
3. `ConfigChunk` e `DefaultConfigChunk` sono `InjectionToken<Config[]>` **senza factory**, da fornire come multi provider; la JSDoc dice di non usarli direttamente e che "all config provided in libraries should be provided as default config" — core-libs/core/src/config/config-tokens.ts:L49-L62.
4. Gli helper costruiscono i provider multi: `provideConfig(config, defaultConfig = false)` → `ValueProvider` su `ConfigChunk` (o `DefaultConfigChunk`), `provideConfigFactory` → `FactoryProvider`, `provideDefaultConfig` e `provideDefaultConfigFactory` → su `DefaultConfigChunk` — core-libs/core/src/config/config-providers.ts:L18-L79. Restituiscono `ValueProvider`/`FactoryProvider` normali, non `EnvironmentProviders` — core-libs/core/src/config/config-providers.ts:L7-L7.
5. `deepMerge` fonde ricorsivamente ogni valore oggetto non-array (non solo gli oggetti plain: anche le istanze di classi vengono copiate proprietà per proprietà in un `{}`, perché `isObject` controlla solo `typeof item === 'object' && !Array.isArray(item)`); `Date`, array e primitivi sono assegnati per sostituzione; `__proto__` e `constructor` sono ignorati; muta e restituisce `target` — core-libs/core/src/config/utils/deep-merge.ts:L7-L49.
6. Le sezioni tipizzate sono classi astratte `@Injectable({providedIn: 'root', useExisting: Config})` (es. `RoutingConfig`): iniettarle restituisce **lo stesso oggetto** `Config` — core-libs/core/src/routing/

configurable-routes/config/routing-config.ts:L22-L28. Il tipo `Config` è esteso per module augmentation — core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L47-L49. Il pattern `useExisting: Config` compare in 70 file (comando: `grep -rln "useExisting: Config" --include=*.ts core-libs feature-libs integration-libs | grep -v spec | wc -l`).

7. La regola ESLint `@nx/workspace-use-provide-default-config` segnala ogni `provideConfig(...)` nelle librerie: lì va usato `provideDefaultConfig()` — tools/eslint-rules/rules/use-provide-default-config.ts:L22-L48. `ConfigModule.withConfig` e `ConfigModule.forRoot` usano `provideConfig` (disabilitando `@nx/workspace-use-provide-default-config`); `withConfigFactory` usa `provideConfigFactory` (disabilitando `@nx/workspace-use-provide-default-config-factory`) — core-libs/core/src/config/config.module.ts:L28-L34, core-libs/core/src/config/config.module.ts:L48-L52, core-libs/core/src/config/config.module.ts:L60-L66.

Statistiche d'uso (comando per ciascun pattern: `grep -rF "<pattern>" --include=*.ts core-libs feature-libs integration-libs projects/storefrontapp/src | grep -v "\.spec\.ts" | wc -l`; i conteggi includono le definizioni degli helper): `provideDefaultConfig(` 397, `provideDefaultConfigFactory(` 73, `provideConfig(` 133, `provideConfigFactory(` 13, `provide: HTTP_INTERCEPTORS` 22, `MODULE_INITIALIZER` 21, `provideConfigValidator(` 6. Righe con `providedIn: 'root'` nelle librerie: 807 (`grep -rn "providedIn: 'root'" --include=*.ts core-libs feature-libs integration-libs | grep -v spec | wc -l`); righe con `new InjectionToken`: 220 (stesso comando con `"new InjectionToken"`).

3.9 Spartacus: config "ambientale" e moduli lazy

Un `provideDefaultConfig` dentro un modulo lazy finisce nell'injector di quel modulo, non nel root (§3.4). `ConfigurationService` (istanziato eagerly perché `ConfigModule` lo inietta nel costruttore — core-libs/core/src/config/config.module.ts:L13-L20) recupera questi chunk:

1. inietta `RootConfig`, `DefaultConfig`, `UnifiedInjector` e `Config`; espone `config` e `unifiedConfig$` — core-libs/core/src/config/services/configuration.service.ts:L42-L55;
2. `feedUnifiedConfig` combina con `zip` i `ConfigChunk` e `DefaultConfigChunk` ottenuti da `UnifiedInjector.get(token, [])` e salta la prima emissione (il root) — core-libs/core/src/config/services/configuration.service.ts:L57-L74;
3. `emitUnifiedConfig` ricalcola `deepMerge({}, defaultConfig, ambientDefaultConfig, ambientConfig, rootConfig)`, emette su `unifiedConfig$` e, salvo `disableConfigUpdates`, lo fonde (mutandolo) dentro `this.config` — core-libs/core/src/config/services/configuration.service.ts:L92-L106. La chiave `disableConfigUpdates` è deprecata e "disabled by default" — core-libs/core/src/features-config/config/features-config.ts:L101-L116; `isFeatureEnabled` restituisce `false` se non configurata — core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L65.

`UnifiedInjector` è il ponte fra il DI di Angular e i moduli lazy:

- `injectors$` emette il root injector e poi l'injector di ogni modulo lazy inizializzato — core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35;
- `get(token, notFoundValue)` interroga il root normalmente e gli altri con `{self: true}` — core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68;
- `getMulti(token)` accumula con `scan` gli array di tutti gli injector e lancia "Multi-providers mixed with single providers" se un valore non è un array — core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93.

Il parent degli injector lazy è scelto da `LazyModulesService.resolveModuleInstance`: root senza dipendenze, l'injector dell'unica dipendenza, o un `CombinedInjector(root, deps)` — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L108. `CombinedInjector` cerca prima a livello `self` in tutti gli injector comple-

mentari, poi a tutti i livelli, infine nel main injector; con `{self: true}` restituisce `notFoundValue` o lancia — core-libs/core/src/util/combined-injector.ts:L27-L72. I moduli di dipendenza sono creati una sola volta con parent il root — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L115-L129.

3.10 Spartacus: `CONFIG_INITIALIZER`, config asincrona prima del bootstrap

1. `ConfigInitializer` è `{scopes: string[]; configFactory: () => Promise<Config>}` e `CONFIG_INITIALIZER` è il multi token che li raccoglie — core-libs/core/src/config/config-initializer/config-initializer.ts:L15-L35. Helper: `provideConfigInitializer(Class)` (`ClassProvider` multi) e `provideConfigInitializerFactory(fn)` (`FactoryProvider` multi, la factory può restituire `null`) — core-libs/core/src/config/config-initializer/config-initializer.ts:L46-L67.
2. `ConfigInitializerModule.forRoot()` fornisce il guard `CONFIG_INITIALIZER_FORROOT_GUARD = true`, un `APP_INITIALIZER` multi con `deps: [ConfigInitializerService, [new Optional(), CONFIG_INITIALIZER]]` e un location initializer che "Hold on the initial navigation until the Spartacus configuration is stable" — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L41-L67. La factory dell'`APP_INITIALIZER` restituisce `() => configInitializer.initialize(initializers)` — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L24-L30.
3. `initialize()` ignora gli initializer `null`, lancia se uno non ha `scopes`, avvia tutte le `configFactory()` in parallelo e per ognuna fonde il risultato in `rootConfig` **e** in `config` (mutandoli), poi chiude i suoi scope — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L130-L175.
4. `isStable` è vero se il guard manca o se non ci sono scope in corso; `getStable(...scopes)` restituisce subito `of(this.config)` se stabile, altrimenti attende — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L42-L44, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70.
5. `RoutingModule.forRoot()` sostituisce il token Angular `LOCATION_INITIALIZED` con `locationInitializedFactory`, che invoca tutti i `LOCATION_INITIALIZED_MULTI` e restituisce `Promise.all` — core-libs/core/src/routing/routing.module.ts:L66-L71, core-libs/core/src/routing/routing.module.ts:L102-L105. L'`AppRoutingModule` di storefront configura `initialNavigation: 'enabledBlocking'` — core-libs/storefront/router/app-routing.module.ts:L13-L19. Così la navigazione iniziale attende la config stabile.
6. Esempi di `CONFIG_INITIALIZER`: `SiteContextModule.forRoot()` registra `initSiteContextConfig`, che restituisce `SiteContextConfigInitializer` solo se manca una config statica per `context.baseSite`, altrimenti `null` — core-libs/core/src/site-context/site-context.module.ts:L30-L61; `SiteContextConfigInitializer` ha `scopes = ['context']` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L24-L27.

3.11 Spartacus: altri multi provider e override via DI

- **Converter.** Un converter è un `Converter<S, T>` con `convert(source, target?)` — core-libs/core/src/util/converter.service.ts:L13-L30; è identificato da un `InjectionToken` (es. `PRODUCT_NORMALIZER`) — core-libs/core/src/product/connectors/product/converters.ts:L11-L13 — e fornito come multi provider `useExisting` — core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51. `ConverterService` svuota la cache a ogni nuovo injector lazy, legge i converter con `getLastValueSync(unifiedInjector.getMulti(token))` e li applica in sequenza con `reduce` — core-libs/core/src/util/converter.service.ts:L32-L65, core-libs/core/src/util/converter.service.ts:L130-L137. `getLastValueSync` sottoscrive, prende l'ultimo valore sincrono e annulla — core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23.
- **Interceptor HTTP.** Registrati come multi provider `useExisting` su `HTTP_INTERCEPTORS` — core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L12-L23; l'app demo li abilita con

`provideHttpClient(withFetch(), withInterceptorsFromDi())` — projects/storefrontapp/src/app/app.config.ts:L26-L35. `ErrorHandlingModule.forRoot()` è importato per primo in `BaseCoreModule` proprio perché l'ordine di registrazione determina l'ordine nell'array — core-libs/core/src/base-core.module.ts:L28-L28, angular:packages/core/src/di/r3_injector.ts:L512-L512. Dettagli in 06 — HttpClient e interceptor.

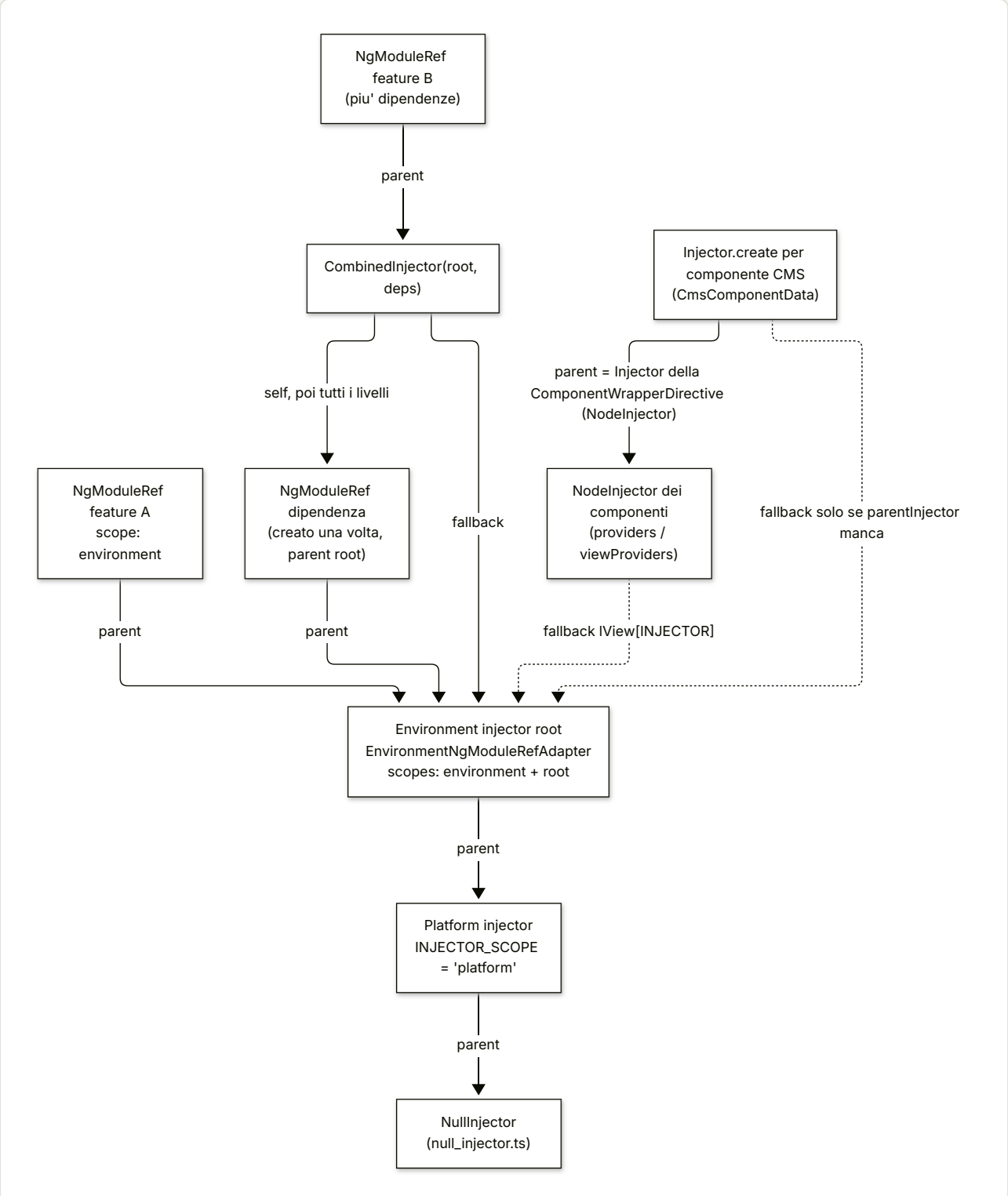
- **Adapter.** Classi astratte (es. `ProductAdapter`) legate all'implementazione OCC con `{provide: ProductAdapter, useClass: OccProductAdapter}` — core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L11, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L34-L37: un provider successivo sullo stesso token le sostituisce (§3.4).
- **Facade lazy.** Le facade sono classi astratte `@Injectable({providedIn: 'root', useFactory: () => facadeFactory({...})})` — feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L18; `facadeFactory` chiama `inject(FacadeFactoryService)` e va quindi usata in un injection context — core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24. Il modulo lazy ridefinisce la facade nel proprio injector con `useExisting: ActiveCartService` e il proxy la ottiene con `moduleRef.injector.get(facadeClass)` — feature-libs/cart/base/core/facade/facade-providers.ts:L27-L30, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52.
- `MODULE_INITIALIZER`. "In a way, an APP_INITIALIZER for lazy loaded modules" — core-libs/core/src/lazy-loading/tokens.ts:L9-L20. Se il modulo è eager, `LazyLoadingModule.forRoot()` li esegue in un `APP_INITIALIZER` — core-libs/core/src/lazy-loading/lazy-loading.module.ts:L18-L52; se è lazy, `runModuleInitializersForModule` li legge con `{self: true}` e li esegue — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L175. Esempio: `AsmCoreModule` — feature-libs/asm/core/asm-core.module.ts:L22-L35.
- **Injector "a mano".** `CmsInjectorService.getInjector` crea con `Injector.create` un injector figlio per ogni componente CMS, con `CmsComponentData` e i `providers` del mapping CMS — core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50.
- **Multi token con default.** `BEFORE_CMS_PAGE_GUARD` ha `providedIn: 'root', factory: () => []`, quindi vale `[]` finché nessuno fornisce provider — core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L22-L24.

3.12 Il bootstrap dell'app demo, visto dal DI

`main.ts` chiama `bootstrapApplication(AppComponent, appConfig)` — projects/storefrontapp/src/main.ts:L17-L17; `appConfig.providers` contiene i provider funzionali di Angular e `importProvidersFrom(AppModule)` — projects/storefrontapp/src/app/app.config.ts:L26-L35. L'env injector riceve, in ordine, i provider interni, `BROWSER_MODULE_PROVIDERS` e poi quelli di `appConfig` — angular:packages/core/src/application/create_application.ts:L66-L71, angular:packages/platform-browser/src/browser.ts:L171-L171. `AppModule` importa `SpartacusModule` e mette nei propri `providers` un `EnvironmentProviders` (`privateProviders`, costruito con `makeEnvironmentProviders` e contenente un `provideConfig` per il backend OCC) — projects/storefrontapp/src/app/app.module.ts:L15-L25, projects/storefrontapp/src/app/private/private.providers.ts:L19-L30. Per la regola di §3.4, questi `providers` sono processati dopo tutti gli import del modulo. L'ordine completo di avvio è nel capitolo 10.

4. Diagrammi

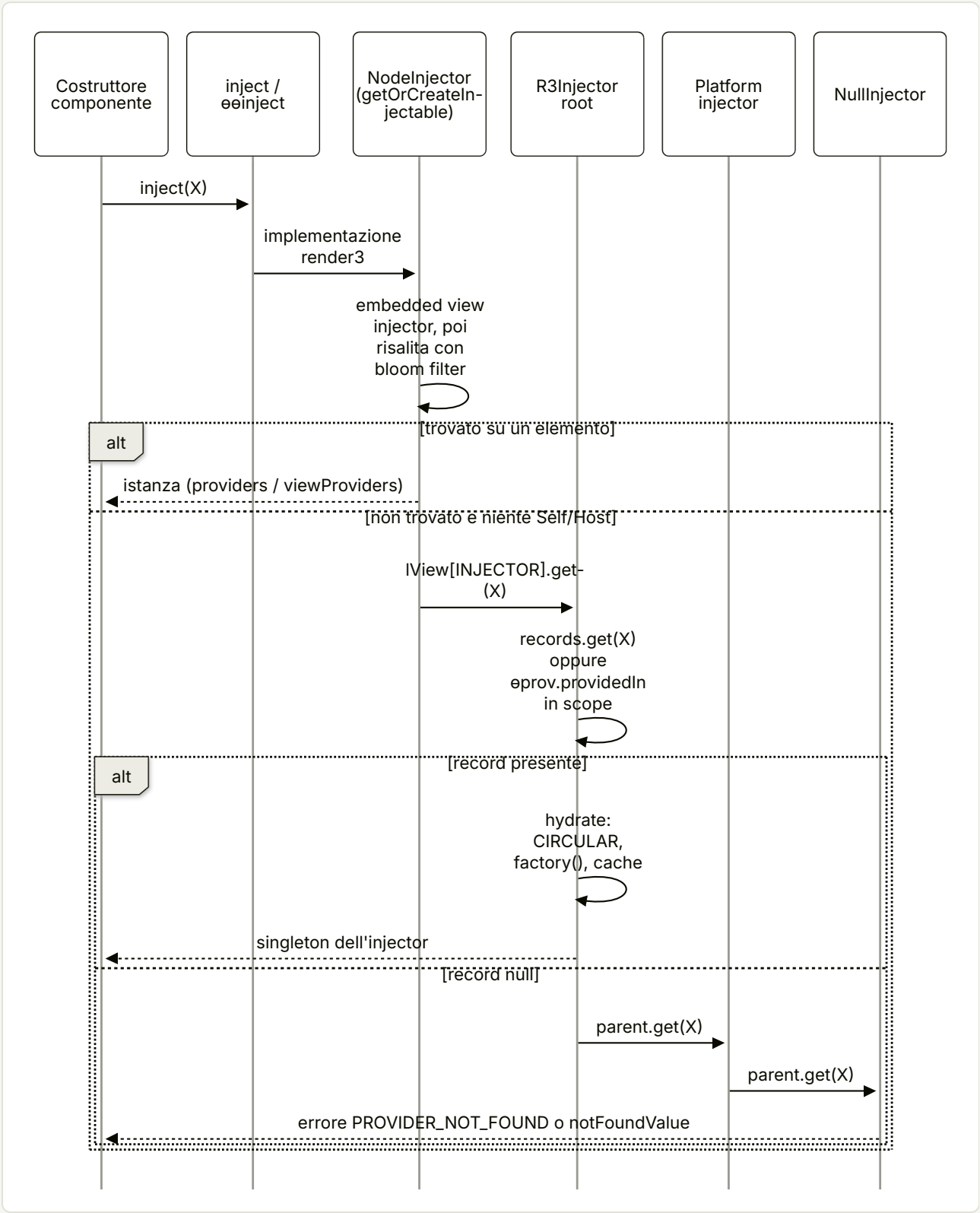
4.1 Gerarchia degli injector in un'app Spartacus



Riferimenti: `NullInjector` e parent di default (`angular:packages/core/src/di/r3_injector.ts:L103-L110`), platform (`angular:packages/core/src/platform/platform.ts:L96-L105`), root (`angular:packages/core/src/application/create_application.ts:L72-L79`, `angular:packages/platform-browser/src/browser.ts:L259-`

L260), moduli lazy e CombinedInjector (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L85-L94, core-libs/core/src/util/combined-injector.ts:L53-L70), dipendenze (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L115-L129), fallback dei NodeInjector (angular:packages/core/src/render3/di.ts:L418-L436), injector CMS (core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L35-L49; il parent è l'Injector iniettato nella ComponentWrapperDirective, core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L59-L59, core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L94-L98; il fallback al root parentInjector ?? this.injector è a core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L48-L48).

4.2 Risoluzione di `inject(X)` da un componente



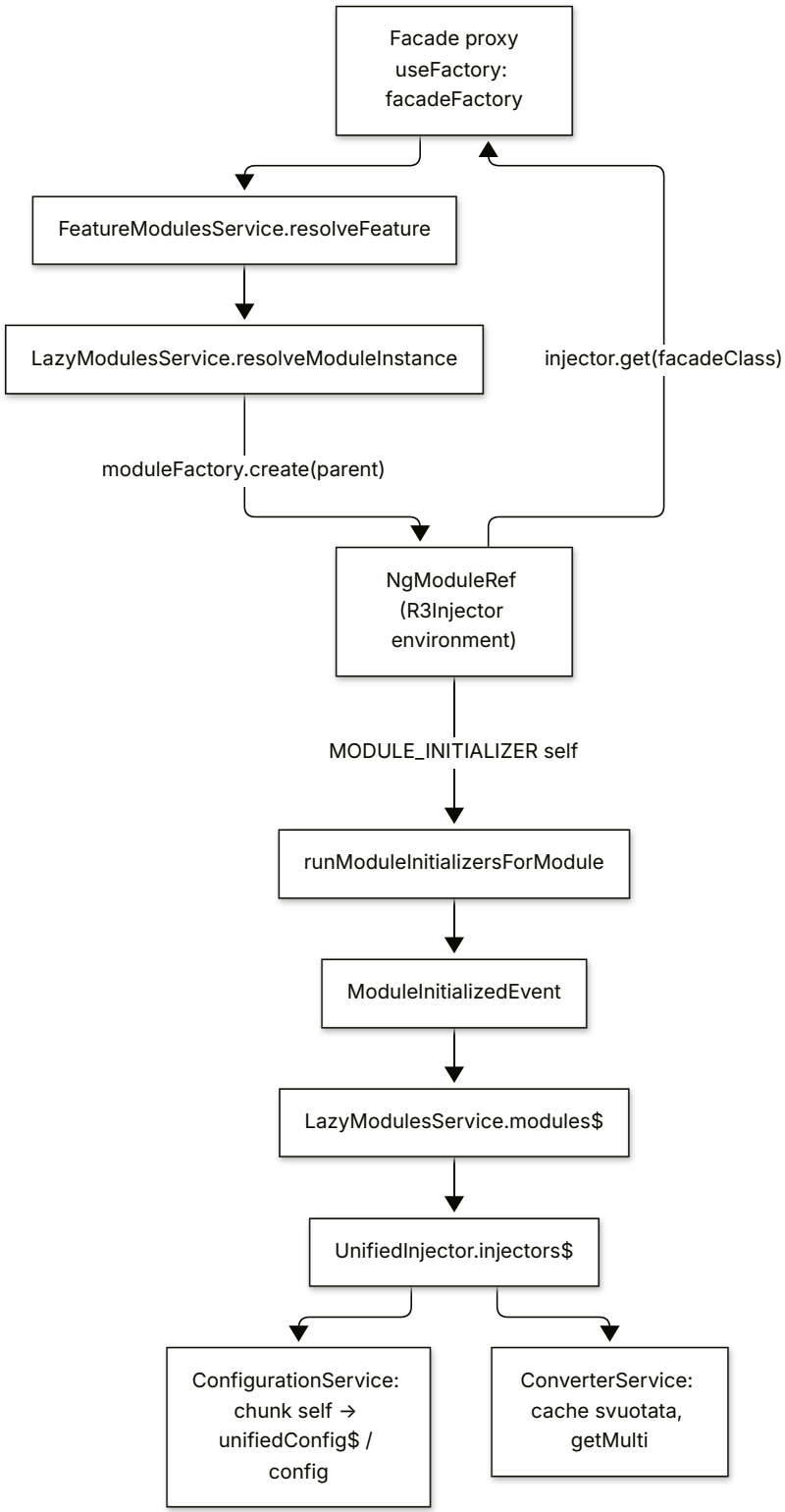
Riferimenti: angular:packages/core/src/di/injector_compatibility.ts:L140-L148, angular:packages/core/src/render3/di.ts:L457-L494, angular:packages/core/src/render3/di.ts:L407-L439, angular:packages/core/src/di/r3_injector.ts:L344-L383, angular:packages/core/src/di/r3_injector.ts:L524-L549, angular:packages/core/src/di/null_injector.ts:L16-L28.

4.3 Costruzione di `Config` e config asincrona al bootstrap

■ **TAVOLA 01.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: angular:packages/core/src/platform/bootstrap.ts:L87-L93, angular:packages/core/src/di/provider_collection.ts:L339-L343, core-libs/core/src/config/config.module.ts:L13-L20, core-libs/core/src/config/config-tokens.ts:L10-L47, angular:packages/core/src/application/application_init.ts:L239-L275, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L24-L67, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L130-L175, angular:packages/router/src/provide_router.ts:L441-L469, core-libs/core/src/routing/routing.module.ts:L66-L71, angular:packages/core/src/platform/bootstrap.ts:L137-L144.

4.4 Propagazione DI quando si carica una feature lazy



Riferimenti: `feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L18`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52`, `core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73`, `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L107`, `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L175`, `core-libs/core/src/lazy-loading/`

lazy-modules.service.ts:L47-L68, core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35, core-libs/core/src/config/services/configuration.service.ts:L57-L106, core-libs/core/src/util/converter.service.ts:L32-L65.

---

5. Estratti di codice reale

5.1 Il punto in cui "l'ultimo vince" e i multi si accumulano

angular:packages/core/src/di/r3_injector.ts:L497-L521

```ts
if (!isTypeProvider(provider) && provider.multi === true) {
  // If the provider indicates that it's a multi-provider, process it specially.
  // First check whether it's been defined already.
  let multiRecord = this.records.get(token);
  if (multiRecord) {
    // It has. Throw a nice error if
    if (ngDevMode && multiRecord.multi === undefined) {
      throwMixedMultiProviderError();
    }
  } else {
    multiRecord = makeRecord(undefined, NOT_YET, true);
    multiRecord.factory = () => injectArgs(multiRecord!.multi!);
    this.records.set(token, multiRecord);
  }
  token = provider;
  multiRecord.multi!.push(provider);
} else {
  if (ngDevMode) {
    const existing = this.records.get(token);
    if (existing && existing.multi !== undefined) {
      throwMixedMultiProviderError();
    }
  }
}
this.records.set(token, record);
```

Da notare: per un multi provider `token = provider` prima di `records.set`, quindi il record è indicizzato dal provider (che a sua volta viene risolto come "token" da `injectArgs`); per un non-multi `records.set(token, record)` rimpiazza silenziosamente.

5.2 Lookup locale, `providedIn` e risalita al parent

angular:packages/core/src/di/r3_injector.ts:L344-L383

```ts
  // Check for the SkipSelf flag.
  if (!(flags & InternalInjectFlags.SkipSelf)) {
    // SkipSelf isn't set, check if the record belongs to this injector.
    let record: Record<T> | undefined | null = this.records.get(token);
    if (record === undefined) {
      // No record, but maybe the token is scoped to this injector. Look for an injectable
      // def with a scope matching this injector.
      const def = couldBeInjectableType(token) && getInjectableDef(token);
      if (def && this.injectableDefInScope(def)) {
        // Found an injectable def and it's scoped to this injector. Pretend as if it was here
        // all along.
        // ...
        record = makeRecord(injectableDefOrInjectorDefFactory(token), NOT_YET);
      } else {
        record = null;
      }
      this.records.set(token, record);
    }
    // If a record was found, get the instance for it and return it.
    if (record != null /* NOT null || undefined */) {
      return this.hydrate(token, record, flags);
    }
  }

  // Select the next injector based on the Self flag - if self is set, the next injector is
  // the NullInjector, otherwise it's the parent.
  const nextInjector = !(flags & InternalInjectFlags.Self) ? this.parent : getNullInjector();
  // ...
  return nextInjector.get(token, notFoundValue);
```

5.3 useFactory, useExisting, useClass

angular:packages/core/src/di/r3_injector.ts:L653-L678

```ts
  if (isValueProvider(provider)) {
    factory = () => resolveForwardRef(provider.useValue);
  } else if (isFactoryProvider(provider)) {
    factory = () => provider.useFactory(...injectArgs(provider.deps || []));
  } else if (isExistingProvider(provider)) {
    factory = (_, flags) =>
      ɵɵinject(
        resolveForwardRef(provider.useExisting),
        flags !== undefined && flags & InternalInjectFlags.Optional
          ? InternalInjectFlags.Optional
          : undefined,
      );
  } else {
    const classRef = resolveForwardRef(
```

```
        provider &&
          ((provider as StaticClassProvider | ClassProvider).useClass || provider.provide),
      );
      // ...
      if (hasDeps(provider)) {
        factory = () => new classRef(...injectArgs(provider.deps));
      } else {
        return getFactoryDef(classRef) || injectableDefOrInjectorDefFactory(classRef);
      }
    }
```

5.4 `InjectionToken`: `'root'` è il default

angular:packages/core/src/di/injection_token.ts:L110-L116

TS

```
    } else if (options !== undefined) {
      this.ɵprov = ɵɵdefineInjectable({
        token: this,
        providedIn: options.providedIn || 'root',
        factory: options.factory,
      });
    }
```

5.5 `runInInjectionContext`

angular:packages/core/src/di/contextual.ts:L37-L59

TS

```
export function runInInjectionContext<ReturnT>(injector: Injector, fn: () => ReturnT): ReturnT {
  let internalInjector: PrimitivesInjector;
  if (injector instanceof R3Injector) {
    assertNotDestroyed(injector);
    internalInjector = injector;
  } else {
    internalInjector = new RetrievingInjector(injector);
  }
  // ...
  const prevInjector = setCurrentInjector(internalInjector);
  const previousInjectImplementation = setInjectImplementation(undefined);
  try {
    return fn();
  } finally {
    setCurrentInjector(prevInjector);
    ngDevMode && setInjectorProfilerContext(prevInjectorProfilerContext!);
    setInjectImplementation(previousInjectImplementation);
  }
}
```

5.6 Ogni NgModule viene istanziato eagerly

angular:packages/core/src/di/provider_collection.ts:L324-L344

```ts
    if (!isDuplicate) {
      // Track the InjectorType and add a provider for it.
      // It's important that this is done after the def's imports.
      const factory = getFactoryDef(defType) || (() => new defType!());
      // ...
      // Provider to create `defType` using its factory.
      visitor({provide: defType, useFactory: factory, deps: EMPTY_ARRAY}, defType);

      // Make this `defType` available to an internal logic that calculates injector scope.
      visitor({provide: INJECTOR_DEF_TYPES, useValue: defType, multi: true}, defType);

      // Provider to eagerly instantiate `defType` via `INJECTOR_INITIALIZER`.
      visitor(
        {provide: ENVIRONMENT_INITIALIZER, useValue: () => inject(defType!), multi: true},
        defType,
      );
    }
```

5.7 Gli app initializer

angular:packages/core/src/application/application_init.ts:L244-L255

```ts
    const asyncInitPromises = [];
    for (const appInits of this.appInits) {
      const initResult = runInInjectionContext(this.injector, appInits);
      if (isPromise(initResult)) {
        asyncInitPromises.push(initResult);
      } else if (isSubscribable(initResult)) {
        const observableAsPromise = new Promise<void>((resolve, reject) => {
          initResult.subscribe({complete: resolve, error: reject});
        });
        asyncInitPromises.push(observableAsPromise);
      }
    }
```

5.8 I token della configurazione di Spartacus

core-libs/core/src/config/config-tokens.ts:L10-L47

```ts
export function configFactory() {
  return deepMerge({}, inject(DefaultConfig), inject(RootConfig));
}

/**
```

```
 * Global Configuration, can be used to inject configuration to any part of the app
 */
@Injectable({
  providedIn: 'root',
  useFactory: configFactory,
})
export abstract class Config {}

export function defaultConfigFactory() {
  return deepMerge(
    {},
    ...(inject(DefaultConfigChunk, { optional: true }) ?? [])
  );
}
// ...
export const DefaultConfig = new InjectionToken('DefaultConfiguration', {
  providedIn: 'root',
  factory: defaultConfigFactory,
});

export function rootConfigFactory() {
  return deepMerge({}, ...(inject(ConfigChunk, { optional: true }) ?? []));
}
// ...
export const RootConfig = new InjectionToken('RootConfiguration', {
  providedIn: 'root',
  factory: rootConfigFactory,
});
```

5.9 `provideConfig` è solo un multi provider

core-libs/core/src/config/config-providers.ts:L18-L27

TS

```
export function provideConfig(
  config: Config = {},
  defaultConfig = false
): ValueProvider {
  return {
    provide: defaultConfig ? DefaultConfigChunk : ConfigChunk,
    useValue: config,
    multi: true,
  };
}
```

5.10 `UnifiedInjector`: root normale, lazy solo `self`

core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68

```ts
get<T>(
  token: Type<T> | InjectionToken<T> | AbstractType<T>,
  notFoundValue?: T
): Observable<T> {
  return this.injectors$.pipe(
    map((injector, index) =>
      injector.get<T>(
        token,
        notFoundValue ?? NOT_FOUND_SYMBOL,
        // we want to get only Self instances from all injectors except the
        // first one, which is a root injector
        index ? { self: true } : undefined
      )
    ),
    filter((instance) => instance !== NOT_FOUND_SYMBOL)
  );
}
```

5.11 Il parent di un modulo lazy

core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L98

```ts
  let parentInjector: Injector;

  if (!dependencyModuleRefs.length) {
    parentInjector = this.injector;
  } else if (dependencyModuleRefs.length === 1) {
    parentInjector = dependencyModuleRefs[0].injector;
  } else {
    parentInjector = new CombinedInjector(
      this.injector,
      dependencyModuleRefs.map((moduleRef) => moduleRef.injector)
    );
  }

  return this.resolveModuleFactory(moduleFunc).pipe(
    map(([moduleFactory]) => moduleFactory.create(parentInjector)),
    concatMap((moduleRef) => this.runModuleInitializersForModule(moduleRef)),
```

5.12 `CONFIG_INITIALIZER` e location initializer

core-libs/core/src/config/config-initializer/config-initializer.module.ts:L46-L64

```ts
    providers: [
      {
        provide: CONFIG_INITIALIZER_FORROOT_GUARD,
        useValue: true,
```

```
    },
    {
      provide: APP_INITIALIZER,
      multi: true,
      useFactory: configInitializerFactory,
      deps: [
        ConfigInitializerService,
        [new Optional(), CONFIG_INITIALIZER],
      ],
    },
    provideLocationInitializerFactory(() => {
      // Hold on the initial navigation until the Spartacus configuration is stable
      const configInitializer = inject(ConfigInitializerService);
      return () => lastValueFrom(configInitializer.getStable());
    }),
```

5.13 Facade: provider root con factory, override nel modulo lazy

feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L18

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: ActiveCartFacade,
      feature: CART_BASE_CORE_FEATURE,
```

feature-libs/cart/base/core/facade/facade-providers.ts:L25-L30

```ts
export const facadeProviders: Provider[] = [
  ActiveCartService,
  {
    provide: ActiveCartFacade,
    useExisting: ActiveCartService,
  },
```

6. Configurazione

Nel contesto DI la "configurazione" sono i token stessi. Tabella dei token rilevanti:

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`Config`	classe astratta (token, augmentable)	`deepMerge({}, DefaultConfig, RootConfig)`	core-libs/core/src/config/config-tokens.ts:L10-L21	ovunque; es. core-libs/core/src/config/services/configuration.service.ts:L46-L48

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
DefaultConfig	InjectionToken root con factory	merge dei DefaultConfigChunk	core-libs/core/src/config/config-tokens.ts:L23-L35	core-libs/core/src/config/config-tokens.ts:L11-L11, core-libs/core/src/config/services/configuration.service.ts:L44-L44
RootConfig	InjectionToken root con factory	merge dei ConfigChunk	core-libs/core/src/config/config-tokens.ts:L37-L47	core-libs/core/src/config/config-tokens.ts:L11-L11, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L32-L32
ConfigChunk	InjectionToken<Config[]> multi	nessuno (letto optional → [])	core-libs/core/src/config/config-tokens.ts:L53-L53	core-libs/core/src/config/config-tokens.ts:L38-L38, core-libs/core/src/config/services/configuration.service.ts:L58-L61
DefaultConfigChunk	InjectionToken<Config[]> multi	nessuno (letto optional → [])	core-libs/core/src/config/config-tokens.ts:L60-L62	core-libs/core/src/config/config-tokens.ts:L26-L26, core-libs/core/src/config/services/configuration.service.ts:L62-L65
CONFIG_INITIALIZER	InjectionToken<ConfigInitializer | null> multi	letto con new Optional()	core-libs/core/src/config/config-initializer/config-initializer.ts:L33-L35	core-libs/core/src/config/config-initializer/config-initializer.module.ts:L55-L58
CONFIG_INITIALIZER_FORROOT_GUARD	InjectionToken<void>	assente → isStable vero	core-libs/core/src/config/config-initializer/config-initializer.ts:L37-L39	core-libs/core/src/config/config-initializer/config-initializer.service.ts:L42-L44
MODULE_INITIALIZER	InjectionToken<(() => any)[]> multi	[] (lazy, self) / optional (eager)	core-libs/core/src/lazy-loading/tokens.ts:L19-L20	core-libs/core/src/lazy-loading/lazy-modules.service.ts:L153-L157, core-libs/core/src/lazy-loading/lazy-loading.module.ts:L47-L47
LOCATION_INITIALIZED_MULTI	InjectionToken<LocationInitializer[]> multi	optional → []	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L36-L38	core-libs/core/src/routing/routing.module.ts:L66-L71
ConfigValidatorToken	InjectionToken multi	optional	core-libs/core/src/config/config-validator/config-validator.ts:L11-L13	core-libs/core/src/config/config-validator/config-validator.module.ts:L55-L58
features.disableConfigUpdates	boolean (deprecata)	false (non configurata)	core-libs/core/src/features-config/config/features-config.ts:L101-L116	core-libs/core/src/config/services/configuration.service.ts:L103-L103
context.baseSite (statico)	config	se presente disattiva SiteContextConfigInitializer	—	core-libs/core/src/site-context/site-context.module.ts:L37-L40
Angular APP_INITIALIZER	InjectionToken<ReadonlyArray<fn>> multi	optional → []	angular:packages/core/src/application/application_init.ts:L150-L152	angular:packages/core/src/application/application_init.ts:L223-L223
Angular ENVIRONMENT_INITIALIZER	InjectionToken<ReadonlyArray<() => void>> multi	EMPTY_ARRAY (self)	angular:packages/core/src/di/initializer_token.ts:L24-L26	angular:packages/core/src/di/r3_injector.ts:L436-L436
Angular INJECTOR_SCOPE	InjectionToken<InjectorScope | null>	—	angular:packages/core/src/di/scope.ts:L18-L20	angular:packages/core/src/di/r3_injector.ts:L238-L241

7. Commenti e note notevoli dal sorgente

- `@deprecated from v19.0.0, use provideEnvironmentInitializer instead` (`ENVIRONMENT_INITIALIZER`) — `angular:packages/core/src/di/initializer_token.ts:L15-L15`.
- "Previously, this was achieved using the `APP_INITIALIZER` token which is now deprecated." — `angular:packages/core/src/application/application_init.ts:L167-L167`. Anche la costante `APP_INITIALIZER` (`angular:packages/core/src/application/application_init.ts:L150-L152`) porta il tag `@deprecated from v19.0.0, use provideAppInitializer instead` — `angular:packages/core/src/application/application_init.ts:L38-L38`; Spartacus la usa ancora in molti `forRoot()` (es. `core-libs/core/src/config/config-initializer/config-initializer.module.ts:L51-L58`).
- `@deprecated use the standalone function runInInjectionContext instead` (`EnvironmentInjector.runInContext`) — `angular:packages/core/src/di/r3_injector.ts:L174-L174`.
- `@deprecated from v5 use the new signature Injector.create(options)` — `angular:packages/core/src/di/injector.ts:L83-L86`.
- `// TODO: improve the typings here.` / "`token` can be a multi: true provider definition, which is considered as a Token but not represented in the typings" — `angular:packages/core/src/di/injector_compatibility.ts:L108-L109`. È la conferma, lato `inject`, che per i multi provider il provider stesso fa da chiave (§5.1).
- "General rule is, that all config provided in libraries should be provided as default config." — `core-libs/core/src/config/config-tokens.ts:L58-L58`.
- "To make sure ConfigurationService will be instantiated, we inject it into module constructor" — `core-libs/core/src/config/config.module.ts:L15-L17`.
- "We need to use subscription to propagate changes to the config from the beginning. It will be possible to make it lazy, when we drop this compatibility feature" — `core-libs/core/src/config/services/configuration.service.ts:L51-L54`.
- "@internal Not a part of a public API, used by APP_INITIALIZER to initialize all provided CONFIG_INITIALIZERS" — `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L130`.
- "CAUTION: The injection context is lost after leaving the synchronous execution, so any injections must be performed before moving into asynchronous execution." — `core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L18-L20`. Unica menzione di `runInInjectionContext` nelle librerie (comando: `grep -rn "runInInjectionContext" --include=*.ts core-libs feature-libs integration-libs | grep -v spec` → 1 riga).
- "CAUTION: Don't use Router and don't perform any Router navigation in its implementations." — `core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L33-L34`.
- `// using compiler here is for jit compatibility, there is no overhead for aot production builds as it will be stubbed` — `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L228-L230`. In Angular 21 `compileModuleAsync` è `Promise.resolve(this.compileModuleSync(moduleType))` — `angular:packages/core/src/linker/compiler.ts:L64-L66`.
- `ErrorHandlingModule.forRoot(), // Import this module before any other interceptor to handle HTTP errors efficiently` — `core-libs/core/src/base-core.module.ts:L28-L28`.

8. Extension point

EXTENSION POINT	MECCANISMO DI	DEFINIZIONE	RICETTA
Config dell'app	multi `ConfigChunk` via `provideConfig`/`provideConfigFactory`	core-libs/core/src/config/config-providers.ts:L18-L48	50 — Ricette config/DI
Config di default di una libreria	multi `DefaultConfigChunk` via `provideDefaultConfig`/`provideDefaultConfigFactory`	core-libs/core/src/config/config-providers.ts:L55-L79	50, 53 — feature lib lazy
Config asincrona	multi `CONFIG_INITIALIZER` via `provideConfigInitializer(Factory)`	core-libs/core/src/config/config-initializer/config-initializer.ts:L46-L67	50
Validazione config	multi `ConfigValidatorToken` via `provideConfigValidator`	core-libs/core/src/config/config-validator/config-validator.ts:L27-L35	50
Sostituire un servizio/adapter	provider non-multi sullo stesso token (ultimo vince)	angular:packages/core/src/di/r3_injector.ts:L513-L521, es. core-libs/core/src/occ/adapters/product/product-occ.module.ts:L34-L37	50, 52 — OCC
Aggiungere un converter	multi provider su `*_NORMALIZER`/`*_SERIALIZER`	core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51, core-libs/core/src/util/converter.service.ts:L130-L137	52
Interceptor HTTP	multi `HTTP_INTERCEPTORS` (`withInterceptorsFromDi()`)	core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L12-L23	50
Inizializzazione di un modulo lazy	multi `MODULE_INITIALIZER`	core-libs/core/src/lazy-loading/tokens.ts:L19-L20	53
Facade lazy	`useFactory: () => facadeFactory(...)` + `useExisting` nel modulo lazy	core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24, feature-libs/cart/base/core/facade/facade-providers.ts:L27-L30	50, 53
Attendere prima della navigazione iniziale	multi `LOCATION_INITIALIZED_MULTI` via `provideLocationInitializer(Factory)`	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L43-L62	50
Provider per un singolo componente CMS	`providers` del mapping `cmsComponents` → `Injector.create`	core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L33-L48	51 — CMS/outlet

Quali simboli sono API pubblica (e quindi sicuri da usare negli override) è discusso in 54 — upgrade-safe. Due note verificate qui:

- `CONFIG_INITIALIZER_FORROOT_GUARD` non è riesportato dal barrel `core-libs/core/src/config/config-initializer/index.ts` — core-libs/core/src/config/config-initializer/index.ts:L7-L14.
- `CombinedInjector` non è riesportato da `core-libs/core/src/util/index.ts` (che esporta `./converter.service`) ed è usato solo da `LazyModulesService` — core-libs/core/src/util/index.ts:L11-L11, core-

libs/core/src/lazy-loading/lazy-modules.service.ts:L30-L30.

Esempio illustrativo (non presente nel repo; usa solo simboli esportati da `@spartacus/core` e `@angular/core` citati sopra) di un override in un NgModule applicativo. Per quanto visto in §3.4 i `providers` del modulo vengono processati dopo i suoi `imports`:

```ts
// Esempio illustrativo, non presente nel repo
@NgModule({
  imports: [SpartacusModule],
  providers: [
    provideConfig({ backend: { occ: { baseUrl: 'https://example.com' } } }), // ConfigChunk: vince sui default
    { provide: ProductAdapter, useClass: MyProductAdapter },                  // non-multi: sostituisce OccProductAdapter
    { provide: PRODUCT_NORMALIZER, useExisting: MyNormalizer, multi: true }, // multi: si aggiunge in coda
  ],
})
export class AppCustomizationModule {}
```

`ProductAdapter` e `PRODUCT_NORMALIZER` sono raggiungibili dal public API di `@spartacus/core` — core-libs/core/public_api.ts:L29-L29, core-libs/core/src/product/index.ts:L7-L7, core-libs/core/src/product/connectors/index.ts:L7-L7, core-libs/core/src/product/connectors/product/index.ts:L8-L9. MyProductAdapter e MyNormalizer sono classi applicative ipotetiche.

DA VERIFICARE: l'esempio non è stato compilato (`node_modules` non installato nel repo), quindi la correttezza dei tipi non è provata da un build.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Multi token e moduli lazy.** Un multi token risolve solo i provider del primo injector che ha il record: i chunk forniti in un modulo lazy non sono visti da `inject(ConfigChunk)` nel root; li raccoglie `UnifiedInjector` con `{self: true}` — angular:packages/core/src/di/r3_injector.ts:L347-L371, core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67.
2. **Anche il contrario.** Un modulo lazy che fornisce anche un solo provider multi per un token ha un proprio record multi: `injector.get(TOKEN)` da quel modulo vede **solo** i suoi provider, non quelli del root (stessa regola del punto 1). Per questo `ConverterService` usa `getMulti`, che concatena root + lazy — core-libs/core/src/util/converter.service.ts:L52-L65, core-libs/core/src/lazy-loading/unified-injector.ts:L91-L91.
3. **Mescolare multi e non-multi.** In dev mode entrambi i casi lanciano `throwMixedMultiProviderError()`. In produzione un non-multi dopo un multi sovrascrive silenziosamente il record; un multi dopo un non-multi fa fallire la creazione dell'injector con un `TypeError` (`multiRecord.multi` è `undefined`, angular:packages/core/src/di/r3_injector.ts:L692-L702). `UnifiedInjector.getMulti` lancia comunque se il valore non è un array — angular:packages/core/src/di/r3_injector.ts:L501-L521, core-libs/core/src/lazy-loading/unified-injector.ts:L84-L88.
4. **Gli array nella config si sostituiscono.** `deepMerge` assegna gli array per sostituzione: un chunk con un array rimpiazza completamente quello di default — core-libs/core/src/config/utils/deep-merge.ts:L7-

L9, core-libs/core/src/config/utils/deep-merge.ts:L36-L45.

5. `Config` **è mutabile.** L'oggetto viene mutato dopo la creazione da `ConfigInitializerService` e da `ConfigurationService`: chi legge la config nel costruttore prima che sia stabile può vedere valori parziali; per questo esiste `getStable()`, che però restituisce lo stesso oggetto, non uno snapshot — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L160-L165, core-libs/core/src/config/services/configuration.service.ts:L103-L105, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70.
6. `isStable` **sempre vero senza** `forRoot()`. Se `ConfigInitializerModule.forRoot()` non è importato manca il guard e `isStable` è `true` — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L42-L44.
7. **Initializer senza scope = bootstrap rotto.** Un `CONFIG_INITIALIZER` senza `scopes` provoca `throw new Error('CONFIG_INITIALIZER should provide scope!')` dentro l'`APP_INITIALIZER` — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L144-L146.
8. **Gli** `APP_INITIALIZER` **partono tutti insieme.** Sono avviati in sequenza sincrona e girano in parallelo: un `APP_INITIALIZER` diverso da quello della config non attende i `CONFIG_INITIALIZER`, deve usare `getStable()` come fa `ConfigValidatorModule` — angular:packages/core/src/application/application_init.ts:L245-L263, core-libs/core/src/config/config-validator/config-validator.module.ts:L28-L35.
9. **Observable che non completa.** Un `APP_INITIALIZER` che restituisce un Observable che non completa blocca il bootstrap (`subscribe({complete: resolve})`) — angular:packages/core/src/application/application_init.ts:L249-L253.
10. `MODULE_INITIALIZER` **senza injection context (lazy).** Nei moduli lazy sono invocati direttamente (`initFunctions[i]()`), non con `runInInjectionContext`: al loro interno `inject()` fallisce se non c'è un contesto attivo. Nel caso eager girano invece dentro il frame dell'`APP_INITIALIZER`, che è eseguito in injection context — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L193-L194, core-libs/core/src/lazy-loading/lazy-loading.module.ts:L23-L27, angular:packages/core/src/application/application_init.ts:L246-L246. Soluzione: usare `useFactory` con `deps` (come `AsmCoreModule`, feature-libs/asm/core/asm-core.module.ts:L22-L35) o iniettare nella factory e chiudere i valori nella funzione restituita.
11. `inject()` **dopo** `await`**.** Lancia `MISSING_INJECTION_CONTEXT` — angular:packages/core/src/di/contextual.ts:L25-L27, angular:packages/core/src/di/injector_compatibility.ts:L97-L103, core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L18-L20.
12. **Ogni NgModule viene istanziato al bootstrap.** Via `ENVIRONMENT_INITIALIZER () => inject(defType)`: i costruttori dei moduli (es. `ConfigModule`) e le loro dipendenze vengono creati subito — angular:packages/core/src/di/provider_collection.ts:L339-L343, core-libs/core/src/config/config.module.ts:L17-L19.
13. **Ordine dei provider negli NgModule.** I provider dei `ModuleWithProviders` importati sono processati dopo tutti gli import del modulo; i provider propri del modulo dopo ancora. Un override non-multi va quindi messo in un punto successivo (es. `providers` dell'`AppModule`). Attenzione: il dedup vale per la definizione del modulo, non per i `providers` dei `ModuleWithProviders`, che sono aggiunti a ogni import (un `forRoot()` importato due volte duplica i suoi multi provider) — angular:packages/core/src/di/provider_collection.ts:L316-L354, angular:packages/core/src/di/provider_collection.ts:L360-L362, angular:packages/core/src/di/r3_injector.ts:L521-L521.
14. `providedIn: 'root'` **non è in scope in un injector lazy.** Un injector di modulo lazy ha scope solo `environment`: con `{self: true}` i token `providedIn: 'root'` risultano non trovati — angular:packages/core/src/render3/ng_module_ref.ts:L94-L94, angular:packages/core/src/di/r3_injector.ts:L551-L561, angular:packages/core/src/di/r3_injector.ts:L374-L376.
15. `CombinedInjector` **non supporta** `self`**.** Restituisce `notFoundValue` o lancia; va usato solo come parent — core-libs/core/src/util/combined-injector.ts:L44-L51.

16. `useValue: undefined` **è un provider valido.** Il controllo è `USE_VALUE in value`; un `useExisting`/`useFactory`/`useClass` falsy invece non viene riconosciuto come tale — angular:packages/core/src/di/provider_collection.ts:L404-L427.
17. **Converter senza registrazioni.** `convert`/`pipeable` restituiscono la sorgente invariata con un cast: il tipo dichiarato `T` non corrisponde al valore reale — core-libs/core/src/util/converter.service.ts:L80-L128.
18. `finishScopes` **e** `indexOf`**.** Usa `splice(indexOf(scope), 1)`: se uno scope non fosse presente, `indexOf` = -1 rimuoverebbe l'ultimo elemento — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L77-L83.

---

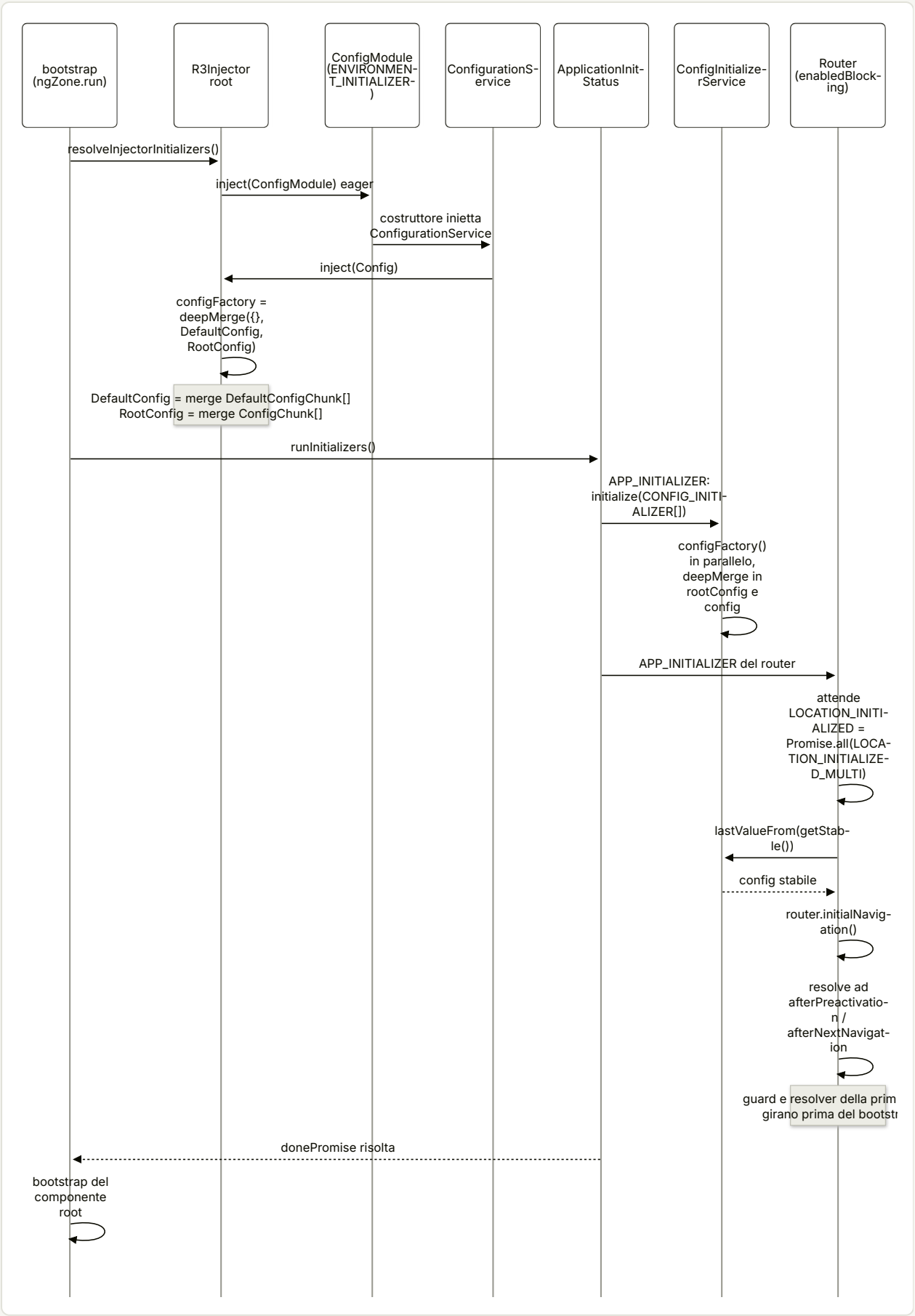
10. Come fare debug

- **Leggere il path dell'errore.** In dev mode un `PROVIDER_NOT_FOUND` o un ciclo riportano tutto il path di dipendenza, assemblato risalendo lo stack degli `R3Injector.get` — angular:packages/core/src/di/r3_injector.ts:L384-L410. In produzione il messaggio è vuoto (`new RuntimeError(errorCode, null)`) — angular:packages/core/src/di/r3_injector.ts:L411-L413: riprodurre in dev.
- **"inject() must be called from an injection context".** Cercare `inject(` dopo un `await`, dentro `subscribe`, `setTimeout` o dentro un `MODULE_INITIALIZER` lazy (trappola 10). Il messaggio esatto è in angular:packages/core/src/di/injector_compatibility.ts:L102-L102. Per verificare a runtime usare `isInInjectionContext()` — angular:packages/core/src/di/contextual.ts:L64-L66.
- **Quale config vince?** Confrontare `inject(DefaultConfig)` e `inject(RootConfig)` (i due input di `configFactory`, core-libs/core/src/config/config-tokens.ts:L10-L12) e osservare `ConfigurationService.unifiedConfig$` per vedere l'effetto dei moduli lazy — core-libs/core/src/config/services/configuration.service.ts:L92-L100.
- **Bootstrap che non parte.** Controllare che ogni `APP_INITIALIZER` restituisca una Promise che si risolve o un Observable che completa (angular:packages/core/src/application/application_init.ts:L247-L253), che ogni `CONFIG_INITIALIZER` abbia `scopes` e che le `configFactory()` non restino appese (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L144-L173); il location initializer di Spartacus blocca la navigazione iniziale finché `getStable()` non emette — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64.
- **Warning di scope duplicati.** In dev mode due initializer sullo stesso scope producono "More than one CONFIG_INITIALIZER is initializing the same config scope." — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L148-L152.
- **Override che "non prende".** Verificare (a) che il token sia la stessa istanza (angular:packages/core/src/di/injection_token.ts:L26-L29), (b) che il provider sia registrato dopo l'originale nello stesso injector (§3.4), (c) che il consumatore non stia in un modulo lazy che ri-fornisce il token nel proprio injector, (d) per le facade, che l'override sia nell'injector del modulo lazy da cui il proxy legge con `moduleRef.injector.get(facadeClass)` — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52.
- **Ispezionare i multi provider lazy.** `UnifiedInjector.getMulti(TOKEN)` mostra l'accumulo per injector — core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93; `LazyModulesService.modules$` emette ogni `NgModuleRef` inizializzato — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L68.
- **Errori** `MODULE_INITIALIZER`**.** Sono loggati come "MODULE_INITIALIZER promise was rejected during app initialization." (eager) o "... while lazy loading a module." (lazy) — core-libs/core/src/lazy-loading/lazy-loading.module.ts:L28-L33, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L162-L168.

11. Domande di autoverifica

1. **Se due moduli forniscono** `{provide: ProductAdapter, useClass: ...}` **nello stesso injector, quale vince?** L'ultimo processato: per i provider non-multi `records.set(token, record)` sovrascrive — angular:packages/core/src/di/r3_injector.ts:L513-L521. L'ordine dipende dall'appiattimento (imports, poi provider dei `ModuleWithProviders`, poi `providers` del modulo) — angular:packages/core/src/di/provider_collection.ts:L292-L354.
2. **Perché** `provideConfig` **vince su** `provideDefaultConfig` **indipendentemente dall'ordine?** Perché finiscono su due token diversi (`ConfigChunk` vs `DefaultConfigChunk`) e `configFactory` fonde `RootConfig` dopo `DefaultConfig` — core-libs/core/src/config/config-providers.ts:L18-L61, core-libs/core/src/config/config-tokens.ts:L10-L12.
3. **Un** `provideDefaultConfig` **dentro un modulo lazy finisce in** `inject(Config)` **del root?** Non tramite il DI di Angular (il multi token si ferma al primo injector con record); ci arriva perché `ConfigurationService` legge i chunk `self` via `UnifiedInjector` e li fonde in `config`, salvo `disableConfigUpdates` — angular:packages/core/src/di/r3_injector.ts:L347-L371, core-libs/core/src/config/services/configuration.service.ts:L57-L106.
4. **Che differenza c'è fra** `useExisting` **e** `useClass` **per esporre un servizio sotto un token astratto?** `useExisting` crea un alias (`ɵɵinject(useExisting)`) e condivide l'istanza; `useClass` crea una nuova istanza per quel token — angular:packages/core/src/di/r3_injector.ts:L657-L677.
5. **Un** `InjectionToken` **con** `factory` **ma senza** `providedIn` **dove è disponibile?** In `'root'`, perché è il default nel costruttore — angular:packages/core/src/di/injection_token.ts:L110-L116.
6. **Perché** `inject()` **funziona in una** `useFactory` **ma non dentro un** `subscribe`**?** `R3Injector.get` imposta l'injector corrente solo per la durata sincrona della risoluzione e lo ripristina nel `finally`; fuori da quel frame `injectInjectorOnly` trova `undefined` e lancia — angular:packages/core/src/di/r3_injector.ts:L341-L342, angular:packages/core/src/di/r3_injector.ts:L417-L420, angular:packages/core/src/di/injector_compatibility.ts:L97-L103.
7. **In che ordine girano environment initializer e app initializer al bootstrap?** Prima `resolveInjectorInitializers()` (environment, inclusa l'istanziazione eager dei moduli), poi `ApplicationInitStatus.runInitializers()`; il root component parte a `donePromise` risolta — angular:packages/core/src/platform/bootstrap.ts:L87-L93, angular:packages/core/src/platform/bootstrap.ts:L137-L144.
8. **Come fa Spartacus a impedire la navigazione iniziale prima che la config asincrona sia pronta?** Il router con `enabledBlocking` attende `LOCATION_INITIALIZED`; Spartacus lo sostituisce con il `Promise.all` dei `LOCATION_INITIALIZED_MULTI`, tra cui `lastValueFrom(getStable())` — angular:packages/router/src/provide_router.ts:L441-L469, core-libs/core/src/routing/routing.module.ts:L66-L71, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64.
9. **Quale injector è parent di un modulo lazy con due dipendenze?** Un `CombinedInjector(root, [depA, depB])`, che cerca prima `self` nelle dipendenze, poi a tutti i livelli, poi nel root — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L89-L94, core-libs/core/src/util/combined-injector.ts:L53-L70.
10. **Perché il** `ConverterService` **vede anche i normalizer dei moduli lazy?** Perché legge `unifiedInjector.getMulti(token)`, che concatena i multi provider di root e injector lazy, e svuota la cache a ogni nuovo injector — core-libs/core/src/util/converter.service.ts:L32-L65, core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93.

TAVOLA 01.1 **4.3 Costruzione di Config e config asincrona al bootstrap**



02

NgModule, standalone e bootstrap dell'applicazione

`bootstrapApplication` contro `bootstrapModule`: la funzione interna comune e l'appiattimento dei provider dei moduli.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Sorgenti Angular: `/tmp/angular` @ `v21.2.23` (prefisso `angular:`).

---

1. A cosa serve e perché esiste

Un'app Angular 21 può nascere in due modi: dal **bootstrap standalone** (`bootstrapApplication(RootComponent, appConfig)`) o dal **bootstrap per NgModule** (`platformBrowser().bootstrapModule(AppModule)`). Entrambi finiscono nella stessa funzione interna `bootstrap()` (angular:packages/core/src/platform/bootstrap.ts:L75-L181) e, quando c'è di mezzo un NgModule, entrambi usano lo stesso algoritmo di appiattimento dei provider, `walkProviderTree` (angular:packages/core/src/di/create_injector.ts:L42-L56, angular:packages/core/src/di/provider_collection.ts:L151-L189).

Spartacus si trova esattamente a cavallo tra i due mondi:

- l'app demo e le app generate da `ng add` sono **standalone**: `main.ts` chiama `bootstrapApplication(AppComponent, appConfig)` (projects/storefrontapp/src/main.ts:L17-L17) e il root component `AppComponent` è standalone (projects/storefrontapp/src/app/app.component.ts:L14-L19);
- ma l'intera libreria si configura ancora tramite **NgModule** e `forRoot()`: `BaseCoreModule` aggrega 16 `forRoot()` (core-libs/core/src/base-core.module.ts:L26-L46), `BaseStorefrontModule` aggrega `BaseCoreModule.forRoot()` e i moduli UI (core-libs/storefront/base-storefront.module.ts:L26-L49);
- il "ponte" tra i due mondi è una sola riga: `importProvidersFrom(AppModule)` dentro `appConfig` (projects/storefrontapp/src/app/app.config.ts:L33-L33).

Capire questo capitolo serve a rispondere a domande concrete che si presentano in ogni progetto Spartacus:

- perché l'ordine degli `imports` di `AppModule` conta, e perché i provider di `StoreModule.forRoot({})` finiscono **dopo** quelli di tutto `SpartacusModule` pur essendo dichiarati prima? (§3.4, angular:packages/core/src/di/provider_collection.ts:L316-L321);
- perché importare `OutletModule` e anche `OutletModule.forRoot()` non duplica il modulo ma aggiunge i provider? (angular:packages/core/src/di/provider_collection.ts:L360-L362);
- perché in Angular 21 un'app Spartacus deve fornire esplicitamente `provideZoneChangeDetection(...)`? (angular:packages/core/src/application/create_application.ts:L66-L71);
- perché lato server `main.server.ts` esporta una **funzione** che riceve un `BootstrapContext` e non un modulo? (angular:packages/core/src/application/create_application.ts:L47-L54);
- cosa fa, e cosa **non** fa, lo schematic `modernize-app-to-standalone-bootstrap-application`? (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/index.ts:L26-L54).

Perimetro. Questo capitolo copre: metadati `@NgModule`, `ModuleWithProviders`/`forRoot`, appiattimento (`walkProviderTree`), componenti standalone, i due percorsi di bootstrap, piattaforma, `ApplicationRef`, `ApplicationConfig` e il loro uso in Spartacus. Restano altrove:

- la risoluzione dei token nel `R3Injector`, la regola "l'ultimo provider vince", `inject()`, `EnvironmentProviders` in generale → 01 — Dependency Injection (in particolare §3.4 e §3.7);
- `APP_INITIALIZER`, `CONFIG_INITIALIZER`, `LOCATION_INITIALIZED`, la navigazione iniziale bloccante e il primo render → 10 — Bootstrap e inizializzazione;
- il dettaglio di `ng add`, delle migrazioni e dei `modernize-app-*` → 43 — Schematics e migrazioni;
- i moduli lazy di Spartacus (`LazyModulesService`) → 03 — Lazy loading e 12 — Feature module e lazy loading;

- zone/zoneless nel dettaglio → 04 — Change detection, zone e signals; SSR → 07 — SSR, hydration e TransferState.

2. Mappa dei file

2.1 Angular

PATH	RUOLO
angular:packages/core/src/metadata/ng_module.ts	interfaccia e decoratore `@NgModule` (`providers`, `declarations`, `imports`, `exports`, `bootstrap`, `schemas`, `id`, `jit`)
angular:packages/core/src/metadata/ng_module_def.ts	`NgModuleDef` (`ɵmod`): scope di compilazione e `bootstrap`
angular:packages/core/src/di/interface/defs.ts	`ɵɵdefineInjector` (`ɵinj`), `InjectorTypeWithProviders`, `getInjectorDef`
angular:packages/core/src/di/interface/provider.ts	`ModuleWithProviders`, `EnvironmentProviders`, `isEnvironmentProviders`
angular:packages/core/src/di/provider_collection.ts	`makeEnvironmentProviders`, `provideEnvironmentInitializer`, `importProvidersFrom`, `walkProviderTree`
angular:packages/core/src/di/create_injector.ts	`createInjectorWithoutInjectorInstances` (usa `importProvidersFrom`)
angular:packages/core/src/metadata/directives.ts	`standalone?`, `Component.imports`
angular:packages/compiler-cli/src/ngtsc/core/src/compiler.ts	default `standalone: true` in AOT da v19
angular:packages/compiler-cli/src/ngtsc/annotations/ng_module/src/handler.ts	cosa finisce negli `imports` di `ɵinj`
angular:packages/core/src/render3/standalone_service.ts	injector `Standalone[...]` per componenti standalone creati dinamicamente
angular:packages/core/src/render3/ng_module_ref.ts	`NgModuleRef`, `NgModuleFactory`, `EnvironmentNgModuleRefAdapter`
angular:packages/core/src/platform/platform.ts	`createPlatform`, `createPlatformFactory`, `createOrReusePlatformInjector`, `getPlatform`
angular:packages/core/src/platform/platform_ref.ts	`PlatformRef`, `bootstrapModule`, `bootstrapModuleFactory`, `destroy`
angular:packages/core/src/platform/bootstrap.ts	`bootstrap()` comune, `ENABLE_ROOT_COMPONENT_BOOTSTRAP`, `_moduleDoBootstrap`
angular:packages/core/src/application/create_application.ts	`internalCreateApplication` (percorso standalone)
angular:packages/core/src/application/application_config.ts	`ApplicationConfig`, `mergeApplicationConfig`
angular:packages/core/src/application/application_ref.ts	`ApplicationRef`, `BootstrapOptions`, `APP_BOOTSTRAP_LISTENER`
angular:packages/core/src/application/application_module.ts	`ApplicationModule` (istanza eager di `ApplicationRef`)

PATH	RUOLO
angular:packages/platform-browser/src/browser.ts	bootstrapApplication, createApplication, platformBrowser, BrowserModule, BootstrapContext
angular:packages/router/src/router_module.ts	RouterModule.forRoot: il pattern forRoot di riferimento

2.2 Spartacus — app demo

PATH	RUOLO
projects/storefrontapp/project.json	entry browser (main.ts), server (main.server.ts), ssr.entry (server.ts), polyfill zone.js
projects/storefrontapp/src/main.ts	bootstrapApplication(AppComponent, appConfig)
projects/storefrontapp/src/main.server.ts	funzione bootstrap(context) per SSR
projects/storefrontapp/src/server.ts	passa bootstrap a ngExpressEngine
projects/storefrontapp/src/app/app.component.ts	root standalone app-root → <cx-storefront />
projects/storefrontapp/src/app/app.config.ts	ApplicationConfig browser + importProvidersFrom(AppModule)
projects/storefrontapp/src/app/app.config.server.ts	mergeApplicationConfig(appConfig, serverConfig)
projects/storefrontapp/src/app/app.module.ts	AppModule: ponte NgModule verso Spartacus
projects/storefrontapp/src/app/app.module.server.ts	AppServerModule (solo provideServer)
projects/storefrontapp/src/app/private/private.providers.ts	provider privati di test/e2e (makeEnvironmentProviders)
projects/storefrontapp/src/app/spartacus/spartacus.module.ts	aggregatore Base/Features/Configuration
projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts	import dei moduli core/UI/feature, condizionati da environment
projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts	configurazione globale via provideConfig

2.3 Spartacus — librerie e schematics

PATH	RUOLO
core-libs/core/src/base-core.module.ts	aggregatore dei forRoot() core
core-libs/storefront/base-storefront.module.ts	aggregatore storefront (include BaseCoreModule.forRoot())
core-libs/storefront/router/app-routing.module.ts	RouterModule.forRoot([]) di Spartacus
core-libs/storefront/layout/main/storefront.component.ts	StorefrontComponent standalone (cx-storefront)
core-libs/storefront/layout/main/storefront-component.module.ts	shim NgModule che importa/esporta il componente standalone
core-libs/core/src/config/config.module.ts, core-libs/core/src/state/state.module.ts, core-libs/core/src/process/process.module.ts, core-libs/core/src/error-handling/error-handling.module.ts, core-libs/core/src/site-context/site-context.module.ts	esempi di forRoot() core

PATH	RUOLO
core-libs/storefront/cms-structure/outlet/outlet.module.ts, core-libs/storefront/cms-structure/routing/routing.module.ts, core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.module.ts	esempi di forRoot()/forChild() storefront
core-libs/core/src/lazy-loading/lazy-modules.service.ts	istanziazione degli NgModule lazy via NgModuleFactory
core-libs/schematics/src/collection.json	registrazione dello schematic modernize-app-to-standalone-bootstrap-application
core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/index.ts	catena di regole CSR+SSR
core-libs/schematics/src/add-spartacus/index.ts	ng add: richiede app standalone

3. Come funziona passo per passo

3.1 Un NgModule, due definizioni

L'interfaccia NgModule dichiara providers, declarations, imports, exports, bootstrap, schemas, id, jit (angular:packages/core/src/metadata/ng_module.ts:L33-L202). Due dettagli di tipo contano:

- providers accetta anche EnvironmentProviders, non solo Provider (angular:packages/core/src/metadata/ng_module.ts:L81-L81): per questo l'app demo può mettere in @NgModule.providers un oggetto costruito con makeEnvironmentProviders (§3.13);
- imports accetta tipi, ModuleWithProviders<{}> e array annidati (angular:packages/core/src/metadata/ng_module.ts:L138-L138): per questo SpartacusFeaturesModule può spalmare un array costruito a runtime (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L285-L285).

bootstrap elenca i componenti avviati quando il modulo viene bootstrappato (angular:packages/core/src/metadata/ng_module.ts:L172-L175); id registra il modulo in getNgModuleById, e senza id il modulo non è registrato (angular:packages/core/src/metadata/ng_module.ts:L189-L193). Il decoratore è creato con makeDecorator e in JIT chiama compileNgModule(type, meta) (angular:packages/core/src/metadata/ng_module.ts:L207-L224).

A runtime un NgModule compilato ha **due** definizioni statiche separate (angular:packages/core/src/metadata/ng_module_def.ts:L13-L15, angular:packages/core/src/metadata/ng_module_def.ts:L42-L77, angular:packages/core/src/di/interface/defs.ts:L197-L199):

- ɵmod (NgModuleDef): bootstrap, declarations, imports, exports, schemas, id — serve al **compilatore di template** (scope di compilazione);
- ɵinj (injector def): solo providers e imports — serve al **DI**.

getInjectorDef legge ɵinj solo se è proprietà "own" del tipo, non ereditata (angular:packages/core/src/di/interface/defs.ts:L253-L258): un NgModule che estende un altro NgModule non ne eredita i provider.

Il punto chiave è **cosa** il compilatore AOT mette negli imports di ɵinj:

- gli import top-level, saltando direttive e pipe ("can't carry providers") e i componenti standalone che non possono esportare provider (angular:packages/compiler-cli/src/ngtsc/annotations/ng_module/src/handler.ts:L798-L848);

- **anche** gli NgModule elencati in `exports`: un modulo ri-esportato porta i suoi provider a chi importa il modulo che lo ri-esporta (angular:packages/compiler-cli/src/ngtsc/annotations/ng_module/src/handler.ts:L867-L877).

Conseguenza pratica: `RouterModule` ha come `imports`/`exports` solo direttive standalone (`ROUTER_DIRECTIVES`), che quindi non entrano nell'injector (angular:packages/router/src/router_module.ts:L56-L56, angular:packages/router/src/router_module.ts:L104-L108). Importare `RouterModule` "nudo" (come fa `BaseStorefrontModule`, core-libs/storefront/base-storefront.module.ts:L29-L29) porta le direttive nel template scope, ma nessun provider del router: quelli arrivano solo da `RouterModule.forRoot(...)`.

3.2 `ModuleWithProviders` e il pattern `forRoot`

`ModuleWithProviders<T>` è una semplice interfaccia `{ngModule: Type<T>; providers?: Array<Provider | EnvironmentProviders>}`; usarla senza generico è deprecato (angular:packages/core/src/di/interface/provider.ts:L387-L396). L'equivalente interno è `InjectorTypeWithProviders`, "private type and should not be exported" (angular:packages/core/src/di/interface/defs.ts:L125-L136).

`forRoot()` **non** è un'API di Angular: è una convenzione. Una funzione statica restituisce un `ModuleWithProviders` che associa al modulo dei provider "da registrare una volta sola nel root". Il pattern canonico è `RouterModule.forRoot(routes, config)`, che restituisce `{ngModule: RouterModule, providers: [ROUTER_PROVIDERS, ROUTES, ROUTER_FORROOT_GUARD, ...]}` (angular:packages/router/src/router_module.ts:L133-L166). Angular protegge il router dal doppio `forRoot`: `provideForRootGuard` cerca `Router` con `skipSelf` e lancia `FOR_ROOT_CALLED_TWICE`, e il costruttore di `RouterModule` attiva la guardia in dev (angular:packages/router/src/router_module.ts:L109-L113, angular:packages/router/src/router_module.ts:L222-L233).

Spartacus usa `forRoot()` ovunque nella sua configurazione core (§3.14), ma **senza** guardie analoghe (§9, T4).

3.3 `EnvironmentProviders` e `importProvidersFrom` (richiamo)

Il dettaglio è in 01 §3.7; qui i fatti necessari al resto del capitolo:

- `EnvironmentProviders` è un tipo "brand" opaco, accettato solo nella creazione di un `EnvironmentInjector` (e quindi in un NgModule) e non in `@Component.providers` (angular:packages/core/src/di/interface/provider.ts:L345-L362); internamente contiene `ɵproviders` e opzionalmente `ɵfromNgModule` (angular:packages/core/src/di/interface/provider.ts:L364-L379).
- `makeEnvironmentProviders` incapsula l'array in `{ɵproviders}` e nient'altro (angular:packages/core/src/di/provider_collection.ts:L46-L52). `provideEnvironmentInitializer(fn)` lo usa per registrare un `ENVIRONMENT_INITIALIZER`, token che il JSDoc dichiara deprecato (angular:packages/core/src/di/provider_collection.ts:L54-L89).
- `importProvidersFrom(...sources)` accetta `Type`, `ModuleWithProviders` o array annidati (angular:packages/core/src/di/provider_collection.ts:L96-L99) e restituisce `{ɵproviders: internalImportProvidersFrom(true, sources), ɵfromNgModule: true}` (angular:packages/core/src/di/provider_collection.ts:L144-L149). Il JSDoc dice che i provider estratti valgono solo in application injector o altri environment injector, non in provider di componente (angular:packages/core/src/di/provider_collection.ts:L106-L143).
- In dev mode `importProvidersFrom` rifiuta un componente standalone (`IMPORT_PROVIDERS_FROM_STANDALONE`, angular:packages/core/src/di/provider_collection.ts:L163-L174); se il risultato finisce in un injector non-environment si ottiene `PROVIDER_IN_WRONG_CONTEXT` con messaggio dedicato grazie a `ɵfromNgModule` (angular:packages/core/src/render3/errors_di.ts:L50-L61).

- Il `R3Injector` apre ricorsivamente array ed `EnvironmentProviders` prima di registrare i record (angular:packages/core/src/di/r3_injector.ts:L725-L738).

Nota importante: `importProvidersFrom(AppModule)` è una **chiamata di funzione eseguita quando si valuta il modulo ES** `app.config.ts`, perché compare in un oggetto letterale (projects/storefrontapp/src/app/app.config.ts:L26-L35). L'appiattimento avviene quindi prima ancora di `bootstrapApplication`.

3.4 `walkProviderTree`: l'algoritmo di appiattimento

Questo è il cuore del capitolo: decide **in che ordine** i provider di tutti gli NgModule finiscono nell'environment injector, e quindi (per la regola "l'ultimo vince" di 01 §3.4) chi vince.

Ingresso. `internalImportProvidersFrom` crea un unico `Set` `dedup` per tutte le sorgenti della chiamata e raccoglie a parte i `ModuleWithProviders` top-level, i cui provider sono aggiunti dopo aver visitato tutte le sorgenti (angular:packages/core/src/di/provider_collection.ts:L151-L189). Lo stesso algoritmo vale per il percorso NgModule classico: `createInjectorWithoutInjectorInstances` costruisce i provider come `[additionalProviders, importProvidersFrom(defType)]` (angular:packages/core/src/di/create_injector.ts:L42-L56).

Classificazione del contenitore. `walkProviderTree` riconosce tre forme (angular:packages/core/src/di/provider_collection.ts:L245-L266):

1. un `InjectorType` (ha `ɵinj`) → NgModule;
2. un oggetto con `ngModule` (cioè un `ModuleWithProviders`) → si scarta l'involucro e si usa `ngModule` come `defType`;
3. un componente standalone.

Un componente non standalone, una direttiva o una pipe vengono ignorati (`return false`). In dev, uno stack `parents` rileva gli import ciclici (angular:packages/core/src/di/provider_collection.ts:L268-L273).

Componente standalone. Visita ricorsivamente `cmpDef.dependencies` (i suoi `imports`), una sola volta per tipo (angular:packages/core/src/di/provider_collection.ts:L278-L291).

NgModule. L'ordine di emissione è questo:

1. Se il modulo non è già in `dedup`: lo aggiunge, poi visita gli `imports` depth-first nell'ordine dichiarato (angular:packages/core/src/di/provider_collection.ts:L292-L310).
2. Gli import che erano `ModuleWithProviders` con `providers` vengono messi da parte; i loro `providers` sono emessi **solo dopo** che tutti gli import di quel modulo sono stati visitati — semantica View Engine, FW-1349 (angular:packages/core/src/di/provider_collection.ts:L316-L321).
3. Tre provider sintetici per il modulo: il tipo stesso (factory), `INJECTOR_DEF_TYPES` (multi) e un `ENVIRONMENT_INITIALIZER` che fa `inject(defType)`, cioè **istanziazione eager** del modulo; emessi dopo gli import (angular:packages/core/src/di/provider_collection.ts:L324-L344).
4. I `providers` di `@NgModule` (`ɵinj.providers`), sempre solo se non duplicato (angular:packages/core/src/di/provider_collection.ts:L346-L354).

Valore di ritorno. `walkProviderTree` restituisce `true` solo se il contenitore era un wrapper `ModuleWithProviders` con `providers !== undefined` (angular:packages/core/src/di/provider_collection.ts:L360-L362). Ne seguono due regole poco intuitive:

- i `providers` di un `ModuleWithProviders` sono aggiunti **anche se** il suo `ngModule` era già stato visitato (il dedup vale solo sui tipi);
- un `forRoot()` che restituisce `{ngModule}` **senza** `providers` si comporta esattamente come l'import diretto del modulo; uno con `providers: []` attiva il ramo "rinviato" con zero provider.

`processInjectorTypesWithProviders` appiattisce i `providers` di ogni `ModuleWithProviders` aprendo anche gli `EnvironmentProviders` e, in dev, li valida (angular:packages/core/src/di/provider_collection.ts:L195-L209, angular:packages/core/src/di/provider_collection.ts:L388-L402).

Istanziazione. Gli `ENVIRONMENT_INITIALIZER` sintetici sono eseguiti da `R3Injector.resolveInjectorInitializers()` in ordine, leggendo il token con `{self: true}` (angular:packages/core/src/di/r3_injector.ts:L426-L448): le istanze dei moduli vengono quindi create in **post-ordine** (prima gli importati, poi l'importatore). È così che girano i costruttori "di servizio" come quello di `ConfigModule`, che inietta `ConfigurationService` per istanziarlo (core-libs/core/src/config/config.module.ts:L13-L19).

3.5 Componenti standalone

Da Angular 19 componenti, direttive e pipe sono standalone per default: il compilatore AOT imposta `implicitStandaloneValue` a `true` per `@angular/core >= 19.0.0` (angular:packages/compiler-cli/src/ngtsc/core/src/compiler.ts:L477-L482), e in JIT `standalone` non specificato vale `true` (angular:packages/core/src/render3/jit/directive.ts:L393-L393). Per **dichiarare** un componente in un NgModule serve `standalone: false` (angular:packages/core/src/metadata/directives.ts:L52-L66, angular:packages/core/src/metadata/directives.ts:L627-L630).

`Component.imports` elenca le dipendenze di template di un componente standalone: altri componenti/direttive/pipe standalone **e** NgModule esistenti (angular:packages/core/src/metadata/directives.ts:L632-L640).

Per un componente standalone la definizione espone `dependencies` e `getStandaloneInjector`, che delega a `StandaloneService` dell'environment injector padre (angular:packages/core/src/render3/definition.ts:L359-L364). `StandaloneService` (`providedIn: 'environment'`) crea e mette in cache, per ogni componente standalone creato dinamicamente (root incluso), un environment injector figlio `Standalone[<Nome>]` con i provider raccolti dal grafo dei suoi `imports` via `internalImportProvidersFrom(false, ...)` — solo se la lista non è vuota (angular:packages/core/src/render3/standalone_service.ts:L16-L47, angular:packages/core/src/render3/standalone_service.ts:L62-L66). Nella creazione della root view, se il componente ha `getStandaloneInjector`, l'environment injector usato diventa quello standalone (angular:packages/core/src/render3/component_ref.ts:L150-L153).

Se si passa a `bootstrapApplication` un componente non standalone, `assertStandaloneComponentType` lancia `TYPE_IS_NOT_STANDALONE` (angular:packages/core/src/render3/errors.ts:L25-L37).

3.6 La piattaforma

`PlatformRef` è `@Injectable({providedIn: 'platform'})`: una per pagina, comune a tutte le app (angular:packages/core/src/platform/platform_ref.ts:L23-L33).

- `createPlatformFactory(parent, name, providers)` restituisce una factory che riusa la piattaforma esistente o ne crea una aggiungendo un token marker `Platform: <name>`; fuori da SSR verifica con `assertPlatform(marker)` che la piattaforma esistente sia della stessa configurazione (angular:packages/core/src/platform/platform.ts:L67-L90).
- `createPlatformInjector` crea un injector con `INJECTOR_SCOPE = 'platform'` e `PLATFORM_DESTROY_LISTENERS` (angular:packages/core/src/platform/platform.ts:L96-L105).
- `createPlatform` lancia `MULTIPLE_PLATFORMS` se ne esiste già una e in SSR non memorizza l'injector nella variabile globale `_platformInjector` (concorrenza tra richieste) (angular:packages/core/src/platform/platform.ts:L36-L54); `getPlatform()` lato server restituisce sempre `null` (angular:packages/core/src/platform/platform.ts:L137-L143).

- Il bootstrap standalone usa `createOrReusePlatformInjector`, che riusa `_platformInjector` o lo crea ed esegue i `PLATFORM_INITIALIZER` "avoid referencing `PlatformRef` class" (angular:packages/core/src/platform/platform.ts:L157-L181).
- `platformCore` è la piattaforma base; `platformBrowser` la estende con `INTERNAL_BROWSER_PLATFORM_PROVIDERS` (angular:packages/core/src/platform/platform_core_providers.ts:L19-L20, angular:packages/platform-browser/src/browser.ts:L231-L232), cioè `PLATFORM_ID` browser, un `PLATFORM_INITIALIZER` che installa `BrowserDomAdapter` e `DOCUMENT` (angular:packages/platform-browser/src/browser.ts:L205-L223).
- `PlatformRef.destroy()` distrugge i moduli registrati, chiama i listener e i `PLATFORM_DESTROY_LISTENERS`; una seconda chiamata lancia `PLATFORM_ALREADY_DESTROYED` (angular:packages/core/src/platform/platform_ref.ts:L123-L140).

3.7 Percorso standalone: `bootstrapApplication`

1. `bootstrapApplication(rootComponent, options?, context?)` combina il root component con `createProvidersConfig(options, context)`, in JIT risolve le risorse dei componenti e delega a `internalCreateApplication` (angular:packages/platform-browser/src/browser.ts:L125-L140). Gli errori di `resolveJitResources` sono solo loggati (angular:packages/platform-browser/src/browser.ts:L177-L185).
2. `createProvidersConfig` mette `BROWSER_MODULE_PROVIDERS` **prima** dei provider dell'utente e usa `context?.platformRef` se fornito (angular:packages/platform-browser/src/browser.ts:L168-L174). `BROWSER_MODULE_PROVIDERS` contiene `INJECTOR_SCOPE='root'`, `ErrorHandler`, i plugin eventi, `DomRendererFactory2`, `SharedStylesHost`, `EventManager`, `RendererFactory2`, `XhrFactory` e in dev un marker (angular:packages/platform-browser/src/browser.ts:L259-L276). `Testability` **non** è incluso: si aggiunge con `provideProtractorTestingSupport()` (angular:packages/platform-browser/src/browser.ts:L104-L113, angular:packages/platform-browser/src/browser.ts:L198-L203).
3. `internalCreateApplication`: lato server senza `platformRef` lancia `PLATFORM_NOT_FOUND` (angular:packages/core/src/application/create_application.ts:L47-L54); poi prende `platformRef.injector` o `createOrReusePlatformInjector(...)` e, in dev, verifica che il root component sia standalone (angular:packages/core/src/application/create_application.ts:L56-L62).
4. L'environment injector dell'app riceve, nell'ordine: `provideZonelessChangeDetectionInternal()`, `errorHandlerEnvironmentInitializer`, (dev) `validAppIdInitializer`, poi gli `appProviders` (angular:packages/core/src/application/create_application.ts:L64-L71). In v21 il default è quindi **zoneless**: `NgZone` → `NoopNgZone`, `ZONELESS_ENABLED = true` (angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L388-L394). Un `provideZoneChangeDetection(options)` nei provider utente ridefinisce `ZONELESS_ENABLED = false` e `NgZone`, e marca `PROVIDED_NG_ZONE = true` (angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L153-L167, angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L94-L95); arrivando dopo, vince.
5. L'injector viene creato da un `EnvironmentNgModuleRefAdapter` con debug name `'Environment Injector'` e `runEnvironmentInitializers: false`, perché gli initializer devono girare dentro la `NgZone` (angular:packages/core/src/application/create_application.ts:L72-L85). L'adapter è un `NgModuleRef<null>` (`instance = null`) che crea un `R3Injector` con scope `'environment'` e registra se stesso come `NgModuleRef` (angular:packages/core/src/render3/ng_module_ref.ts:L145-L172).
6. Si passa a `bootstrap({r3Injector, platformInjector, rootComponent})` (§3.9).

`createApplication(options?, context?)` fa lo stesso senza bootstrappare componenti (angular:packages/platform-browser/src/browser.ts:L142-L166). `BootstrapContext` contiene solo `platformRef`, pensato per casi

come l'SSR in cui la piattaforma è creata per richiesta (angular:packages/platform-browser/src/browser.ts:L52-L63, angular:packages/platform-browser/src/browser.ts:L118-L120).

3.8 Percorso NgModule: `platformBrowser().bootstrapModule(AppModule)`

È il percorso legacy, quello che lo schematic Spartacus converte (§3.16).

1. `PlatformRef.bootstrapModule(moduleType, compilerOptions)` riduce le opzioni, abilita l'implementazione di bootstrap per moduli (`setModuleBootstrapImpl`) e passa da `compileNgModuleFactory` a `bootstrapModuleFactory` (angular:packages/core/src/platform/platform_ref.ts:L87-L102). In AOT `compileNgModuleFactory` restituisce subito `new R3NgModuleFactory(moduleType)` (angular:packages/core/src/application/application_ngmodule_factory_compiler.ts:L22-L34).
2. `bootstrapModuleFactory` (deprecato come API pubblica con factory) costruisce gli app provider — zoneless di default + `applicationProviders` + error handler initializer — e crea l'`NgModuleRef` senza eseguire gli initializer (angular:packages/core/src/platform/platform_ref.ts:L41-L69). Le `BootstrapOptions` (`ngZone`, `ngZoneEventCoalescing`, `ngZoneRunCoalescing`) sono `@deprecated 20.2` (angular:packages/core/src/application/application_ref.ts:L93-L99).
3. `NgModuleRef` legge `emod.bootstrap` in `_bootstrapComponents` e crea l'injector con `createInjectorWithoutInjectorInstances(ngModuleType, ...)` scope `'environment'` (angular:packages/core/src/render3/ng_module_ref.ts:L67-L95); `createNgModuleRefWithProviders` lo crea con `runInjectorInitializers = false` (angular:packages/core/src/render3/ng_module_ref.ts:L127-L143).
4. `bootstrap({moduleRef, ...})` (§3.9) chiama alla fine `_moduleDoBootstrap`: avvia i componenti di `@NgModule.bootstrap`, altrimenti `ngDoBootstrap(appRef)`, altrimenti lancia `BOOTSTRAP_COMPONENTS_NOT_FOUND`; la funzione è separata per poter essere eliminata dal tree-shaking nelle app standalone (angular:packages/core/src/platform/bootstrap.ts:L183-L215).

`NgModuleRef.resolveInjectorInitializers()` esegue gli environment initializer e poi ottiene l'istanza del modulo (angular:packages/core/src/render3/ng_module_ref.ts:L105-L108).

3.9 `bootstrap()`: il tratto comune

1. `isApplicationBootstrapConfig` distingue i percorsi dall'assenza di `moduleRef` (angular:packages/core/src/platform/bootstrap.ts:L69-L73).
2. Dentro `ngZone.run` esegue prima gli environment initializer dell'injector dell'app o del modulo: qui vengono istanziati eagerly tutti gli NgModule (angular:packages/core/src/platform/bootstrap.ts:L84-L93).
3. In dev avvisa con `PROVIDED_BOTH_ZONE_AND_ZONELESS` se sono forniti sia zone sia zoneless (angular:packages/core/src/platform/bootstrap.ts:L95-L105).
4. Registra un listener sulla piattaforma: alla sua `destroy()` viene distrutto l'environment injector (standalone) o il `moduleRef` (angular:packages/core/src/platform/bootstrap.ts:L114-L135).
5. Aggiunge un pending task, esegue `ApplicationInitStatus.runInitializers()` e attende `donePromise`, poi imposta `LOCALE_ID` (angular:packages/core/src/platform/bootstrap.ts:L137-L147). Il dettaglio di questa fase, che per Spartacus è la più importante, è in 10 §3.5.
6. Se `ENABLE_ROOT_COMPONENT_BOOTSTRAP` vale `false` (lo usa `@angular/ssr` nell'estrazione delle rotte) salta il root component (angular:packages/core/src/platform/bootstrap.ts:L31-L53, angular:packages/core/src/platform/bootstrap.ts:L149-L160).
7. Standalone: `appRef = envInjector.get(ApplicationRef)` e `appRef.bootstrap(rootComponent)`; NgModule: `moduleBootstrapImpl` (angular:packages/core/src/platform/bootstrap.ts:L167-L176). Il pending task è rimosso nel `finally` (angular:packages/core/src/platform/bootstrap.ts:L178-L178).

8. Errori sincroni o Promise rifiutata finiscono all'error handler interno (fuori zona) e sono rilanciati (angular:packages/core/src/platform/bootstrap.ts:L217-L238).

3.10 `ApplicationConfig`, `ApplicationRef`, `ApplicationModule`

- `ApplicationConfig` ha un solo campo, `providers: Array<Provider | EnvironmentProviders>` (angular:packages/core/src/application/application_config.ts:L16-L21). `mergeApplicationConfig(...configs)` concatena i `providers` da sinistra a destra (angular:packages/core/src/application/application_config.ts:L31-L38): chi viene dopo, a parità di token, vince.
- `ApplicationRef` è `providedIn: 'root'` (angular:packages/core/src/application/application_ref.ts:L259-L260); il suo `injector` è l'`EnvironmentInjector` che l'ha creata (angular:packages/core/src/application/application_ref.ts:L359-L367).
- `isStable` deriva da `PendingTasksInternal.hasPendingTasksObservable`, `whenStable()` è una Promise risolta alla prima emissione `true` (angular:packages/core/src/application/application_ref.ts:L328-L357); il JSDoc avverte che l'app non diventa mai stabile se all'avvio si avvia un task ricorrente (angular:packages/core/src/application/application_ref.ts:L172-L179).
- `bootstrapImpl` lancia `ASYNC_INITIALIZERS_STILL_RUNNING` se gli initializer non sono conclusi (angular:packages/core/src/application/application_ref.ts:L508-L521); poi risolve la factory, la associa all'`NgModuleRef` dell'injector (nel percorso standalone è l'`EnvironmentNgModuleRefAdapter`), crea il componente sul selettore e registra la Testability se presente (angular:packages/core/src/application/application_ref.ts:L523-L548). Passare una `ComponentFactory` è deprecato (angular:packages/core/src/application/application_ref.ts:L445-L451).
- `_loadComponent` aggancia la host view, esegue un primo `tick()`, aggiunge il `ComponentRef` a `components` e invoca gli `APP_BOOTSTRAP_LISTENER` (angular:packages/core/src/application/application_ref.ts:L775-L795, token definito in angular:packages/core/src/application/application_ref.ts:L54-L66).
- `tick()` è non rientrante (`RECURSIVE_APPLICATION_REF_TICK`) e in dev esegue `checkNoChanges` (angular:packages/core/src/application/application_ref.ts:L570-L609); `synchronize()` ripete al massimo `MAXIMUM_REFRESH_RERUNS = 10` volte (angular:packages/core/src/application/application_ref.ts:L158-L159, angular:packages/core/src/application/application_ref.ts:L625-L649). Dettagli in 04.
- `ApplicationRef.destroy()` distrugge l'injector sottostante; doppia chiamata → `APPLICATION_REF_ALREADY_DESTROYED` (angular:packages/core/src/application/application_ref.ts:L834-L850).
- `ApplicationModule` è un NgModule vuoto il cui costruttore inietta `ApplicationRef` "to make it eager"; è riesportato da `BrowserModule` (angular:packages/core/src/application/application_module.ts:L13-L24).

3.11 `BrowserModule` e il marker anti-duplicazione

`BrowserModule` fornisce `BROWSER_MODULE_PROVIDERS` + `TESTABILITY_PROVIDERS` ed esporta `CommonModule` e `ApplicationModule` (angular:packages/platform-browser/src/browser.ts:L278-L291). In dev il costruttore cerca il marker con `{optional: true, skipSelf: true}` e lancia `BROWSER_MODULE_ALREADY_LOADED` se i provider esistono già in un injector **antenato** (angular:packages/platform-browser/src/browser.ts:L292-L307). Il marker esiste perché il bootstrap standalone importa i provider di `BrowserModule` senza referenziare il modulo (angular:packages/platform-browser/src/browser.ts:L234-L242). Il controllo usa `skipSelf`: due copie nello **stesso** injector non vengono rilevate (§9, T5).

3.12 Spartacus: gli entry point dell'app demo

- `project.json` dichiara `browser: main.ts` con polyfill `zone.js`, `server: main.server.ts` e `ssr.entry: server.ts` (projects/storefrontapp/project.json:L22-L23, projects/storefrontapp/project.json:L142-

L145).

- `main.ts` chiama `bootstrapApplication(AppComponent, appConfig)` dopo `enableProdMode()` se `environment.production` e logga l'errore con `console.error` (projects/storefrontapp/src/main.ts:L13-L23).
- `AppComponent` è standalone (nessun `standalone: false`), selettore `app-root`, template `<cx-storefront />`, `imports: [StorefrontComponent]` (projects/storefrontapp/src/app/app.component.ts:L10-L19).
- `appConfig` fornisce `provideHttpClient(withFetch(), withInterceptorsFromDi())`, `provideClientHydration(withEventReplay(), withNoHttpTransferCache())`, `provideZoneChangeDetection({eventCoalescing: true})`, `provideBrowserGlobalErrorListeners()` e infine `importProvidersFrom(AppModule)` (projects/storefrontapp/src/app/app.config.ts:L26-L35). Il ruolo di `withInterceptorsFromDi()` per gli `HTTP_INTERCEPTORS` multi registrati dai moduli Spartacus DA VERIFICARE: implementazione non letta in questo capitolo (la catena di interceptor è trattata in 06 — HttpClient e interceptor).
- Server: `main.server.ts` esporta di default `bootstrap = (context: BootstrapContext) => bootstrapApplication(AppComponent, config, context)` (projects/storefrontapp/src/main.server.ts:L23-L26); `server.ts` la passa a `ngExpressEngine({bootstrap, ...})` (projects/storefrontapp/src/server.ts:L20-L20, projects/storefrontapp/src/server.ts:L53-L54). Il `BootstrapContext` per richiesta lo costruisce `renderApplication` di `@angular/platform-server`, che crea una piattaforma server e chiama `bootstrap({platformRef})` (angular:packages/platform-server/src/utils.ts:L359-L362); `CommonEngine` di `@angular/ssr` invoca `renderApplication` quando riceve una funzione di bootstrap (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L203-L203). Dettagli del motore in 07 e 39 — SSR ottimizzato.
- `config` server = `mergeApplicationConfig(appConfig, serverConfig)`, con `serverConfig` che aggiunge `provideServerRendering()`, `importProvidersFrom(AppServerModule)` e `importProvidersFrom(TestConfigServerModule.forRoot())` ("DO NOT USE IN CUSTOMERS APPS") (projects/storefrontapp/src/app/app.config.server.ts:L17-L30). `AppServerModule` contiene solo `providers: [...provideServer({serverRequestOrigin: process.env['SERVER_REQUEST_ORIGIN']})]`, e `provideServer` restituisce `Provider[]` (projects/storefrontapp/src/app/app.module.server.ts:L10-L17, core-libs/setup/ssr/providers/ssr-providers.ts:L27-L27).

3.13 Spartacus: `AppModule` come ponte

`AppModule` non ha `declarations` né `bootstrap`: è puro "contenitore di provider". Importa `BrowserModule`, `StoreModule.forRoot({})`, `AppRoutingModule`, `EffectsModule.forRoot([])`, `SpartacusModule` e fornisce `privateProviders` (projects/storefrontapp/src/app/app.module.ts:L15-L25).

- `SpartacusModule` importa `BaseStorefrontModule`, `SpartacusFeaturesModule`, `SpartacusConfigurationModule` e ri-esporta `BaseStorefrontModule` (projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L20).
- `SpartacusConfigurationModule` è un NgModule di soli `providers` (`provideConfigFactory(layoutConfigFactory)`, `provideConfig(mediaConfig)`, `defaultCmsContentProviders`, PWA, i18n, provider di canale) e chiama `registerLocaleData` a livello di modulo ES (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L29-L72). I provider B2C/B2B sono `EnvironmentProviders` creati con `makeEnvironmentProviders([...provideConfig(...)])` (projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L25-L40, projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L19-L28); la scelta avviene alla valutazione del modulo ES via `environment.b2b` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L33-L42).
- `SpartacusFeaturesModule` costruisce a livello di modulo ES un array `featureModules` condizionato da flag `environment.*` e lo spalma in `imports` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L115-L175, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L285-

L285); mescola `forRoot()` (`AuthModule.forRoot()`, `AnonymousConsentsModule.forRoot()`, `ProductModule.forRoot()`, `ExternalRoutesModule.forRoot()`) con import diretti (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L176-L246).

- Ogni feature module della demo importa il `*RootModule` della feature e registra i moduli lazy con `provideConfig({featureModules: {...}})`, es. `CartBaseFeatureModule` (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L35). Il caricamento lazy vero e proprio è in 12.
- `privateProviders` è un `makeEnvironmentProviders` con `provideConfig(...)` e un `importProvidersFrom(TestOutletModule, TestConfigModule.forRoot({cookie: 'cxConfigE2E'}), ... StoreDevtoolsModule.instrument())` annidato (projects/storefrontapp/src/app/private/private.providers.ts:L19-L57): un `importProvidersFrom` dentro un `makeEnvironmentProviders` dentro `@NgModule.providers` di un modulo a sua volta passato a `importProvidersFrom`.
- `AppRoutingModule` (esportato da `@spartacus/storefront`) importa `RouterModule.forRoot([], {anchorScrolling: 'enabled', initialNavigation: 'enabledBlocking'})` con route vuote (core-libs/storefront/router/app-routing.module.ts:L13-L30). Le route vere sono generate da Spartacus: vedi 05 — Router e 14 — Routing.

3.14 Spartacus: `BaseCoreModule`, `BaseStorefrontModule` e i `forRoot()`

`BaseCoreModule` importa 16 `forRoot()` più `FederatedLoginModule`, in quest'ordine: ErrorHandling, State, Config, ConfigInitializer, ConfigValidator, I18n, Cms, GlobalMessage, Process, FeaturesConfig, SiteContext, MetaTagConfig, BaseOcc, LazyLoading, Http, SiteTheme (core-libs/core/src/base-core.module.ts:L26-L46). Il suo `forRoot()` restituisce `{ngModule: BaseCoreModule}` **senza** `providers` (core-libs/core/src/base-core.module.ts:L47-L53): per §3.4 è equivalente a importare il tipo.

Due commenti dichiarano vincoli d'ordine (core-libs/core/src/base-core.module.ts:L28-L28, core-libs/core/src/base-core.module.ts:L38-L38):

- `ErrorHandlingModule` "before any other interceptor";
- `SiteContextModule` "after RouterModule.forRoot, because it overwrites UrlSerializer": infatti `SiteContextModule.forRoot()` include `siteContextParamsProviders`, che ridefinisce `UrlSerializer` con `useExisting: SiteContextUrlSerializer` (core-libs/core/src/site-context/site-context.module.ts:L47-L54, core-libs/core/src/site-context/providers/site-context-params-providers.ts:L16-L17), mentre `ROUTER_PROVIDERS` contiene `{provide: UrlSerializer, useClass: DefaultUrlSerializer}` (angular:packages/router/src/router_module.ts:L65-L71).

Le forme di `forRoot()` in Spartacus sono varie:

MODULO	FORMA	RIFERIMENTO
`BaseCoreModule`	`{ngModule}` senza `providers`	core-libs/core/src/base-core.module.ts:L47-L53
`ErrorHandlingModule`	sostituisce `ErrorHandler` con `CxErrorHandler`, aggiunge `provideMultiErrorHandler()`; il modulo importa altri due `forRoot()`	core-libs/core/src/error-handling/error-handling.module.ts:L13-L32
`ConfigModule`	`@NgModule({})` vuoto, costruttore che inietta `ConfigurationService`; tre factory `withConfig`, `withConfigFactory`, `forRoot(config = {})`	core-libs/core/src/config/config.module.ts:L13-L67
`StateModule`	`@NgModule({})`, `forRoot()` porta solo `stateMetaReducers`	core-libs/core/src/state/state.module.ts:L10-L18

MODULO	FORMA	RIFERIMENTO
`ProcessModule`	`providers: []` (array vuoto)	`core-libs/core/src/process/process.module.ts:L10-L20`
`CmsLcpModule`	`forRoot()` senza tipo di ritorno dichiarato	`core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.module.ts:L11-L19`
`OutletModule`	`forRoot()` con `APP_INITIALIZER`, `forChild()` con `MODULE_INITIALIZER` (per moduli lazy)	`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L49-L89`
`RoutingModule` (storefront)	importa `CoreRoutingModule.forRoot()`; `forRoot()` aggiunge config rotte e due `BEFORE_CMS_PAGE_GUARD` multi	`core-libs/storefront/cms-structure/routing/routing.module.ts:L18-L40`

La coppia `forRoot()`/`forChild()` di `OutletModule` mostra perché Spartacus ha bisogno di un proprio `MODULE_INITIALIZER`: gli `APP_INITIALIZER` girano una volta sola al bootstrap, e un modulo lazy ne richiede l'equivalente (vedi 03 e 17 — Outlet).

`BaseStorefrontModule` importa `BaseCoreModule.forRoot()`, `RouterModule` (senza `forRoot`), i moduli UI e i `forRoot()` di `PageComponentModule`, `RoutingModule`, `MediaModule`, `OutletModule`, `CmsLcpModule`; esporta `LayoutModule` e `StorefrontComponentModule` (`core-libs/storefront/base-storefront.module.ts:L26-L49`). Entrambi gli aggregatori sono API pubblica (`core-libs/core/public_api.ts:L12-L12`, `core-libs/storefront/public_api.ts:L20-L20`).

3.15 Spartacus: componenti standalone dentro NgModule

- `StorefrontComponent` (`cx-storefront`) è standalone e importa direttamente `RouterModule` e altri componenti/direttive standalone (`core-libs/storefront/layout/main/storefront.component.ts:L51-L67`).
- `StorefrontComponentModule` mette il componente standalone negli `imports` (non in `declarations`) e lo riesporta: l'NgModule resta come **shim di compatibilità** per chi importa ancora moduli (`core-libs/storefront/layout/main/storefront-component.module.ts:L19-L34`). Lo stesso fa `OutletModule` con la direttiva standalone `OutletDirective` (`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L49-L52`).
- La guida di migrazione dice "standalone is now the default" (`docs/migration/221121_7/modernize-app-to-standalone-bootstrap-application.md:L126-L126`).
- Le feature lazy di Spartacus non usano `loadComponent` o `createEnvironmentInjector` ma `NgModuleFactory`: `resolveModuleFactory` usa la factory se già tale, altrimenti `compiler.compileModuleAsync(module)`, poi `moduleFactory.create(parentInjector)` (`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L98`, `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L236`). Il motivo per cui gli NgModule restano centrali in Spartacus è proprio questo: l'unità di caricamento lazy è un NgModule (03).

3.16 Spartacus: lo schematic `modernize-app-to-standalone-bootstrap-application`

Il dettaglio delle regole è nel capitolo 43; qui il collegamento con i concetti del capitolo.

- È registrato in `collection.json`, non in `migrations.json`: va lanciato esplicitamente, non parte con `ng update` (`core-libs/schematics/src/collection.json:L49-L52`). Si lancia con `ng g @spartacus/schematics:modernize-app-to-standalone-bootstrap-application` (`docs/migration/221121_7/modernize-app-to-standalone-bootstrap-application.md:L16-L16`).
- `migrate()` concatena 5 regole comuni (AppComponent, AppModule, app.config, main.ts, angular.json) e, se `isSsrUsed(tree)`, 5 regole SSR, ognuna avvolta in `withFallbackDocsForStandaloneBootstrap` (`core-`

libs/schematics/src/modernize-app-to-standalone-bootstrap-application/index.ts:L26-L54), che cattura l'eccezione, logga un link alla guida e restituisce il tree invariato (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L28-L43).

- Mappatura sui concetti di questo capitolo:
  - `updateAppComponent` rende `AppComponent` standalone (rimuove `standalone`, aggiunge `StorefrontComponent` a `imports`) — §3.5 (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-app-component.ts:L22-L66);
  - `updateAppModule` toglie da `AppModule` `bootstrap`, `declarations`, i provider funzionali e `BrowserModule` — §3.11 (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-app-module.ts:L16-L67, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-app-module.ts:L95-L99);
  - `createAppConfig` genera `app.config.ts` con `provideBrowserGlobalErrorListeners()`, `provideZoneChangeDetection({eventCoalescing: true})`, `provideHttpClient(withFetch()`, `withInterceptorsFromDi())`, `importProvidersFrom(AppModule)`, e lo salta se il file esiste (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/create-app-config.ts:L16-L51);
  - `updateMainTs` sostituisce `platformBrowser().bootstrapModule(...)` con `bootstrapApplication(AppComponent, appConfig)` — §3.7/§3.8 (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-main-ts.ts:L41-L67, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-main-ts.ts:L102-L111);
  - `updateAngularJson` cancella `standalone` dalle opzioni degli schematic `component|directive|pipe` (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-angular-json.ts:L11-L32, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-angular-json.ts:L48-L57);
  - SSR: `updateMainServerTs` genera la funzione `bootstrap(context)` e `createAppConfigServer` il `mergeApplicationConfig` (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/ssr/update-main-server-ts.ts:L97-L102, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/ssr/create-app-config-server.ts:L29-L41); `moveHydrationConfig`, `updateAppModuleServer`, `updateServerTs` completano il quadro (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/ssr/move-hydration-config.ts:L13-L14, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/ssr/update-app-module-server.ts:L12-L14, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/ssr/update-server-ts.ts:L12-L14).
- Coerentemente, `ng add` richiede un'app standalone: `detectAndSetupAppStructure` lancia `SchematicsException` se manca `app.config.ts` (core-libs/schematics/src/add-spartacus/index.ts:L317-L353), poi `createAppModule` crea un `AppModule` vuoto e `updateAppConfig` aggiunge `importProvidersFrom(AppModule)` (core-libs/schematics/src/add-spartacus/index.ts:L539-L544, core-libs/schematics/src/add-spartacus/create-app-module.ts:L20-L67, core-libs/schematics/src/add-spartacus/update-app-config.ts:L19-L46). Il risultato ha la stessa forma dell'app demo: root standalone + `AppModule` ponte.

3.17 L'ordine di appiattimento dell'app demo, passo per passo

Applichiamo §3.4 a `importProvidersFrom(AppModule)`:

1. `walkProviderTree(AppModule)` — angular:packages/core/src/di/provider_collection.ts:L151-L182.
2. `BrowserModule` (tipo): visita i suoi import di injector (`CommonModule`, `ApplicationModule` da `exports`, §3.1), poi emette i 3 provider sintetici e `BROWSER_MODULE_PROVIDERS + TESTABILITY_PROVIDERS` — projects/

storefrontapp/src/app/app.module.ts:L16-L17, angular:packages/platform-browser/src/browser.ts:L287-L290, angular:packages/core/src/di/provider_collection.ts:L292-L354.

3. StoreModule.forRoot({}) (MWP con ngModule: StoreRootModule, deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1543-L1547): visita StoreRootModule, ma i providers del MWP sono **rinviati** alla fine degli import di AppModule — angular:packages/core/src/di/provider_collection.ts:L303-L321.
4. AppRoutingModule (tipo): visita RouterModule.forRoot(...); a fine import di AppRoutingModule emette i provider del router (incluso UrlSerializer → DefaultUrlSerializer), poi quelli di AppRoutingModule — core-libs/storefront/router/app-routing.module.ts:L13-L30, angular:packages/router/src/router_module.ts:L69-L71.
5. EffectsModule.forRoot([]) (MWP con ngModule: EffectsRootModule, deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L485-L489): rinviato come al passo 3.
6. SpartacusModule → BaseStorefrontModule → BaseCoreModule.forRoot() (equivale al tipo) → 16 MWP visitati; a fine import di BaseCoreModule sono emessi i loro providers in ordine dichiarato, incluso UrlSerializer → SiteContextUrlSerializer — core-libs/storefront/base-storefront.module.ts:L28-L28, core-libs/core/src/base-core.module.ts:L27-L45, angular:packages/core/src/di/provider_collection.ts:L316-L321.
7. Resto di BaseStorefrontModule: moduli UI, poi a fine import i provider di PageComponentModule.forRoot(), RoutingModule.forRoot(), MediaModule.forRoot(), OutletModule.forRoot() (anche se OutletModule era già stato visitato come tipo poco prima, nello stesso BaseStorefrontModule: core-libs/storefront/base-storefront.module.ts:L31-L31 e core-libs/storefront/base-storefront.module.ts:L43-L43), CmsLcpModule.forRoot() — core-libs/storefront/base-storefront.module.ts:L36-L44, angular:packages/core/src/di/provider_collection.ts:L360-L362.
8. SpartacusFeaturesModule, SpartacusConfigurationModule — projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L13-L17.
9. Fine import di AppModule: i provider rinviati di StoreModule.forRoot({}) ed EffectsModule.forRoot([]) — angular:packages/core/src/di/provider_collection.ts:L319-L321.
10. Provider sintetici di AppModule, poi privateProviders (ultimi della lista) — angular:packages/core/src/di/provider_collection.ts:L324-L354, projects/storefrontapp/src/app/app.module.ts:L23-L23.

Risultato: UrlSerializer è registrato due volte (passi 4 e 6) e vince SiteContextUrlSerializer, come richiede il commento di BaseCoreModule (core-libs/core/src/base-core.module.ts:L38-L38). E privateProviders, con TestConfigModule.forRoot che "should be imported after other config modules" (projects/storefrontapp/src/app/private/private.providers.ts:L54-L54), è davvero in coda.

---

4. Diagrammi

4.1 Sequenza di bootstrapApplication nell'app demo (browser)

■ TAVOLA 02.1 diagramma a tutta pagina, in fondo al capitolo

Ogni passo: §3.3 (riga 1), §3.7 (righe 2-8), §3.9 (righe 9-12 e 15), §3.5 e §3.10 (righe 13-14).

4.2 walkProviderTree per un singolo contenitore

■ TAVOLA 02.2 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: angular:packages/core/src/di/provider_collection.ts:L245-L362 (dettaglio in §3.4).

4.3 I due percorsi di bootstrap convergono

■ TAVOLA 02.3 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: §3.7, §3.8, §3.9, §3.16.

4.4 Albero dei moduli dell'app demo e ordine di emissione

■ TAVOLA 02.4 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: §3.13, §3.14, §3.17 (i numeri nei nodi sono i passi del §3.17); projects/storefrontapp/src/app/app.module.ts:L16-L23.

---

5. Estratti di codice reale

5.1 L'entry point standalone

projects/storefrontapp/src/main.ts:L13-L23

```ts
if (environment.production) {
  enableProdMode();
}

bootstrapApplication(AppComponent, appConfig)
  /* eslint-disable-next-line no-console
  --
  It's just an example application file. This message is not crucial
  to be logged using any special logger. Moreover, we don't have
  any special logger available in this context. */
  .catch((err) => console.error(err));
```

5.2 `ApplicationConfig` dell'app demo: il ponte `importProvidersFrom(AppModule)`

projects/storefrontapp/src/app/app.config.ts:L26-L35

```ts
export const appConfig: ApplicationConfig = {
  providers: [
    provideHttpClient(withFetch(), withInterceptorsFromDi()),
    provideClientHydration(withEventReplay(), withNoHttpTransferCache()),
    provideZoneChangeDetection({ eventCoalescing: true }),
    provideBrowserGlobalErrorListeners(),
```

```
    importProvidersFrom(AppModule),
  ],
};
```

5.3 Root component standalone

projects/storefrontapp/src/app/app.component.ts:L10-L19

```ts
/**
 * Root component that belongs to the example app, not Spartacus libraries.
 * In customers' applications, the analogical root AppComponent belongs to the custom app.
 */
@Component({
  selector: 'app-root',
  template: `<cx-storefront />`,
  imports: [StorefrontComponent],
})
export class AppComponent {}
```

5.4 AppModule: nessun declarations, nessun bootstrap

projects/storefrontapp/src/app/app.module.ts:L15-L25

```ts
@NgModule({
  imports: [
    BrowserModule,
    StoreModule.forRoot({}),
    AppRoutingModule,
    EffectsModule.forRoot([]),
    SpartacusModule,
  ],
  providers: [privateProviders],
})
export class AppModule {}
```

5.5 Entry SSR: una funzione che riceve il BootstrapContext

projects/storefrontapp/src/main.server.ts:L23-L26

```ts
const bootstrap = (context: BootstrapContext) =>
  bootstrapApplication(AppComponent, config, context);

export default bootstrap;
```

projects/storefrontapp/src/app/app.config.server.ts:L17-L30

```ts
const serverConfig: ApplicationConfig = {
  providers: [
    provideServerRendering(),

    importProvidersFrom(AppServerModule),

    importProvidersFrom(
      // DO NOT USE IN CUSTOMERS APPS:
      TestConfigServerModule.forRoot() // Injects config dynamically from e2e tests for SSR.
    ),
  ],
};

export const config = mergeApplicationConfig(appConfig, serverConfig);
```

5.6 Angular: `bootstrapApplication` e `createProvidersConfig`

angular:packages/platform-browser/src/browser.ts:L125-L140

```ts
export async function bootstrapApplication(
  rootComponent: Type<unknown>,
  options?: ApplicationConfig,
  context?: BootstrapContext,
): Promise<ApplicationRef> {
  const config = {
    rootComponent,
    ...createProvidersConfig(options, context),
  };

  if ((typeof ngJitMode === 'undefined' || ngJitMode) && typeof fetch === 'function') {
    await resolveJitResources();
  }

  return internalCreateApplication(config);
}
```

angular:packages/platform-browser/src/browser.ts:L168-L174

```ts
function createProvidersConfig(options?: ApplicationConfig, context?: BootstrapContext) {
  return {
    platformRef: context?.platformRef,
    appProviders: [...BROWSER_MODULE_PROVIDERS, ...(options?.providers ?? [])],
    platformProviders: INTERNAL_BROWSER_PLATFORM_PROVIDERS,
  };
}
```

5.7 Angular: l'environment injector dell'app (default zoneless)

angular:packages/core/src/application/create_application.ts:L56-L85

TS

```ts
try {
  const platformInjector =
    platformRef?.injector ?? createOrReusePlatformInjector(platformProviders as StaticProvider[]);

  if ((typeof ngDevMode === 'undefined' || ngDevMode) && rootComponent !== undefined) {
    assertStandaloneComponentType(rootComponent);
  }

  // Create root application injector based on a set of providers configured at the platform
  // bootstrap level as well as providers passed to the bootstrap call by a user.
  const allAppProviders = [
    provideZonelessChangeDetectionInternal(),
    errorHandlerEnvironmentInitializer,
    ...(ngDevMode ? [validAppIdInitializer] : []),
    ...(appProviders || []),
  ];
  const adapter = new EnvironmentNgModuleRefAdapter({
    providers: allAppProviders,
    parent: platformInjector as EnvironmentInjector,
    debugName: typeof ngDevMode === 'undefined' || ngDevMode ? 'Environment Injector' : '',
    // We skip environment initializers because we need to run them inside the NgZone, which
    // happens after we get the NgZone instance from the Injector.
    runEnvironmentInitializers: false,
  });

  return bootstrap({
    r3Injector: adapter.injector,
    platformInjector,
    rootComponent,
  });
```

5.8 Angular: `bootstrap()` e la biforcazione finale

angular:packages/core/src/platform/bootstrap.ts:L84-L93

TS

```ts
const envInjector = isApplicationBootstrapConfig(config)
  ? config.r3Injector
  : config.moduleRef.injector;
const ngZone = envInjector.get(NgZone);
return ngZone.run(() => {
  if (isApplicationBootstrapConfig(config)) {
    config.r3Injector.resolveInjectorInitializers();
  } else {
    config.moduleRef.resolveInjectorInitializers();
  }
```

angular:packages/core/src/platform/bootstrap.ts:L167-L176

```ts
      if (isApplicationBootstrapConfig(config)) {
        const appRef = envInjector.get(ApplicationRef);
        if (config.rootComponent !== undefined) {
          appRef.bootstrap(config.rootComponent);
        }
        return appRef;
      } else {
        moduleBootstrapImpl?.(config.moduleRef, config.allPlatformModules);
        return config.moduleRef;
      }
```

5.9 Angular: `importProvidersFrom`

angular:packages/core/src/di/provider_collection.ts:L144-L149

```ts
export function importProvidersFrom(...sources: ImportProvidersSource[]): EnvironmentProviders {
  return {
    ɵproviders: internalImportProvidersFrom(true, sources),
    ɵfromNgModule: true,
  } as InternalEnvironmentProviders;
}
```

5.10 Angular: `walkProviderTree`, ramo NgModule

angular:packages/core/src/di/provider_collection.ts:L292-L362

```ts
  } else if (injDef) {
    // First, include providers from any imports.
    if (injDef.imports != null && !isDuplicate) {
      // ...
      dedup.add(defType);

      let importTypesWithProviders: InjectorTypeWithProviders<any>[] | undefined;
      try {
        deepForEach(injDef.imports, (imported) => {
          if (walkProviderTree(imported, visitor, parents, dedup)) {
            importTypesWithProviders ||= [];
            // ...
            importTypesWithProviders.push(imported);
          }
        });
      } finally {
        // ...
      }

      // Imports which are declared with providers (TypeWithProviders) need to be processed
```

```
    // after all imported modules are processed. This is similar to how View Engine
    // processes/merges module imports in the metadata resolver. See: FW-1349.
    if (importTypesWithProviders !== undefined) {
      processInjectorTypesWithProviders(importTypesWithProviders, visitor);
    }
  }

  if (!isDuplicate) {
    // Track the InjectorType and add a provider for it.
    // It's important that this is done after the def's imports.
    const factory = getFactoryDef(defType) || (() => new defType!());
    // ...
    visitor({provide: defType, useFactory: factory, deps: EMPTY_ARRAY}, defType);
    // ...
    visitor({provide: INJECTOR_DEF_TYPES, useValue: defType, multi: true}, defType);
    // ...
    visitor(
      {provide: ENVIRONMENT_INITIALIZER, useValue: () => inject(defType!), multi: true},
      defType,
    );
  }

  // Next, include providers listed on the definition itself.
  const defProviders = injDef.providers as Array<SingleProvider | InternalEnvironmentProviders>;
  if (defProviders != null && !isDuplicate) {
    // ...
  }
} else {
  // Should not happen, but just in case.
  return false;
}

return (
  defType !== container && (container as InjectorTypeWithProviders<any>).providers !== undefined
);
```

5.11 Angular: l'injector `Standalone[...]`

angular:packages/core/src/render3/standalone_service.ts:L26-L47

```ts
getOrCreateStandaloneInjector(componentDef: ComponentDef<unknown>): EnvironmentInjector | null {
  if (!componentDef.standalone) {
    return null;
  }

  if (!this.cachedInjectors.has(componentDef)) {
    const providers = internalImportProvidersFrom(false, componentDef.type);
    const standaloneInjector =
```

```
        providers.length > 0
          ? createEnvironmentInjector(
              [providers],
              this._injector,
              typeof ngDevMode !== 'undefined' && ngDevMode
                ? `Standalone[${componentDef.type.name}]`
                : '',
            )
          : null;
      this.cachedInjectors.set(componentDef, standaloneInjector);
    }

    return this.cachedInjectors.get(componentDef)!;
  }
```

5.12 Angular: il pattern `forRoot` con guardia (`RouterModule`)

angular:packages/router/src/router_module.ts:L222-L233

TS

```
export function provideForRootGuard(): any {
  const router = inject(Router, {optional: true, skipSelf: true});

  if (router) {
    throw new RuntimeError(
      RuntimeErrorCode.FOR_ROOT_CALLED_TWICE,
      `The Router was provided more than once. This can happen if 'forRoot' is used outside of the root
injector.` +
        ` Lazy loaded modules should use RouterModule.forChild() instead.`,
    );
  }
  return 'guarded';
}
```

5.13 Spartacus: `BaseCoreModule`

core-libs/core/src/base-core.module.ts:L26-L53

TS

```
@NgModule({
  imports: [
    ErrorHandlingModule.forRoot(), // Import this module before any other interceptor to handle HTTP
errors efficiently
    StateModule.forRoot(),
    ConfigModule.forRoot(),
    ConfigInitializerModule.forRoot(),
    ConfigValidatorModule.forRoot(),
    I18nModule.forRoot(),
    CmsModule.forRoot(),
```

```
    GlobalMessageModule.forRoot(),
    ProcessModule.forRoot(),
    FeaturesConfigModule.forRoot(),
    SiteContextModule.forRoot(), // should be imported after RouterModule.forRoot, because it overwrites
UrlSerializer
    MetaTagConfigModule.forRoot(),
    BaseOccModule.forRoot(),
    LazyLoadingModule.forRoot(),
    HttpModule.forRoot(),
    SiteThemeModule.forRoot(),
    FederatedLoginModule,
  ],
})
export class BaseCoreModule {
  static forRoot(): ModuleWithProviders<BaseCoreModule> {
    return {
      ngModule: BaseCoreModule,
    };
  }
}
```

5.14 Spartacus: `BaseStorefrontModule`, tipo e `forRoot()` dello stesso modulo

core-libs/storefront/base-storefront.module.ts:L26-L49

TS

```
@NgModule({
  imports: [
    BaseCoreModule.forRoot(),
    RouterModule,
    GlobalMessageComponentModule,
    OutletModule,
    OutletRefModule,
    PwaModule,
    PageLayoutModule,
    SeoModule,
    PageComponentModule.forRoot(),
    PageSlotModule,
    SkipLinkModule,
    KeyboardFocusModule,
    LayoutModule,
    RoutingModule.forRoot(),
    MediaModule.forRoot(),
    OutletModule.forRoot(),
    CmsLcpModule.forRoot(),
    StorefrontComponentModule,
  ],
```

```
  exports: [LayoutModule, StorefrontComponentModule],
})
export class BaseStorefrontModule {}
```

5.15 Spartacus: shim NgModule per un componente standalone

core-libs/storefront/layout/main/storefront-component.module.ts:L19-L34

```ts
@NgModule({
  imports: [
    CommonModule,
    RouterModule,
    GlobalMessageComponentModule,
    OutletModule,
    OutletRefModule,
    PageLayoutModule,
    PageSlotModule,
    KeyboardFocusModule,
    SkipLinkModule,
    StorefrontComponent,
  ],
  exports: [StorefrontComponent],
})
export class StorefrontComponentModule {}
```

5.16 Spartacus: modulo "contenitore di servizio"

core-libs/core/src/config/config.module.ts:L13-L19

```ts
@NgModule({})
export class ConfigModule {
  // To make sure ConfigurationService will be instantiated, we inject it into
  // module constructor
  constructor(_configurationService: ConfigurationService) {
    // Intentional empty constructor
  }
```

5.17 Schematic: il riconoscimento testuale di `platformBrowser()`

core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-main-ts.ts:L102-L111

```ts
      if (
        propAccess &&
        propAccess.getName() === 'bootstrapModule' &&
        propAccess.getExpression().getText() === 'platformBrowser()'
      ) {
```

```
        // Replace this call expression with bootstrapApplication(AppComponent, appConfig)
        callExpr.replaceWithText(
          'bootstrapApplication(AppComponent, appConfig)'
        );
        break;
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`ApplicationConfig.providers`	`Array<Provider | EnvironmentProviders>`	obbligatorio	`angular:packages/core/src/application/application_config.ts:L16-L21`	`angular:packages/platform-browser/src/browser.ts:L171-L171`
`BootstrapContext.platformRef`	`PlatformRef`	`undefined` → piattaforma creata/riusata	`angular:packages/platform-browser/src/browser.ts:L58-L63`	`angular:packages/core/src/application/create_application.ts:L57-L58`
`ENABLE_ROOT_COMPONENT_BOOTSTRAP`	`InjectionToken<boolean>`	`true`	`angular:packages/core/src/platform/bootstrap.ts:L51-L53`	`angular:packages/core/src/platform/bootstrap.ts:L149-L152`
`APP_BOOTSTRAP_LISTENER`	multi `InjectionToken<(compRef) => void>`	`[]`	`angular:packages/core/src/application/application_ref.ts:L64-L66`	`angular:packages/core/src/application/application_ref.ts:L784-L794`
`BootstrapOptions` (`ngZone`, `ngZoneEventCoalescing`, `ngZoneRunCoalescing`)	opzioni di `bootstrapModule` (deprecate 20.2)	—	`angular:packages/core/src/application/application_ref.ts:L93-L156`	nessun punto: fuse da `optionsReducer` (`angular:packages/core/src/platform/platform_ref.ts:L97-L97`) e passate a `bootstrapModuleFactory`, che legge solo `applicationProviders` (`angular:packages/core/src/platform/platform_ref.ts:L51-L56`), quindi ignorate a runtime; Angular fornisce la migrazione `bootstrap-options-migration` (`angular:packages/core/schematics/migrations/bootstrap-options-migration/migration.ts:L40-L41`)
`NgModule.id`	`string`	`undefined` (non registrato)	`angular:packages/core/src/metadata/ng_module.ts:L189-L193`	`registerNgModuleType` (`angular:packages/core/src/linker/ng_module_registration.ts:L46-L46`), chiamata in JIT da `angular:packages/core/src/render3/jit/module.ts:L109-L109`
`Component.standalone`	`boolean`	`true` (v19+)	`angular:packages/core/src/metadata/directives.ts:L627-L630`	`angular:packages/compiler-cli/src/ngtsc/core/src/compiler.ts:L480-L482`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`environment.production` (demo)	`boolean`	per environment	`projects/storefrontapp/src/environments/*` (**DA VERIFICARE**: non letto)	`projects/storefrontapp/src/main.ts:L13-L15`, `projects/storefrontapp/src/app/private/private.providers.ts:L55-L55`
`environment.b2b` e altri flag feature (demo)	`boolean`	per build	`projects/storefrontapp/src/environments/*` (**DA VERIFICARE**: non letto)	`projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L40-L42`, `projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L117-L175`
`SERVER_REQUEST_ORIGIN` (variabile d'ambiente)	`string`	—	—	`projects/storefrontapp/src/app/app.module.server.ts:L13-L13`
`TestConfigModule.forRoot({cookie})`	`TestConfigModuleOptions`	—	`core-libs/core/src/config/test-config.module.ts:L83-L92`	`projects/storefrontapp/src/app/private/private.providers.ts:L54-L54`

Per la configurazione di Spartacus vera e propria (`provideConfig`, `Config`, scope) vedi 11 — Configurazione.

---

7. Commenti e note notevoli dal sorgente

- `angular:packages/core/src/di/provider_collection.ts:L316-L318` — "Imports which are declared with providers (TypeWithProviders) need to be processed after all imported modules are processed. This is similar to how View Engine processes/merges module imports in the metadata resolver. See: FW-1349." È la regola che rende l'ordine dei provider diverso dall'ordine testuale degli `imports`.
- `angular:packages/core/src/di/provider_collection.ts:L325-L326` — "Track the InjectorType and add a provider for it. It's important that this is done after the def's imports."
- `angular:packages/core/src/di/provider_collection.ts:L61-L61` — "Previously, this was achieved using the `ENVIRONMENT_INITIALIZER` token which is now deprecated."
- `angular:packages/core/src/di/interface/provider.ts:L389-L389` — `ModuleWithProviders`: "Usage without a generic type is deprecated."
- `angular:packages/core/src/application/create_application.ts:L76-L77` — "We skip environment initializers because we need to run them inside the NgZone, which happens after we get the NgZone instance from the Injector."
- `angular:packages/core/src/platform/bootstrap.ts:L183-L186` — `moduleBootstrapImpl` separato "to tree shake the module based bootstrap implementation in standalone apps".
- `angular:packages/core/src/platform/platform_ref.ts:L44-L45` — `@deprecated` `bootstrapModuleFactory` con factory.
- `angular:packages/core/src/application/application_ref.ts:L97-L97` — `BootstrapOptions` `@deprecated 20.2`.
- `angular:packages/core/src/application/application_ref.ts:L445-L446` — `@deprecated` `ApplicationRef.bootstrap(ComponentFactory)`.
- `angular:packages/core/src/render3/ng_module_ref.ts:L58-L63` — ciclo `ApplicationRef -> ComponentFactoryResolver -> NgModuleRef` risolto creando il CFR a mano.
- `angular:packages/core/src/platform/platform.ts:L47-L49` — in SSR l'injector globale non viene memorizzato per la concorrenza tra richieste (con un refuso "requjest" nel sorgente).

- angular:packages/router/src/router_module.ts:L65-L68 — TODO: ROUTER_PROVIDERS mantenuti nel modulo per non introdurre una dipendenza dall'ordine di import (proprio quella di cui dipende il commento UrlSerializer di Spartacus).
- core-libs/core/src/base-core.module.ts:L28-L28 e core-libs/core/src/base-core.module.ts:L38-L38 — vincoli d'ordine di import (interceptor; UrlSerializer).
- core-libs/core/src/config/config.module.ts:L15-L16 — "To make sure ConfigurationService will be instantiated, we inject it into module constructor".
- projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L180-L181 — LogoutModule/LoginRouteModule "will be come part of auth package".
- projects/storefrontapp/src/app/private/private.providers.ts:L14-L18 — "CAUTION: IT IS NOT MEANT FOR CUSTOMERS' APPLICATIONS!".
- projects/storefrontapp/src/app/private/private.providers.ts:L54-L54 — TestConfigModule.forRoot "Should be imported after other config modules."
- projects/storefrontapp/src/app/app.config.server.ts:L24-L25 — "DO NOT USE IN CUSTOMERS APPS".
- projects/storefrontapp/src/main.server.ts:L7-L9 — commento "Load $localize onto the global scope" senza alcun import corrispondente nel file: commento residuo.
- core-libs/core/src/config/test-config.module.ts:L81-L81 — "CAUTION: DON'T USE IT IN PRODUCTION! IT HASN'T BEEN REVIEWED FOR SECURITY ISSUES."

8. Extension point

EXTENSION POINT	TIPO	DEFINIZIONE	USO TIPICO IN UN PROGETTO SPARTACUS
ApplicationConfig / mergeApplicationConfig	interfaccia/funzione (@angular/core)	angular:packages/core/src/application/application_config.ts:L16-L38	aggiungere provider funzionali Angular in app.config.ts; comporre config server (projects/storefrontapp/src/app/app.config.server.ts:L30-L30)
importProvidersFrom	funzione (@angular/core)	angular:packages/core/src/di/provider_collection.ts:L144-L149	portare un NgModule (o forRoot()) di Spartacus o di terze parti in un'app standalone
makeEnvironmentProviders / provideEnvironmentInitializer	funzioni (@angular/core)	angular:packages/core/src/di/provider_collection.ts:L46-L89	raggruppare provider di configurazione come fa privateProviders (projects/storefrontapp/src/app/private/private.providers.ts:L19-L57)
ModuleWithProviders<T>	interfaccia (@angular/core)	angular:packages/core/src/di/interface/provider.ts:L393-L396	scrivere il forRoot() di un proprio modulo custom
BootstrapContext	interfaccia (@angular/platform-browser)	angular:packages/platform-browser/src/browser.ts:L58-L63	entry SSR (projects/storefrontapp/src/main.server.ts:L23-L24)
APP_BOOTSTRAP_LISTENER	multi token (@angular/core)	angular:packages/core/src/application/application_ref.ts:L64-L66	eseguire codice per ogni componente bootstrappato

EXTENSION POINT	TIPO	DEFINIZIONE	USO TIPICO IN UN PROGETTO SPARTACUS
`providePlatformInitializer` / `PLATFORM_INITIALIZER`	funzione/token (`@angular/core`)	`angular:packages/core/src/platform/platform.ts:L201-L207`	inizializzazione a livello di piattaforma
`NgModule.ngDoBootstrap(appRef)`	hook (solo percorso NgModule)	`angular:packages/core/src/platform/bootstrap.ts:L203-L204`	bootstrap manuale (non usato dall'app demo)
`BaseCoreModule` (+ `forRoot()`)	NgModule	`core-libs/core/src/base-core.module.ts:L26-L53` (export `core-libs/core/public_api.ts:L12-L12`)	importato indirettamente via `BaseStorefrontModule`
`BaseStorefrontModule`	NgModule	`core-libs/storefront/base-storefront.module.ts:L26-L49` (export `core-libs/storefront/public_api.ts:L20-L20`)	aggregatore da importare nello `SpartacusModule` dell'app
`AppRoutingModule`	NgModule	`core-libs/storefront/router/app-routing.module.ts:L13-L30`	importato in `AppModule` prima di `SpartacusModule` (§9, T16)
`StorefrontComponent`	componente standalone	`core-libs/storefront/layout/main/storefront.component.ts:L51-L67`	negli `imports` del root component
`ConfigModule.withConfig` / `withConfigFactory` / `forRoot`	factory `ModuleWithProviders`	`core-libs/core/src/config/config.module.ts:L28-L66`	alternativa NgModule a `provideConfig`
`OutletModule.forRoot()` / `forChild()`	factory `ModuleWithProviders`	`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88`	`forChild()` nei moduli lazy
schematic `modernize-app-to-standalone-bootstrap-application`	schematic	`core-libs/schematics/src/collection.json:L49-L52`	migrare un'app Spartacus da `bootstrapModule` a `bootstrapApplication`

I tre simboli sono importabili dagli entry point pubblici, come mostrano gli usi nel repo: `import { AppRoutingModule } from '@spartacus/storefront';` (`projects/storefrontapp/src/app/app.module.ts:L11-L11`), `import { ConfigModule } from '@spartacus/core';` (`projects/storefrontapp/src/test-outlets/test-outlet.module.ts:L11-L11`), `import { Card, CardComponent, OutletModule } from '@spartacus/storefront';` (`feature-libs/order/components/order-details/order-detail-billing/order-detail-billing.component.ts:L16-L16`). Per la distinzione tra API pubblica e interna vedi 54 — API pubblica vs interna.

Ricette correlate nella Parte III:

- override di configurazione e di servizi via DI, dove scegliere tra `provideConfig` e `ConfigModule.withConfig`: 50 — Ricette: configurazione, DI, interceptor, eventi;
- creare una feature library custom lazy-loaded (root module + `featureModules`), cioè un nuovo NgModule da agganciare a `SpartacusFeaturesModule`: 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni;
- cosa si rompe negli upgrade (incluse le modernizzazioni come questo schematic) e checklist upgrade-safe: 54 — API pubblica vs interna, upgrade, anti-pattern, checklist.

9. Trappole, casi limite, comportamenti controintuitivi

1. **L'ordine dei provider non è l'ordine testuale degli** `imports`**.** I `providers` dei `ModuleWithProviders` sono emessi solo dopo tutti gli import fratelli, anche quelli dichiarati dopo. In `AppModule` i provider di `StoreModule.forRoot({})` finiscono **dopo** quelli dell'intero sotto-albero di `SpartacusModule` — angular:packages/core/src/di/provider_collection.ts:L316-L321, projects/storefrontapp/src/app/app.module.ts:L16-L22.
2. `forRoot()` **senza** `providers` **= import del tipo.** `BaseCoreModule.forRoot()` restituisce solo `{ngModule}` e si comporta come l'import diretto; `ProcessModule.forRoot()` con `providers: []` attiva invece il ramo "rinviato", con zero provider — angular:packages/core/src/di/provider_collection.ts:L360-L362, core-libs/core/src/base-core.module.ts:L48-L52, core-libs/core/src/process/process.module.ts:L14-L18.
3. **Tipo +** `forRoot()` **dello stesso modulo.** `OutletModule` e `OutletModule.forRoot()` in `BaseStorefrontModule` non duplicano il modulo ma aggiungono i provider del MWP (dedup solo sul tipo). Due chiamate allo stesso `forRoot()` registrerebbero due volte i provider, ad esempio due `APP_INITIALIZER` multi — core-libs/storefront/base-storefront.module.ts:L31-L31, core-libs/storefront/base-storefront.module.ts:L43-L43, angular:packages/core/src/di/provider_collection.ts:L276-L276, angular:packages/core/src/di/provider_collection.ts:L360-L362.
4. **Nessuna guardia contro il doppio** `forRoot` **nei moduli Spartacus.** Una ricerca testuale nei `*.module.ts` di `core-libs` non trova `@SkipSelf()`/`skipSelf: true`, a differenza di `RouterModule` (`FOR_ROOT_CALLED_TWICE`) — angular:packages/router/src/router_module.ts:L222-L233. Importare `BaseStorefrontModule` due volte, ad esempio in un modulo lazy, non incontra alcuna guardia esplicita contro il doppio `forRoot` (DA VERIFICARE a runtime l'assenza di altri errori: build non eseguibile).
5. `BrowserModule` **nell'**`AppModule` **della demo.** `bootstrapApplication` aggiunge già `BROWSER_MODULE_PROVIDERS`, ma il controllo `BROWSER_MODULE_ALREADY_LOADED` usa `skipSelf: true` e le due copie stanno nello stesso environment injector: nessun errore. Effetti: i provider non multi di `BrowserModule` sono registrati due volte (vince la seconda copia, salvo override successivi come `ErrorHandler` → `CxErrorHandler` di `ErrorHandlingModule.forRoot()`, core-libs/core/src/error-handling/error-handling.module.ts:L25-L28); i multi provider `EVENT_MANAGER_PLUGINS` (`DomEventsPlugin`, `KeyEventsPlugin`, angular:packages/platform-browser/src/browser.ts:L262-L267) compaiono invece due volte nell'array multi: `R3Injector` fa `multiRecord.multi!.push(provider)` per ogni copia (angular:packages/core/src/di/r3_injector.ts:L497-L521). Poiché le due copie sono lo stesso oggetto provider e il record è indicizzato per oggetto provider, entrambe le voci risolvono allo stesso record: la stessa istanza di ciascun plugin appare due volte nell'array (angular:packages/core/src/di/r3_injector.ts:L508-L521, angular:packages/core/src/di/injector_compatibility.ts:L327-L330). Inoltre si aggiungono i `TESTABILITY_PROVIDERS`, che `bootstrapApplication` di per sé non include. Lo schematic invece rimuove `BrowserModule` dalle app migrate: demo e app migrate divergono — projects/storefrontapp/src/app/app.module.ts:L17-L17, angular:packages/platform-browser/src/browser.ts:L171-L171, angular:packages/platform-browser/src/browser.ts:L288-L288, angular:packages/platform-browser/src/browser.ts:L294-L297, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-app-module.ts:L136-L156.
6. **Zoneless di default.** In Angular 21 `bootstrapApplication` è zoneless (`NoopNgZone`); demo e schematic aggiungono `provideZoneChangeDetection({eventCoalescing: true})`, che vince perché arriva dopo nella lista. Toglierlo cambia il modello di change detection dell'intera app — angular:packages/core/src/application/create_application.ts:L66-L71, projects/storefrontapp/src/app/app.config.ts:L30-L30, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/create-app-config.ts:L46-L46. Vedi 04.
7. `importProvidersFrom` **ha due limiti.** Non accetta componenti standalone (errore in dev) e il risultato non può stare in `@Component.providers` (`PROVIDER_IN_WRONG_CONTEXT`) — angular:packages/core/src/di/

provider_collection.ts:L163-L174, angular:packages/core/src/render3/errors_di.ts:L50-L55.

8. **L'appiattimento avviene all'import di** `app.config.ts`**.** `importProvidersFrom(AppModule)` è valutato con il modulo ES, non dentro `bootstrapApplication`: errori di grafo (cicli, provider invalidi in dev) emergono già all'import del file, fuori dal `.catch` di `main.ts` — derivato da projects/storefrontapp/src/app/app.config.ts:L26-L35 e angular:packages/core/src/di/provider_collection.ts:L144-L149. DA VERIFICARE a runtime se, in un bundle ESM, l'eccezione arrivi davvero prima della chiamata a `bootstrapApplication` (dipende dall'ordine di valutazione dei moduli prodotto dal builder).
9. **Ogni NgModule è istanziato eager.** L'`ENVIRONMENT_INITIALIZER` sintetico fa `inject(defType)`: i costruttori dei moduli (es. `ConfigModule` che inietta `ConfigurationService`) girano all'avvio, prima degli `APP_INITIALIZER` — angular:packages/core/src/di/provider_collection.ts:L339-L343, angular:packages/core/src/platform/bootstrap.ts:L89-L93, angular:packages/core/src/platform/bootstrap.ts:L140-L141, core-libs/core/src/config/config.module.ts:L17-L19.
10. **Un injector in più per il root component.** Per `AppComponent` viene creato un environment injector figlio `Standalone[AppComponent]` se il grafo dei suoi `imports` produce provider; poiché `walkProviderTree` emette 3 provider sintetici per ogni NgModule e `StorefrontComponent` importa `RouterModule`, la lista non dovrebbe essere vuota. DA VERIFICARE a runtime (build non eseguibile: `node_modules` assente) che `AppComponent.ɵcmp.dependencies` includa `RouterModule` tramite `StorefrontComponent` e che l'injector venga creato — angular:packages/core/src/render3/standalone_service.ts:L31-L43, angular:packages/core/src/di/provider_collection.ts:L285-L291, core-libs/storefront/layout/main/storefront.component.ts:L54-L56.
11. **Lo schematic riconosce solo** `platformBrowser()` **letterale.** Varianti (`platformBrowserDynamic()`, variabile intermedia) non vengono convertite e, poiché la regola non lancia, non compare neanche il messaggio di fallback — core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-main-ts.ts:L102-L112.
12. `app.config.ts` **preesistente.** `createAppConfig`/`createAppConfigServer` saltano con warning se il file esiste: un `app.config.ts` già presente non riceve `importProvidersFrom(AppModule)` — core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/create-app-config.ts:L20-L25, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/ssr/create-app-config-server.ts:L16-L22.
13. **Migrazione parziale "riuscita".** Il wrapper di fallback inghiotte le eccezioni e restituisce il tree: la migrazione può terminare con file in stato intermedio (es. `main.ts` aggiornato ma `AppModule` no) — core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L28-L43.
14. **Server senza** `context`**.** `bootstrapApplication` lato server senza terzo argomento fallisce con `PLATFORM_NOT_FOUND`: per questo `main.server.ts` esporta una funzione — angular:packages/core/src/application/create_application.ts:L47-L54, projects/storefrontapp/src/main.server.ts:L23-L24.
15. `mergeApplicationConfig` **concatena.** I provider server seguono quelli browser e quindi li sovrascrivono a parità di token; inoltre `importProvidersFrom` è chiamato due volte separatamente (`AppServerModule`, `TestConfigServerModule.forRoot()`), ciascuna con il proprio `dedup` — angular:packages/core/src/application/application_config.ts:L31-L38, angular:packages/core/src/di/provider_collection.ts:L156-L156, projects/storefrontapp/src/app/app.config.server.ts:L21-L26.
16. `AppRoutingModule` **prima di** `SpartacusModule`**.** Il commento di `BaseCoreModule` richiede `SiteContextModule.forRoot()` "after RouterModule.forRoot": nella demo vale perché `AppRoutingModule` precede `SpartacusModule`; invertire l'ordine farebbe vincere il `DefaultUrlSerializer` di `ROUTER_PROVIDERS` — core-libs/core/src/base-core.module.ts:L38-L38, projects/storefrontapp/src/app/app.module.ts:L19-L21, angular:packages/router/src/router_module.ts:L69-L71.
17. `CmsLcpModule.forRoot()` **senza tipo di ritorno.** Funziona strutturalmente, ma è incoerente con gli altri `forRoot()` (e non beneficia del controllo di tipo `ModuleWithProviders`) — core-libs/storefront/cms-

structure/cms-lcp-context/cms-lcp.module.ts:L13-L18.

10. Come fare debug

- **Vedere la lista appiattita.** `importProvidersFrom(AppModule)` restituisce un oggetto con `ɵproviders` (angular:packages/core/src/di/provider_collection.ts:L144-L149): in sviluppo un `console.log(importProvidersFrom(AppModule))` temporaneo in `app.config.ts` mostra l'array nell'ordine esatto prodotto da `walkProviderTree`. Cercare nell'array tutte le voci con `provide: UrlSerializer` permette di verificare chi vince (l'ultima, §3.17). `ɵproviders` è interno: usarlo solo per diagnosi.
- **Breakpoint mirati.** In `walkProviderTree` (angular:packages/core/src/di/provider_collection.ts:L292-L362) un breakpoint condizionale su `defType.name === '...'` mostra quando un modulo viene visitato e se è `isDuplicate`. In `processInjectorTypesWithProviders` (angular:packages/core/src/di/provider_collection.ts:L195-L209) si vedono i provider "rinviati" dei `forRoot()`.
- **Riconoscere gli injector.** L'environment injector dell'app ha debug name `'Environment Injector'` (angular:packages/core/src/application/create_application.ts:L75-L75); quello del root component standalone `Standalone[AppComponent]` (angular:packages/core/src/render3/standalone_service.ts:L38-L40); gli injector creati da `NgModuleRef` hanno come nome il nome del modulo (angular:packages/core/src/render3/ng_module_ref.ts:L67-L95). Utile per T10.
- **Accedere all'injector dell'app.** `bootstrapApplication` restituisce `Promise<ApplicationRef>`; `appRef.injector` è l'`EnvironmentInjector` dell'app (angular:packages/core/src/application/application_ref.ts:L359-L367). Un `.then((appRef) => ...)` temporaneo in `main.ts` (projects/storefrontapp/src/main.ts:L17-L17) consente `appRef.injector.get(X)` per controllare quale implementazione ha vinto.
- **Stabilità.** Se l'SSR non termina o `whenStable()` non risolve, sospettare task ricorrenti avviati all'avvio, come avverte il JSDoc (angular:packages/core/src/application/application_ref.ts:L172-L179); `appRef.isStable` deriva dai pending task (angular:packages/core/src/application/application_ref.ts:L328-L357). Approfondimento in 04 e 07.
- **Hook dopo il bootstrap.** Un provider `APP_BOOTSTRAP_LISTENER` multi in `appConfig` viene chiamato per ogni componente bootstrappato (angular:packages/core/src/application/application_ref.ts:L775-L795): utile per misurare quando il root è montato.
- **Errori di bootstrap e loro significato.**

ERRORE (RUNTIMEERRORCODE)	CAUSA	RIFERIMENTO
`TYPE_IS_NOT_STANDALONE`	root component non standalone passato a `bootstrapApplication`	angular:packages/core/src/render3/errors.ts:L25-L37
`IMPORT_PROVIDERS_FROM_STANDALONE`	componente standalone passato a `importProvidersFrom`	angular:packages/core/src/di/provider_collection.ts:L163-L174
`PROVIDER_IN_WRONG_CONTEXT`	`importProvidersFrom`/`EnvironmentProviders` in `@Component.providers`	angular:packages/core/src/render3/errors_di.ts:L50-L61
ciclo di import (dev)	`cyclicDependencyErrorWithDetails` con il percorso	angular:packages/core/src/di/provider_collection.ts:L268-L273
`PLATFORM_NOT_FOUND`	`bootstrapApplication` lato server senza `BootstrapContext`	angular:packages/core/src/application/create_application.ts:L47-L54
`BROWSER_MODULE_ALREADY_LOADED`	`BrowserModule` in un injector figlio (es. modulo lazy)	angular:packages/platform-browser/src/browser.ts:L292-L307

ERRORE (RUNTIMEERRORCODE)	CAUSA	RIFERIMENTO
FOR_ROOT_CALLED_TWICE	RouterModule.forRoot fuori dal root injector	angular:packages/router/src/router_module.ts:L222-L233
BOOTSTRAP_COMPONENTS_NOT_FOUND	modulo bootstrappato senza bootstrap né ngDoBootstrap	angular:packages/core/src/platform/bootstrap.ts:L196-L215
ASYNC_INITIALIZERS_STILL_RUNNING	appRef.bootstrap chiamato prima della fine degli initializer	angular:packages/core/src/application/application_ref.ts:L508-L521
PROVIDED_BOTH_ZONE_AND_ZONELESS (warning)	forniti sia zone sia zoneless	angular:packages/core/src/platform/bootstrap.ts:L95-L105

- **Schematic.** Se una regola fallisce compare il messaggio con il link alla guida e il tree resta invariato per quel passo (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L28-L43); dopo la migrazione controllare a mano main.ts (T11) e app.config.ts (T12). Approfondimento in 43.

---

11. Domande di autoverifica

1. **In** AppModule StoreModule.forRoot({}) **è dichiarato prima di** SpartacusModule**. In che posizione finiscono i suoi provider rispetto a quelli di Spartacus?** Dopo: i providers dei ModuleWithProviders importati sono emessi solo al termine della visita di tutti gli import di AppModule, quindi dopo l'intero sotto-albero SpartacusModule — angular:packages/core/src/di/provider_collection.ts:L316-L321, projects/storefrontapp/src/app/app.module.ts:L16-L22.
2. **Perché** BaseCoreModule.forRoot() **e l'import diretto di** BaseCoreModule **sono equivalenti?** Perché forRoot() restituisce {ngModule: BaseCoreModule} senza providers, e walkProviderTree restituisce true (provider da emettere) solo se providers !== undefined — core-libs/core/src/base-core.module.ts:L47-L53, angular:packages/core/src/di/provider_collection.ts:L360-L362.
3. **Quale** UrlSerializer **vince nell'app demo e perché?** SiteContextUrlSerializer: RouterModule.forRoot (via AppRoutingModule) registra DefaultUrlSerializer prima, SiteContextModule.forRoot() (via BaseCoreModule) lo ridefinisce dopo, e per i provider non multi vince l'ultimo — angular:packages/router/src/router_module.ts:L69-L71, core-libs/core/src/site-context/providers/site-context-params-providers.ts:L16-L17, core-libs/core/src/base-core.module.ts:L38-L38; regola in 01.
4. **Un'app Angular 21 bootstrappata con** bootstrapApplication **senza ulteriori provider usa zone.js?** No: il primo provider dell'environment injector è provideZonelessChangeDetectionInternal() (NoopNgZone); per usare zone serve provideZoneChangeDetection(...), come fa app.config.ts — angular:packages/core/src/application/create_application.ts:L66-L71, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L388-L394, projects/storefrontapp/src/app/app.config.ts:L30-L30.
5. **Perché** main.server.ts **non può semplicemente chiamare** bootstrapApplication(AppComponent, config)**?** Lato server internalCreateApplication richiede un platformRef e altrimenti lancia PLATFORM_NOT_FOUND; il BootstrapContext lo fornisce, quindi main.server.ts esporta una funzione che lo riceve — angular:packages/core/src/application/create_application.ts:L47-L54, projects/storefrontapp/src/main.server.ts:L23-L26.
6. **Quando viene eseguito il costruttore di** ConfigModule**, e cosa ne garantisce l'esecuzione?** All'avvio, dentro ngZone.run, durante resolveInjectorInitializers(), prima degli APP_INITIALIZER: walkProviderTree registra per ogni NgModule un ENVIRONMENT_INITIALIZER che fa inject(defType) —

angular:packages/core/src/di/provider_collection.ts:L339-L343, angular:packages/core/src/platform/bootstrap.ts:L84-L93, core-libs/core/src/config/config.module.ts:L15-L19.

7. **Perché** `StorefrontComponentModule` **mette** `StorefrontComponent` **negli** `imports` **e non nelle** `declarations`**?** Perché `StorefrontComponent` è standalone (default da v19) e un componente standalone non si dichiara: si importa. L'NgModule resta come shim che lo ri-esporta — core-libs/storefront/layout/main/storefront-component.module.ts:L19-L34, angular:packages/compiler-cli/src/ngtsc/core/src/compiler.ts:L477-L482.
8. **Importare** `RouterModule` **senza** `forRoot` **(come fa** `BaseStorefrontModule`**) porta provider del router nell'injector?** No: `RouterModule` ha come `imports`/`exports` solo direttive standalone, che il compilatore non inserisce negli `imports` di `ɵinj`; i provider arrivano solo da `RouterModule.forRoot` — angular:packages/router/src/router_module.ts:L104-L108, angular:packages/compiler-cli/src/ngtsc/annotations/ng_module/src/handler.ts:L798-L848.
9. **Lo schematic** `modernize-app-to-standalone-bootstrap-application` **viene eseguito da** `ng update`**?** No: è registrato in `collection.json`, non in `migrations.json`, quindi va lanciato esplicitamente — core-libs/schematics/src/collection.json:L49-L52.
10. **Perché l'**`AppModule` **della demo può importare** `BrowserModule` **nonostante il bootstrap standalone, senza** `BROWSER_MODULE_ALREADY_LOADED`**?** Il controllo cerca il marker con `skipSelf: true`, cioè solo negli injector antenati; qui le due copie dei provider stanno nello stesso environment injector, quindi l'errore non scatta (ma i provider sono duplicati: per i non multi vince l'ultima copia, i multi `EVENT_MANAGER_PLUGINS` compaiono due volte nell'array, cfr. §9 T5) — angular:packages/platform-browser/src/browser.ts:L292-L307, angular:packages/platform-browser/src/browser.ts:L171-L171, projects/storefrontapp/src/app/app.module.ts:L17-L17.

TAVOLA 02.1 **4.1 Sequenza di bootstrapApplication nell'app demo (browser)**

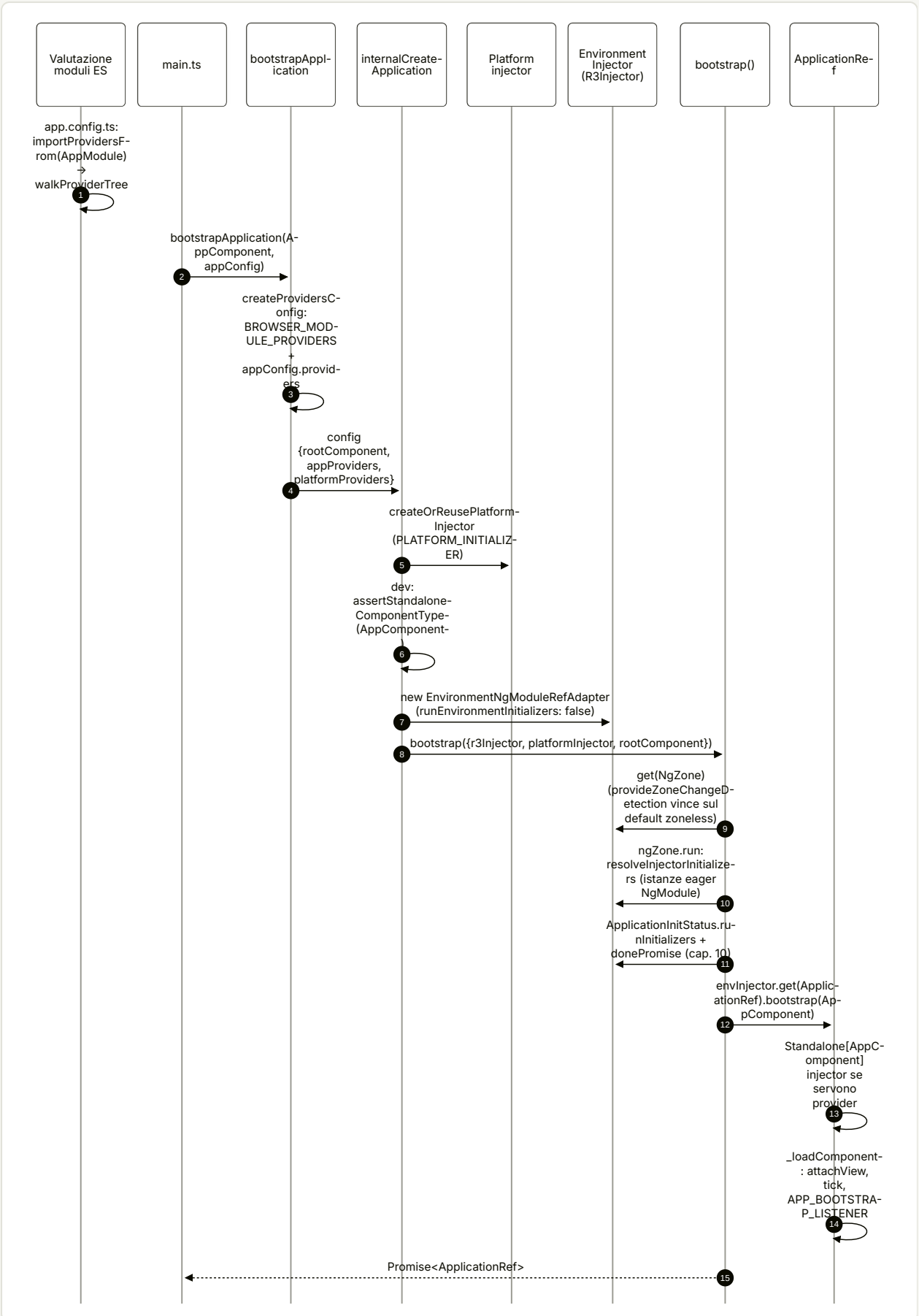


TAVOLA 02.2 **4.2 walkProviderTree per un singolo contenitore**

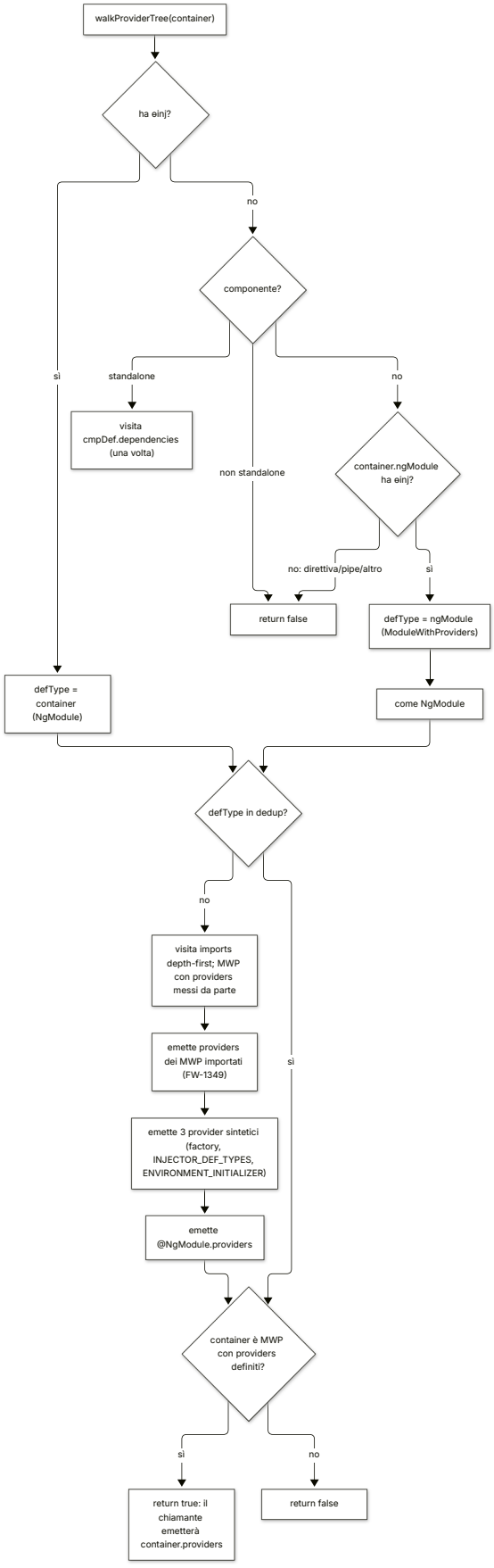


TAVOLA 02.3 **4.3 I due percorsi di bootstrap convergono**

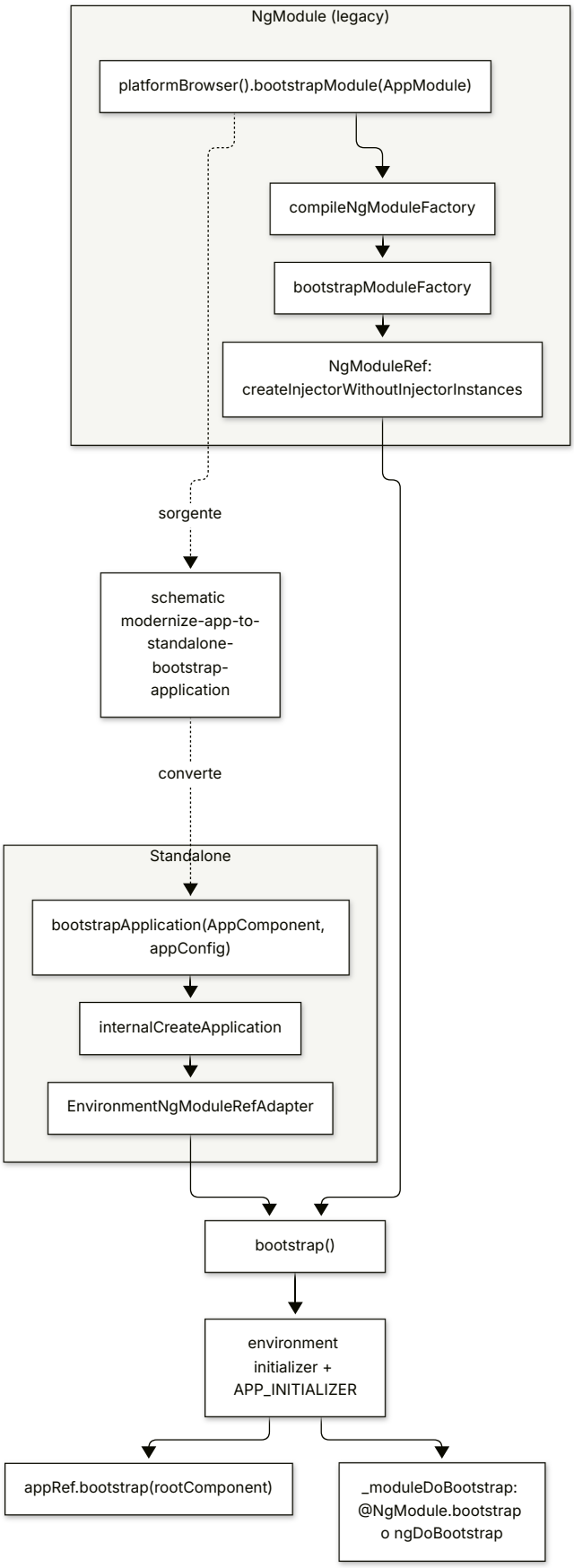
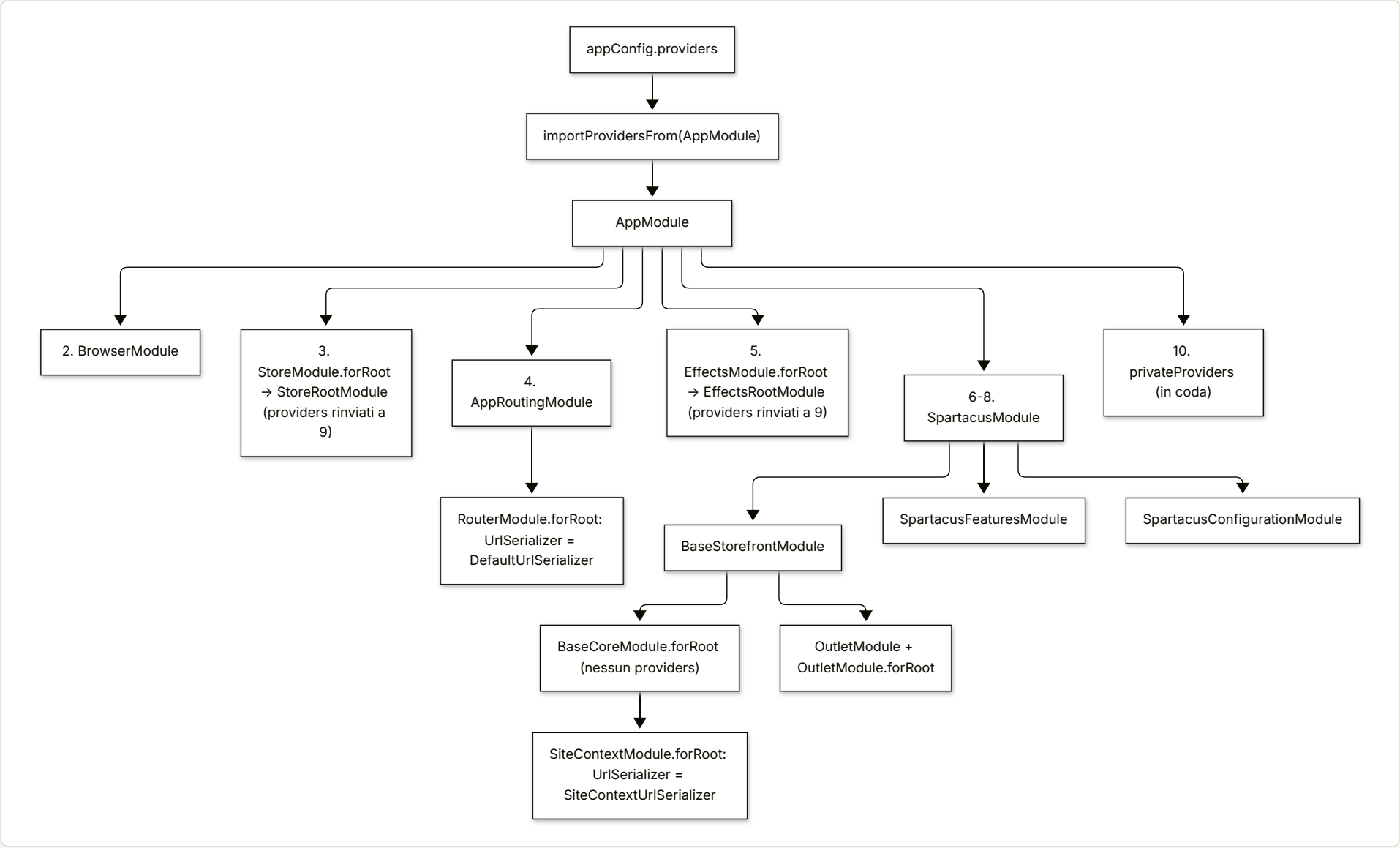


TAVOLA 02.4 **4.4 Albero dei moduli dell'app demo e ordine di emissione**



03

Lazy loading: router e moduli caricati a runtime

Code splitting e nuovo injector a runtime. `loadChildren`/`loadComponent` di Angular, e perché Spartacus non usa il lazy loading del router.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Angular @ `v21.2.23` (prefisso `angular:`).

Prerequisiti: 01 — Dependency Injection (gerarchia degli injector, `R3Injector`, `ENVIRONMENT_INITIALIZER`) e 05 — Il Router di Angular (recognize, guard, attivazione). Il lato Spartacus (configurazione `featureModules`, proxy facade, `MODULE_INITIALIZER` nel dettaglio) è trattato in 12 — Feature module e lazy loading in Spartacus: qui lo si richiama solo per confrontarlo con il meccanismo di Angular.

---

1. A cosa serve e perché esiste

"Lazy loading" in Angular significa due cose distinte che spesso vengono confuse:

1. **Code splitting**: un `import()` dinamico fa sì che il bundler produca un chunk separato, scaricato solo quando la funzione viene chiamata. È una funzionalità del linguaggio e del bundler, non di Angular.
2. **Un nuovo injector a runtime**: quando il chunk contiene un `NgModule`, qualcuno deve istanziarlo, cioè creare un nuovo `EnvironmentInjector` figlio di un injector esistente, eseguire gli initializer del modulo e far sì che i componenti creati "dentro" quella feature vedano i suoi provider.

Angular offre un solo meccanismo *integrato* che fa entrambe le cose: le proprietà `loadChildren` e `loadComponent` di una `Route`, gestite dal `RouterConfigLoader` (`angular:packages/router/src/router_config_loader.ts:L41-L110`). Il router decide *quando* caricare (durante il recognize di una navigazione, o in background con il preloading), *quale injector usare come parent* e *dove memorizzare il risultato* (campi interni della `Route`, `angular:packages/router/src/models.ts:L771-L792`).

Spartacus, invece, **non usa il lazy loading del router**. Nel codice (non di test) delle librerie e dell'app demo non c'è alcun `loadChildren`, `loadComponent:`, `createNgModule`, `createEnvironmentInjector`, `provideRouter` o `withPreloading` (`createEnvironmentInjector` e `provideRouter` compaiono solo in file `*.spec.ts`/`*_spec.ts`); l'unico `compileModuleAsync` è in `LazyModulesService` (`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L230-L230`). L'unico `RouterModule.forRoot` ha un array di route vuoto (`core-libs/storefront/router/app-routing.module.ts:L13-L19`). Il motivo è architetturale: in uno storefront Spartacus una pagina è composta dal CMS, e lo stesso componente (es. il mini-cart) può comparire in qualunque route. Il codice di una feature va quindi caricato **quando il CMS chiede un suo componente**, non quando il router entra in una certa route. Spartacus ha perciò un proprio caricatore, fuori dal router, che però riproduce — quasi riga per riga — gli stessi passi del router: `import()` → `compileModuleAsync` → `factory.create(parentInjector)` → componenti creati con l'injector del modulo come environment injector (`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L107`, `core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L56-L68`).

Questo capitolo serve a:

- capire il meccanismo di Angular (router e API core `createNgModule` / `createEnvironmentInjector`) sui sorgenti v21.2;
- mostrare che `Compiler.compileModuleAsync` e `NgModuleFactory` — deprecati — in Ivy sono solo un involucro di `new NgModuleRef(...)`, quindi l'approccio di Spartacus è equivalente a `createNgModule` (`angular:packages/core/src/linker/compiler.ts:L57-L66`, `angular:packages/core/src/render3/ng_module_ref.ts:L127-L135`);
- mettere in fila le differenze che contano davvero (contesto di iniezione, export `default`, scelta del parent, initializer asincroni, distruzione, route e guard che vivono nel root) e le trappole che ne derivano.

2. Mappa dei file

Angular — router (`angular:`)

PATH	RUOLO
angular:packages/router/src/models.ts	Tipi `LoadChildrenCallback`, `LoadChildren`, campi `Route.loadComponent`, `loadChildren`, `providers`, `canLoad` e cache interne `_injector`, `_loadedRoutes`, `_loadedInjector`, `_loadedNgModuleFactory`, `_loadedComponent`; `LoadedRouterConfig`
angular:packages/router/src/router_config_loader.ts	Token `ROUTES`, classe `RouterConfigLoader` (dedup con `WeakMap`), funzione `loadChildren` (unwrap `default`, `compileModuleAsync`, `factory.create`, lettura `ROUTES` con `self`)
angular:packages/router/src/recognize.ts	`processSegment` / `processSegmentAgainstRoute` / `getChildConfig`: scelta dell'injector e trigger di `loadChildren` + `canLoad`
angular:packages/router/src/navigation_transition.ts	Eventi `RouteConfigLoadStart`/`RouteConfigLoadEnd`; fase "LOAD COMPONENTS" per `loadComponent`
angular:packages/router/src/utils/config.ts	`getOrCreateRouteInjectorIfNeeded` (`createEnvironmentInjector` per i `providers` di route), `assertStandalone`
angular:packages/router/src/utils/config_matching.ts	Creazione dell'injector di route al match
angular:packages/router/src/utils/collection.ts	`wrapIntoPromise`
angular:packages/router/src/router_preloader.ts	`PreloadingStrategy`, `PreloadAllModules`, `NoPreloading`, `RouterPreloader`
angular:packages/router/src/route_injector_cleanup.ts	`destroyUnusedInjectors` (distruzione sperimentale degli injector di route)
angular:packages/router/src/provide_router.ts	`withPreloading`, `withExperimentalAutoCleanupInjectors`
angular:packages/router/src/route_reuse_strategy.ts	`ExperimentalRouteReuseStrategy`, `BaseRouteReuseStrategy.shouldDestroyInjector`
angular:packages/router/src/router.ts	`Router.config` iniziale letto da `ROUTES`
angular:packages/router/src/router_module.ts	`RouterModule.forChild` → provider multi di `ROUTES`
angular:packages/router/src/router_outlet_context.ts, angular:packages/router/src/directives/router_outlet.ts, angular:packages/router/src/operators/activate_routes.ts	Passaggio dell'environment injector della route al componente attivato
angular:packages/router/src/index.ts	Barrel pubblico di `@angular/router`

Angular — core (`angular:`)

PATH	RUOLO
angular:packages/core/src/linker/compiler.ts	`Compiler` (deprecato; in Ivy `compileModuleSync = new NgModuleFactory`)
angular:packages/core/src/linker/ng_module_factory.ts	`NgModuleRef` e `NgModuleFactory` astratti (factory deprecata)

PATH	RUOLO
angular:packages/core/src/render3/ng_module_ref.ts	createNgModule, NgModuleRef R3, NgModuleFactory R3, EnvironmentNgModuleRefAdapter, createEnvironmentInjector
angular:packages/core/src/di/create_injector.ts	createInjectorWithoutInjectorInstances
angular:packages/core/src/di/r3_injector.ts	EnvironmentInjector, R3Injector (scope, get, destroy, initializer)
angular:packages/core/src/di/provider_collection.ts	importProvidersFrom/walkProviderTree, provideEnvironmentInitializer
angular:packages/core/src/di/initializer_token.ts	ENVIRONMENT_INITIALIZER
angular:packages/core/src/di/contextual.ts	runInInjectionContext
angular:packages/core/src/render3/standalone_service.ts, angular:packages/core/src/render3/definition.ts	Injector dei componenti standalone
angular:packages/core/src/linker/view_container_ref.ts, angular:packages/core/src/render3/component_ref.ts, angular:packages/core/src/render3/chained_injector.ts	createComponent con environmentInjector / NgModuleRef

Spartacus

PATH	RUOLO
core-libs/core/src/lazy-loading/lazy-modules.service.ts	LazyModulesService: import() → compileModuleAsync → factory.create(parent) → MODULE_INITIALIZER → ModuleInitializedEvent
core-libs/core/src/lazy-loading/feature-modules.service.ts	FeatureModulesService: cache per nome, alias, dipendenze
core-libs/core/src/lazy-loading/unified-injector.ts	UnifiedInjector: root + injector dei moduli lazy
core-libs/core/src/lazy-loading/tokens.ts	MODULE_INITIALIZER
core-libs/core/src/lazy-loading/lazy-loading.module.ts	LazyLoadingModule.forRoot(): fallback eager via APP_INITIALIZER
core-libs/core/src/lazy-loading/events/module-initialized-event.ts	ModuleInitializedEvent
core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts	Proxy facade risolte sull'injector lazy
core-libs/core/src/util/combined-injector.ts	CombinedInjector (interno, non esportato)
core-libs/core/src/cms/config/cms-config.ts	FeatureModuleConfig, CmsConfig.featureModules
core-libs/storefront/cms-structure/services/cms-features.service.ts	CmsFeaturesService: componentType → feature → NgModuleRef
core-libs/storefront/cms-structure/services/cms-components.service.ts	determineMappings: attende le feature prima del rendering
core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts	Route CMS aggiunte a runtime con resetConfig; guard risolti con UnifiedInjector
core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts	createComponent(..., module): modulo lazy come environment injector
core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts	import() di singoli componenti + PendingTasks

PATH	RUOLO
core-libs/storefront/router/app-routing.module.ts	Unico `RouterModule.forRoot([])`

3. Come funziona passo per passo

3.1 Il contratto di `loadChildren` e `loadComponent`

- `LoadChildrenCallback` è una funzione senza argomenti che può restituire, in modo sincrono, come `Observable` o come `Promise`: una classe (`Type`), un `NgModuleFactory`, un array `Routes`, oppure un oggetto con export `default` di classe o `Routes`. L'`Observable` non ammette `NgModuleFactory` — angular:packages/router/src/models.ts:L267-L274. `LoadChildren` è solo un alias: la vecchia sintassi a stringa (`'./x#XModule'`) non esiste più — angular:packages/router/src/models.ts:L283-L283.
- Con un export `default` si può omettere `.then`: `loadChildren: () => import('./lazy-route/lazy.routes')` — angular:packages/router/src/models.ts:L256-L261.
- `loadComponent` restituisce un `Type` (o Observable/Promise di `Type` o di `DefaultExport<Type>`); il risultato viene memorizzato in `_loadedComponent` — angular:packages/router/src/models.ts:L626-L634.
- `providers` su una route: il router crea un **nuovo** `EnvironmentInjector` per la route e i suoi figli; se la route ha anche `loadChildren` che restituisce un modulo, quell'injector fa da parent del modulo lazy — angular:packages/router/src/models.ts:L760-L769.
- La `Route` contiene campi `@internal` di cache: `_injector`, `_loadedRoutes`, `_loadedInjector`, `_loadedNgModuleFactory` — angular:packages/router/src/models.ts:L771-L792. Il risultato di un caricamento è un `LoadedRouterConfig = {routes, injector, factory?}`, con `injector` `undefined` quando sono state caricate solo `Routes` — angular:packages/router/src/models.ts:L795-L799.
- `canLoad` è deprecato a favore di `canMatch` — angular:packages/router/src/models.ts:L706-L715.

In sintesi, la `Route` stessa è la cache: dopo il primo caricamento il router non richiama più la callback.

3.2 Injector per route: `providers` e `getOrCreateRouteInjectorIfNeeded`

Durante il matching, solo se `match()` ha avuto successo e prima dei `canMatch`, il router chiama `getOrCreateRouteInjectorIfNeeded(route, injector)` — angular:packages/router/src/utils/config_matching.ts:L71-L71. La funzione crea **una sola volta** un injector con `createEnvironmentInjector(route.providers, currentInjector, 'Route: <path>')` e lo salva in `route._injector` — angular:packages/router/src/utils/config.ts:L31-L43.

Da quel momento:

- `processSegment` processa ogni route con `r._injector ?? injector` — angular:packages/router/src/recognize.ts:L244-L247;
- dopo il match, `processSegmentAgainstRoute` passa a `getChildConfig` l'injector dei `providers` e usa `route._loadedInjector ?? injector` per i figli — angular:packages/router/src/recognize.ts:L438-L441.

Questo è il punto in cui `createEnvironmentInjector` entra nel router: **i `providers` di route non richiedono alcun NgModule** (vedi §3.6).

3.3 `getChildConfig`: quando parte `loadChildren`

`getChildConfig` (angular:packages/router/src/recognize.ts:L494-L530) distingue tre casi:

1. `route.children` statici: le route figlie "appartengono allo stesso modulo", l'injector non cambia — angular:packages/router/src/recognize.ts:L499-L502.
2. `route.loadChildren` già caricato (`_loadedRoutes !== undefined`): restituisce la cache; se esiste la factory ma non l'injector (perché distrutto, §3.9), lo **ricrea** con `_loadedNgModuleFactory.create(injector).injector` — angular:packages/router/src/recognize.ts:L504-L512.
3. `route.loadChildren` non ancora caricato: controlla l'abort della navigazione, esegue i guard `canLoad` (`runCanLoadGuards` restituisce subito `of(true)` se non ce ne sono — angular:packages/router/src/operators/check_guards.ts:L230-L240), poi chiama `configLoader.loadChildren(injector, route)` e salva il risultato sulla route; se `canLoad` fallisce lancia `canLoadFails(route)` — angular:packages/router/src/recognize.ts:L514-L527.

Il caricamento di `loadChildren` avviene quindi **durante il recognize**, perché il router deve conoscere le route figlie per continuare il matching.

3.4 `RouterConfigLoader.loadChildren` e la funzione `loadChildren`

`RouterConfigLoader` è `providedIn: 'root'`, deduplica i caricamenti in corso con due `WeakMap<Route, Promise>` e inietta `Compiler` — angular:packages/router/src/router_config_loader.ts:L41-L47.

1. Se c'è già una promise in volo per la route la restituisce; se `_loadedRoutes` è valorizzato restituisce `{routes, injector}` dalla cache — angular:packages/router/src/router_config_loader.ts:L81-L86.
2. Notifica `onLoadStartListener`, che `NavigationTransitions` collega all'evento `RouteConfigLoadStart` (e `onLoadEndListener` a `RouteConfigLoadEnd`) — angular:packages/router/src/navigation_transition.ts:L389-L393.
3. Chiama la funzione esportata `loadChildren(route, compiler, parentInjector, onLoadEndListener)` e salva su `route` i campi `_loadedRoutes`, `_loadedInjector`, `_loadedNgModuleFactory` — angular:packages/router/src/router_config_loader.ts:L91-L102.

La funzione `loadChildren` (angular:packages/router/src/router_config_loader.ts:L120-L163) — usata anche per la route discovery del prerendering in `@angular-devkit/build-angular` (angular:packages/router/src/router_config_loader.ts:L112-L119) e riesportata come API privata (angular:packages/router/src/private_export.ts:L11-L11):

1. esegue `route.loadChildren` dentro `runInInjectionContext(parentInjector, ...)`, quindi la callback può usare `inject()` — angular:packages/router/src/router_config_loader.ts:L126-L129;
2. normalizza il risultato con `wrapIntoPromise` (`firstValueFrom` per gli Observable, `Promise.resolve` per il resto — angular:packages/router/src/utils/collection.ts:L88-L93), spacchetta il `default` (angular:packages/router/src/router_config_loader.ts:L165-L176) e, in JIT, risolve le risorse dei componenti (angular:packages/router/src/router_config_loader.ts:L178-L191);
3. se il valore non è già un `NgModuleFactory` né un array, lo passa a `compiler.compileModuleAsync(t)` — angular:packages/router/src/router_config_loader.ts:L131-L136;
4. con un **array di route** non c'è alcun injector e le route figlie devono usare componenti standalone (`requireStandaloneComponents = true`) — angular:packages/router/src/router_config_loader.ts:L141-L149;
5. con un **modulo** crea `factory.create(parentInjector).injector` e legge `ROUTES` con `{optional: true, self: true}`: senza `self` un modulo privo di `RouterModule.forChild()` erediterebbe le route del parent e il

preloader entrerebbe in loop infinito — angular:packages/router/src/router_config_loader.ts:L150-L158;

6. normalizza con `standardizeConfig` e, in dev, valida — angular:packages/router/src/router_config_loader.ts:L159-L162.

`RouterModule.forChild(routes)` non fa altro che fornire `ROUTES` multi con `useValue: routes` — angular:packages/router/src/router_module.ts:L184-L189 — ed è per questo che il router trova le route del modulo lazy leggendo `ROUTES` dal suo injector.

3.5 `loadComponent` e i componenti standalone

`loadComponent` non viene eseguito nel recognize ma nella fase **LOAD COMPONENTS** della navigazione, che segue recognize, guard (`checkGuards`) e resolver (`resolveData`): una navigazione bloccata da un guard o da un resolver non scarica il chunk del componente — angular:packages/router/src/navigation_transition.ts:L649-L649, angular:packages/router/src/navigation_transition.ts:L690-L690. Usa l'injector `route._environmentInjector` dello snapshot; i caricamenti dell'albero sono attesi con `Promise.all` — angular:packages/router/src/navigation_transition.ts:L717-L739.

`RouterConfigLoader.loadComponent`:

- riusa la promise in volo o `_loadedComponent` — angular:packages/router/src/router_config_loader.ts:L49-L54;
- esegue la callback in `runInInjectionContext(injector, ...)`, poi `wrapIntoPromise`, unwrap del `default` e risorse JIT — angular:packages/router/src/router_config_loader.ts:L59-L64;
- in dev verifica che sia standalone, salva `_loadedComponent` e pulisce la mappa nel `finally` — angular:packages/router/src/router_config_loader.ts:L69-L77. Passare un NgModule a `loadComponent` lancia `INVALID_ROUTE_CONFIG` ("Use 'loadChildren' instead") — angular:packages/router/src/utils/config.ts:L73-L79.

Un componente standalone caricato così non porta con sé un `NgModuleRef`: il suo injector viene creato all'attivazione. `createRootViewInjector` chiama `componentDef.getStandaloneInjector(realEnvironmentInjector)` — angular:packages/core/src/render3/component_ref.ts:L150-L153 — che chiede `StandaloneService` al parent environment injector (angular:packages/core/src/render3/definition.ts:L360-L364); questo crea, se il componente importa provider, un injector con `createEnvironmentInjector([providers], this._injector, ...)` — angular:packages/core/src/render3/standalone_service.ts:L31-L43. `StandaloneService` è `providedIn: 'environment'`, quindi ogni environment injector (anche quello di un modulo lazy) ha la propria cache — angular:packages/core/src/render3/standalone_service.ts:L61-L66.

3.6 Le API core: `Compiler`, `NgModuleFactory`, `createNgModule`, `createEnvironmentInjector`

Il router (e Spartacus) passano per `Compiler.compileModuleAsync`. In Ivy questa API non compila nulla:

- `Compiler` è `providedIn: 'root'` e `@deprecated` ("Ivy JIT mode doesn't require accessing this symbol") — angular:packages/core/src/linker/compiler.ts:L46-L52;
- `compileModuleSync` restituisce `new NgModuleFactoryR3(moduleType)`, `compileModuleAsync` lo avvolge in `Promise.resolve` — angular:packages/core/src/linker/compiler.ts:L57-L66; `clearCache`/`clearCacheFor` sono no-op — angular:packages/core/src/linker/compiler.ts:L97-L109;
- la `NgModuleFactory` R3 ha come unico stato `moduleType` e `create(parentInjector)` esegue `new NgModuleRef(this.moduleType, parentInjector, [])` — angular:packages/core/src/render3/ng_module_ref.ts:L127-L135;

- `createNgModule(ngModule, parentInjector?)` è `@publicApi` ed esegue **esattamente** `new NgModuleRef(ngModule, parentInjector ?? null, [])`; senza parent si usa il `NullInjector` — angular:packages/core/src/render3/ng_module_ref.ts:L27-L42. `createNgModuleRef` ne è un alias deprecato — angular:packages/core/src/render3/ng_module_ref.ts:L44-L50;
- la `NgModuleFactory` astratta del linker è deprecata e suggerisce `createNgModule` — angular:packages/core/src/linker/ng_module_factory.ts:L62-L75.

Quindi `compiler.compileModuleAsync(M).then(f => f.create(parent))` ≡ `createNgModule(M, parent)`, a meno di un microtask.

Cosa succede in `new NgModuleRef(...)`:

1. crea l'`R3Injector` con `createInjectorWithoutInjectorInstances`, registrando `NgModuleRef` e `ComponentFactoryResolver` come `useValue` e lo scope `'environment'` — angular:packages/core/src/render3/ng_module_ref.ts:L82-L95;
2. `createInjectorWithoutInjectorInstances` raccoglie i provider con `importProvidersFrom(defType)` e usa come parent `parent || getNullInjector()` — angular:packages/core/src/di/create_injector.ts:L42-L56;
3. per ogni NgModule del grafo degli import, `walkProviderTree` aggiunge un provider della classe, un `INJECTOR_DEF_TYPES` e un `ENVIRONMENT_INITIALIZER` che fa `inject(defType)`: i costruttori di tutti i moduli importati vengono eseguiti alla creazione — angular:packages/core/src/di/provider_collection.ts:L333-L343;
4. risolve gli injector initializer separatamente (per evitare cicli se il modulo inietta il proprio ref) e ottiene l'istanza del modulo — angular:packages/core/src/render3/ng_module_ref.ts:L97-L108. `resolveInjectorInitializers` legge `ENVIRONMENT_INITIALIZER` con `self: true` e li invoca in modo **sincrono** — angular:packages/core/src/di/r3_injector.ts:L426-L448.

`createEnvironmentInjector(providers, parent, debugName)` è l'equivalente "senza modulo": crea un `EnvironmentNgModuleRefAdapter` con `runEnvironmentInitializers: true` e ne restituisce l'injector — angular:packages/core/src/render3/ng_module_ref.ts:L183-L205. L'adapter crea direttamente un `R3Injector` dai provider (nessuna classe modulo, `instance = null`), scope `'environment'`, parent `config.parent || getNullInjector()` — angular:packages/core/src/render3/ng_module_ref.ts:L145-L172. A differenza di `createNgModule`, il parent è tipizzato `EnvironmentInjector` e non un `Injector` qualsiasi — angular:packages/core/src/render3/ng_module_ref.ts:L37-L40, angular:packages/core/src/render3/ng_module_ref.ts:L193-L196. Questa differenza è rilevante per Spartacus, che a volte usa come parent un `CombinedInjector` che implementa solo `Injector` (§3.10).

Il `R3Injector` con scope `'environment'` registra `INJECTOR` ed `EnvironmentInjector` su se stesso — angular:packages/core/src/di/r3_injector.ts:L228-L234; crea al volo i record per gli `@Injectable` con `providedIn` corrispondente a uno dei suoi scope — angular:packages/core/src/di/r3_injector.ts:L347-L366; e, se non trova il token, delega a `this.parent.get(token, notFoundValue)` **senza** passare le `InjectOptions` — angular:packages/core/src/di/r3_injector.ts:L374-L383.

3.7 Come il componente "vede" l'injector lazy

Avere un `NgModuleRef` non basta: i componenti devono essere creati con il suo injector come **environment injector**.

- Nel router, `OutletContext.injector` è lo `_environmentInjector` dello snapshot attivo (che per i figli di una route lazy è `_loadedInjector`, §3.2) o, in mancanza, il root — angular:packages/router/src/router_outlet_context.ts:L24-L26. `ActivateRoutes` lo passa a `activateWith` — angular:packages/router/src/operators/activate_routes.ts:L206-L206 — che crea il componente con `location.`

`createComponent(component, {index, injector, environmentInjector})` — angular:packages/router/src/directives/router_outlet.ts:L380-L404.

- In generale `createComponent` accetta `environmentInjector` **oppure** `ngModuleRef` (mai entrambi: errore in dev) — angular:packages/core/src/linker/view_container_ref.ts:L518-L526; se nessuno dei due è passato **e la factory non ha** `ngModule` lo cerca con `get(EnvironmentInjector, null)` — angular:packages/core/src/linker/view_container_ref.ts:L536-L563. La firma deprecata con factory accetta come quinto argomento un `EnvironmentInjector | NgModuleRef` — angular:packages/core/src/linker/view_container_ref.ts:L263-L275.
- `ComponentFactory.create` usa `environmentInjector || this.ngModule` — angular:packages/core/src/render3/component_ref.ts:L282-L286; `createRootViewInjector` estrae `.injector` da un `NgModuleRef` e restituisce `new ChainedInjector(injector, realEnvironmentInjector)` — angular:packages/core/src/render3/component_ref.ts:L140-L159. `ChainedInjector.get` interroga prima l'element injector, poi l'environment injector — angular:packages/core/src/render3/chained_injector.ts:L18-L44.

3.8 Preloading

- Di default il preloading è assente: `PreloadingStrategy` è una classe astratta (angular:packages/router/src/router_preloader.ts:L26-L28), `ROUTER_PRELOADER` è fornito solo da `withPreloading` (angular:packages/router/src/provide_router.ts:L629-L633), a cui `RouterModule.forRoot` ricorre solo se `ExtraOptions.preloadingStrategy` è impostato (angular:packages/router/src/router_module.ts:L159-L159), e `setUpPreloading` viene chiamato solo se il token esiste (angular:packages/router/src/provide_router.ts:L335-L335). Il JSDoc descrive questo comportamento come `NoPreloading`, che non carica nulla — angular:packages/router/src/router_preloader.ts:L62-L78; `PreloadAllModules` carica tutto ignorando gli errori (`catchError(() => of(null))`) — angular:packages/router/src/router_preloader.ts:L55-L60. Entrambe, insieme a `withPreloading`, sono esportate da `@angular/router` — angular:packages/router/src/index.ts:L98-L126.
- `RouterPreloader.setUpPreloading` avvia il preload dopo ogni `NavigationEnd`, serializzato con `concatMap` — angular:packages/router/src/router_preloader.ts:L103-L110.
- Il preloader crea anch'esso gli injector dei `providers` e ricrea `_loadedInjector` dalla factory se manca — angular:packages/router/src/router_preloader.ts:L124-L137; salta le route `loadChildren` con `canLoad` definito (i guard possono avere side effect) — angular:packages/router/src/router_preloader.ts:L139-L152; non carica nulla se l'injector è già distrutto e poi ricorre sulle route figlie — angular:packages/router/src/router_preloader.ts:L160-L183.

3.9 Distruzione degli injector di route (sperimentale)

Di default il router non distrugge mai gli injector lazy. In v21.1 è stato introdotto l'opt-in `withExperimentalAutoCleanupInjectors()` (`@experimental 21.1`), che fornisce `ROUTE_INJECTOR_CLEANUP` e richiede che `RouteReuseStrategy.shouldDestroyInjector` restituisca `true` — angular:packages/router/src/provide_router.ts:L795-L810. Il `Router` lo inietta opzionalmente e lo invoca a ogni `NavigationEnd` — angular:packages/router/src/router.ts:L148-L148, angular:packages/router/src/router.ts:L211-L213. `destroyUnusedInjectors` distrugge `_injector` e `_loadedInjector` delle route non attive per cui la strategia acconsente, propagando ai figli — angular:packages/router/src/route_injector_cleanup.ts:L66-L103. `BaseRouteReuseStrategy.shouldDestroyInjector` restituisce `true` di default — angular:packages/router/src/route_reuse_strategy.ts:L140-L150. È per questo che `getChildConfig` e il preloader sanno ricreare `_loadedInjector` dalla factory (§3.3, §3.8).

3.10 Spartacus: il caricatore fuori dal router

Il dettaglio è nel capitolo 12; qui i passi, messi in parallelo con quelli del router (§3.3-§3.7).

1. **Dichiarazione.** `FeatureModuleConfig` ha `module?: () => Promise<any>`, `dependencies?` (funzioni o nomi di feature) e `cmsComponents?`; `CmsConfig.featureModules` mappa nome → config o alias stringa — core-libs/core/src/cms/config/cms-config.ts:L109-L129. Esempio app demo: `module: () => import('@spartacus/quote').then((m) => m.QuoteModule)` — projects/storefrontapp/src/app/spartacus/features/quote-feature.module.ts:L26-L31. Equivale alla callback di `loadChildren`, ma è letta da una config, non da una `Route`.
2. **Trigger.** Non è una navigazione ma il rendering CMS: `CmsFeaturesService.initFeatureMap` costruisce la mappa componentType → feature dalla config stabile `featureModules` — core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L75; `CmsComponentsService.determineMappings` attende con `forkJoin` la risoluzione delle feature dei componenti della pagina prima di proseguire — core-libs/storefront/cms-structure/services/cms-components.service.ts:L92-L113. Altri trigger: le proxy facade, che alla prima chiamata risolvono la feature e prendono la facade concreta con `moduleRef.injector.get(facadeClass)` — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L56.
3. **Cache e dedup** (analogo di `WeakMap` + `_loadedRoutes`). `FeatureModulesService.resolveFeature` risolve l'alias e mette in cache per nome un Observable `resolveDependencies → resolveModuleInstance` con `shareReplay()` — core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L72. Una feature non configurata produce un errore "No module defined for Feature Module ..." — core-libs/core/src/lazy-loading/feature-modules.service.ts:L47-L52. `LazyModulesService.resolveModuleInstance` invece, per sua stessa documentazione, crea un'istanza nuova a ogni chiamata — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L70-L77.
4. **Dipendenze** (nessun equivalente nel router). Risolte in parallelo con `forkJoin`: nome → `resolveFeature`, funzione → `resolveDependencyModuleInstance` — core-libs/core/src/lazy-loading/feature-modules.service.ts:L108-L122; i moduli di dipendenza sono creati una sola volta (mappa per classe) con parent root — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L121-L129.
5. **Scelta del parent** (nel router: `route._injector ?? injector`). Root se non ci sono dipendenze, l'injector dell'unica dipendenza, o un `CombinedInjector(root, [injector delle dipendenze])` se sono più d'una — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L94.
6. **Import e "compilazione".** `resolveModuleFactory` esegue `moduleFunc()` **fuori** da un contesto di iniezione e senza spacchettare `default`; accetta un `NgModuleFactory` già pronto oppure passa la classe a `compileModuleAsync` (stesso ramo del router, §3.4 punto 3); poi riprogramma l'emissione su `queueScheduler` — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L235. Il `NgModuleFactory` importato è quello astratto del linker, quindi `instanceof` riconosce anche le factory R3 — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L7-L15, angular:packages/core/src/linker.ts:L21-L21.
7. **Istanza.** `moduleFactory.create(parentInjector)` (= `createNgModule(M, parent)`, §3.6) — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L97-L97.
8. **Initializer asincroni** (nessun equivalente nel router). `runModuleInitializersForModule` legge `MODULE_INITIALIZER` con `{ self: true }` — lo stesso accorgimento del router con `ROUTES` — e attende le Promise con `Promise.all` — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L171. `ENVIRONMENT_INITIALIZER`, al contrario, non è atteso e non deve essere `async` — angular:packages/core/src/di/initializer_token.ts:L11-L24.
9. **Notifica.** Si emette `ModuleInitializedEvent` (`feature?`, `moduleRef`) — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L99-L106, core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L31; `modules$` è un `connectable` con `ReplaySubject` collegato nel costruttore, così anche i

sottoscrittori tardivi ricevono tutti i moduli — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L55, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L65-L67. Nel router le uniche notifiche sono gli eventi `RouteConfigLoadStart` / `RouteConfigLoadEnd` — angular:packages/router/src/navigation_transition.ts:L390-L391.

10. **Rendering** (= `RouterOutlet.activateWith`). `ComponentWrapperDirective.launchComponent` passa all'handler l'element injector di `CmsInjectorService` e il `NgModuleRef` della feature — core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L90-L102. Il modulo è ottenuto da `CmsComponentsService.getModule`, che delega a `CmsFeaturesService.getModule` solo se `hasFeatureFor(componentType)` (altrimenti restituisce `undefined`) — core-libs/storefront/cms-structure/services/cms-components.service.ts:L149-L153; `CmsFeaturesService.getModule` lo restituisce in modo sincrono solo per feature già risolte — core-libs/storefront/cms-structure/services/cms-features.service.ts:L110-L127. `DefaultComponentHandler` chiama la firma deprecata `createComponent(factory, undefined, injector, undefined, module)` — core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L56-L68 — e per §3.7 il risultato è `ChainedInjector(elementInjector, module.injector)`.
11. **Config della feature.** La config CMS della feature è letta direttamente dall'injector del modulo (`ConfigChunk`, `DefaultConfigChunk` con `self: true`, fusi con `deepMerge`) — core-libs/storefront/cms-structure/services/cms-features.service.ts:L188-L204.

3.11 Spartacus: route e guard restano nel root

Poiché nessun modulo Spartacus è caricato dal router, le route delle feature sono dichiarate con `RouterModule.forChild` nei moduli `root` **eager** e finiscono nel `ROUTES` dell'injector root; il modulo lazy contiene solo l'implementazione. Esempio order, con `path: null` risolto da `cxRoute` — feature-libs/order/root/order-root.module.ts:L82-L90. Il router, infatti, inizializza `config` con `inject(ROUTES, {optional: true})?.flat()` dell'injector in cui vive (il root) — angular:packages/router/src/router.ts:L163-L163 — e legge `ROUTES` di un modulo lazy solo nel flusso `loadChildren` (§3.4 punto 5).

Anche le route CMS aggiunte a runtime non usano `loadChildren`: `updateRouting` antepone una nuova route con `component: PageLayoutComponent` e `children` statici tramite `router.resetConfig` — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L107-L119. Le route figlie quindi "appartengono allo stesso modulo" del root (§3.3 caso 1) e i loro guard sarebbero risolti nel root injector. Per questo Spartacus avvolge ogni `canActivate` in una `CanActivateFn` che risolve il guard tramite `UnifiedInjector` ("even when they are provided only in a child injector of a lazy-loaded module") — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L126-L136 — usando `getLastValueSync(unifiedInjector.get(guard))` e trattando il token come `CanActivateFn` se non trova un'istanza — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L174-L180. Anche `CmsGuardsService.cmsPageCanActivate` risolve i guard dei mapping CMS tramite `UnifiedInjector` + `getLastValueSync`, ma senza fallback a `CanActivateFn`: i guard non trovati (o non istanze `CanActivate`) vengono scartati con `filter(isCanActivate)` e non eseguiti — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L42-L46.

`UnifiedInjector.injectors$` parte dal root e aggiunge l'injector di ogni modulo emesso da `modules$` — core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35; `get` interroga il root a tutti i livelli e gli injector lazy solo con `{ self: true }` — core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67. `getLastValueSync` si sottoscrive e disiscrive subito restituendo l'ultima emissione sincrona — core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23.

3.12 Spartacus: lazy loading di singoli componenti

Oltre ai moduli, un mapping CMS può avere `component: () => import(...)`. `LazyComponentHandler` lo riconosce analizzando i primi 20 caratteri di `toString()` (`function()` o `()=>`) — core-libs/storefront/cms-

structure/page/component/handlers/lazy-component.handler.ts:L38-L48; registra l'`import()` come `PendingTasks` (rimosso in `finalize`) per evitare che l'SSR serializzi e distrugga l'app mentre il chunk è in volo (NG0205), poi delega a `DefaultComponentHandler` passando il `module` — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L80. È l'equivalente CMS di `loadComponent`, con due differenze: non c'è esecuzione in contesto di iniezione e il componente viene creato con l'environment injector della feature (se risolta), non con uno standalone injector della route.

3.13 Confronto riassuntivo

ASPETTO	ROUTER ANGULAR (LOADCHILDREN)	SPARTACUS (FEATUREMODULES)
Trigger	Recognize di una navigazione / preloader (angular:packages/router/src/recognize.ts:L514-L527)	Rendering CMS, facade proxy (core-libs/storefront/cms-structure/services/cms-components.service.ts:L92-L113)
Unità caricabile	Classe NgModule, `NgModuleFactory`, `Routes`, `default` export (angular:packages/router/src/models.ts:L267-L274)	Classe NgModule o `NgModuleFactory` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L225-L232)
Contesto di iniezione della callback	Sì, `runInInjectionContext` (angular:packages/router/src/router_config_loader.ts:L126-L129)	No (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L223-L223)
Unwrap `default`	Sì (angular:packages/router/src/router_config_loader.ts:L165-L176)	No
Istanziazione	`compileModuleAsync` + `factory.create(parent)` (angular:packages/router/src/router_config_loader.ts:L131-L151)	Identica (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L97)
Parent	`route._injector ?? injector` (angular:packages/router/src/recognize.ts:L438-L441)	Root / dipendenza / `CombinedInjector` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L94)
Cache	Campi della `Route` + `WeakMap` (angular:packages/router/src/router_config_loader.ts:L91-L102)	`Map` per nome + `shareReplay` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L56-L68)
Lettura token "self"	`ROUTES` (angular:packages/router/src/router_config_loader.ts:L157-L157)	`MODULE_INITIALIZER`, `ConfigChunk` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L153-L157)
Initializer async	No (`ENVIRONMENT_INITIALIZER` sincrono)	Sì, `MODULE_INITIALIZER` atteso
Uso dell'injector	Environment injector dei componenti attivati (angular:packages/router/src/directives/router_outlet.ts:L380-L404)	Environment injector dei componenti CMS + `UnifiedInjector` (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L62-L68)
Distruzione	Opt-in sperimentale (angular:packages/router/src/provide_router.ts:L795-L810)	Solo dependency module in `ngOnDestroy` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L238-L245)
Preloading	`PreloadingStrategy`	Nessuno

4. Diagrammi

4.1 Navigazione verso una route `loadChildren` (NgModule)

■ **TAVOLA 03.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti dei passi: matching e injector di route §3.2; `getChildConfig` §3.3; loader §3.4; `Compiler`/factory/`NgModuleRef` §3.6; attivazione §3.7.

4.2 Gerarchia di injector: router vs Spartacus

■ **TAVOLA 03.2** diagramma a tutta pagina, in fondo al capitolo

La freccia tratteggiata `modules$` indica che il root non diventa parent del modulo in senso inverso: è `UnifiedInjector` che interroga l'injector della feature con `self` (§3.11). La scelta del parent è in `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L94`; la catena dei componenti in `angular:packages/core/src/render3/component_ref.ts:L140-L159`.

4.3 Componente CMS di una feature lazy (Spartacus)

■ **TAVOLA 03.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: §3.10 passi 2-10.

5. Estratti di codice reale

5.1 La funzione `loadChildren` del router

`angular:packages/router/src/router_config_loader.ts:L120-L163`

TS

```ts
export async function loadChildren(
  route: Route,
  compiler: Compiler,
  parentInjector: Injector,
  onLoadEndListener?: (r: Route) => void,
): Promise<LoadedRouterConfig> {
  const loaded = await wrapIntoPromise(
    runInInjectionContext(parentInjector, () => route.loadChildren!()),
  );
  const t = await maybeResolveResources(maybeUnwrapDefaultExport(loaded));

  let factoryOrRoutes: NgModuleFactory<any> | Routes;
  if (t instanceof NgModuleFactory || Array.isArray(t)) {
```

```
    factoryOrRoutes = t;
  } else {
    factoryOrRoutes = await compiler.compileModuleAsync(t);
  }

  if (onLoadEndListener) {
    onLoadEndListener(route);
  }
  // This injector comes from the `NgModuleRef` when lazy loading an `NgModule`. There is
  // no injector associated with lazy loading a `Route` array.
  let injector: EnvironmentInjector | undefined;
  let rawRoutes: Route[];
  let requireStandaloneComponents = false;
  let factory: NgModuleFactory<unknown> | undefined = undefined;
  if (Array.isArray(factoryOrRoutes)) {
    rawRoutes = factoryOrRoutes;
    requireStandaloneComponents = true;
  } else {
    injector = factoryOrRoutes.create(parentInjector).injector;
    factory = factoryOrRoutes;
    // When loading a module that doesn't provide `RouterModule.forChild()` preloader
    // will get stuck in an infinite loop. The child module's Injector will look to
    // its parent `Injector` when it doesn't find any ROUTES so it will return routes
    // for it's parent module instead.
    rawRoutes = injector.get(ROUTES, [], {optional: true, self: true}).flat();
  }
  const routes = rawRoutes.map(standardizeConfig);
  (typeof ngDevMode === 'undefined' || ngDevMode) &&
    validateConfig(routes, route.path, requireStandaloneComponents);
  return {routes, injector, factory};
}
```

5.2 `RouterConfigLoader.loadChildren`: dedup e cache sulla `Route`

angular:packages/router/src/router_config_loader.ts:L81-L109

TS

```
  loadChildren(parentInjector: Injector, route: Route): Promise<LoadedRouterConfig> {
    if (this.childrenLoaders.get(route)) {
      return this.childrenLoaders.get(route)!;
    } else if (route._loadedRoutes) {
      return Promise.resolve({routes: route._loadedRoutes, injector: route._loadedInjector});
    }

    if (this.onLoadStartListener) {
      this.onLoadStartListener(route);
    }
    const loader = (async () => {
      try {
```

```
      const result = await loadChildren(
        route,
        this.compiler,
        parentInjector,
        this.onLoadEndListener,
      );
      route._loadedRoutes = result.routes;
      route._loadedInjector = result.injector;
      route._loadedNgModuleFactory = result.factory;
      return result;
    } finally {
      this.childrenLoaders.delete(route);
    }
  })();
  this.childrenLoaders.set(route, loader);
  return loader;
}
```

5.3 `getChildConfig` nel recognizer

angular:packages/router/src/recognize.ts:L494-L530

TS

```
private async getChildConfig(
  injector: EnvironmentInjector,
  route: Route,
  segments: UrlSegment[],
): Promise<LoadedRouterConfig> {
  if (route.children) {
    // The children belong to the same module
    return {routes: route.children, injector};
  }

  if (route.loadChildren) {
    // lazy children belong to the loaded module
    if (route._loadedRoutes !== undefined) {
      const ngModuleFactory = route._loadedNgModuleFactory;
      if (ngModuleFactory && !route._loadedInjector) {
        route._loadedInjector = ngModuleFactory.create(injector).injector;
      }
      return {routes: route._loadedRoutes, injector: route._loadedInjector};
    }

    if (this.abortSignal.aborted) {
      throw new Error(this.abortSignal.reason);
    }
    const shouldLoadResult = await firstValueFrom(
      runCanLoadGuards(injector, route, segments, this.urlSerializer, this.abortSignal),
    );
```

```
      if (shouldLoadResult) {
        const cfg = await this.configLoader.loadChildren(injector, route);
        route._loadedRoutes = cfg.routes;
        route._loadedInjector = cfg.injector;
        route._loadedNgModuleFactory = cfg.factory;
        return cfg;
      }
      throw canLoadFails(route);
    }

    return {routes: [], injector};
```

5.4 Injector dei `providers` di route

angular:packages/router/src/utils/config.ts:L31-L43

TS

```
export function getOrCreateRouteInjectorIfNeeded(
  route: Route,
  currentInjector: EnvironmentInjector,
): EnvironmentInjector {
  if (route.providers && !route._injector) {
    route._injector = createEnvironmentInjector(
      route.providers,
      currentInjector,
      `Route: ${route.path}`,
    );
  }
  return route._injector ?? currentInjector;
}
```

5.5 `Compiler`, `NgModuleFactory` R3 e `createNgModule`: la stessa cosa

angular:packages/core/src/linker/compiler.ts:L57-L66

TS

```
  compileModuleSync<T>(moduleType: Type<T>): NgModuleFactory<T> {
    return new NgModuleFactoryR3(moduleType);
  }

  /**
   * Compiles the given NgModule and all of its components
   */
  compileModuleAsync<T>(moduleType: Type<T>): Promise<NgModuleFactory<T>> {
    return Promise.resolve(this.compileModuleSync(moduleType));
  }
```

angular:packages/core/src/render3/ng_module_ref.ts:L127-L135

```ts
export class NgModuleFactory<T> extends viewEngine_NgModuleFactory<T> {
  constructor(public moduleType: Type<T>) {
    super();
  }

  override create(parentInjector: Injector | null): viewEngine_NgModuleRef<T> {
    return new NgModuleRef(this.moduleType, parentInjector, []);
  }
}
```

angular:packages/core/src/render3/ng_module_ref.ts:L37-L42

```ts
export function createNgModule<T>(
  ngModule: Type<T>,
  parentInjector?: Injector,
): viewEngine_NgModuleRef<T> {
  return new NgModuleRef<T>(ngModule, parentInjector ?? null, []);
}
```

5.6 `createEnvironmentInjector`

angular:packages/core/src/render3/ng_module_ref.ts:L193-L205

```ts
export function createEnvironmentInjector(
  providers: Array<Provider | EnvironmentProviders>,
  parent: EnvironmentInjector,
  debugName: string | null = null,
): EnvironmentInjector {
  const adapter = new EnvironmentNgModuleRefAdapter({
    providers,
    parent,
    debugName,
    runEnvironmentInitializers: true,
  });
  return adapter.injector;
}
```

5.7 `createRootViewInjector`: da `NgModuleRef` a `ChainedInjector`

angular:packages/core/src/render3/component_ref.ts:L140-L159

```ts
function createRootViewInjector(
  componentDef: ComponentDef<unknown>,
  environmentInjector: EnvironmentInjector | NgModuleRef<any> | undefined,
  injector: Injector,
): Injector {
  let realEnvironmentInjector =
```

```
    environmentInjector instanceof EnvironmentInjector
      ? environmentInjector
      : environmentInjector?.injector;

  if (realEnvironmentInjector && componentDef.getStandaloneInjector !== null) {
    realEnvironmentInjector =
      componentDef.getStandaloneInjector(realEnvironmentInjector) || realEnvironmentInjector;
  }

  const rootViewInjector = realEnvironmentInjector
    ? new ChainedInjector(injector, realEnvironmentInjector)
    : injector;
  return rootViewInjector;
}
```

5.8 Spartacus: `resolveModuleInstance`

core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L108

TS

```
  public resolveModuleInstance(
    moduleFunc: () => Promise<any>,
    feature?: string,
    dependencyModuleRefs: NgModuleRef<any>[] = []
  ): Observable<NgModuleRef<any>> {
    let parentInjector: Injector;

    if (!dependencyModuleRefs.length) {
      parentInjector = this.injector;
    } else if (dependencyModuleRefs.length === 1) {
      parentInjector = dependencyModuleRefs[0].injector;
    } else {
      parentInjector = new CombinedInjector(
        this.injector,
        dependencyModuleRefs.map((moduleRef) => moduleRef.injector)
      );
    }

    return this.resolveModuleFactory(moduleFunc).pipe(
      map(([moduleFactory]) => moduleFactory.create(parentInjector)),
      concatMap((moduleRef) => this.runModuleInitializersForModule(moduleRef)),
      tap((moduleRef) =>
        this.events.dispatch(
          createFrom(ModuleInitializedEvent, {
            feature,
            moduleRef,
          })
        )
```

```
    )
  );
}
```

5.9 Spartacus: `resolveModuleFactory` (il ramo identico al router)

core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L236

```ts
private resolveModuleFactory(
  moduleFunc: () => Promise<any>
): Observable<[NgModuleFactory<any>, any]> {
  return from(moduleFunc()).pipe(
    switchMap((module) =>
      module instanceof NgModuleFactory
        ? (of([module, module]) as Observable<[NgModuleFactory<any>, any]>)
        : combineLatest([
            // using compiler here is for jit compatibility, there is no overhead
            // for aot production builds as it will be stubbed
            from(this.compiler.compileModuleAsync(module as any)),
            of(module),
          ])
    ),
    observeOn(queueScheduler)
  );
}
```

Confrontalo con le righe `L131-L136` di §5.1: stesso test `instanceof NgModuleFactory`, stessa chiamata `compileModuleAsync`. Mancano `runInInjectionContext` e `maybeUnwrapDefaultExport`.

5.10 Spartacus: `MODULE_INITIALIZER` letto con `self`

core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L175

```ts
public runModuleInitializersForModule(
  moduleRef: NgModuleRef<any>
): Observable<NgModuleRef<any>> {
  const moduleInits: any[] = moduleRef.injector.get<any[]>(
    MODULE_INITIALIZER,
    [],
    { self: true }
  );
  const asyncInitPromises: Promise<any>[] =
    this.runModuleInitializerFunctions(moduleInits);
  if (asyncInitPromises.length) {
    return from(Promise.all(asyncInitPromises)).pipe(
      tap({
        error: (error) => {
          this.logger.error(
```

```
            'MODULE_INITIALIZER promise was rejected while lazy loading a module.',
            error
          );
        },
      }),
      switchMap(() => of(moduleRef))
    );
  } else {
    return of(moduleRef);
  }
}
```

5.11 Spartacus: `CombinedInjector.get`

core-libs/core/src/util/combined-injector.ts:L43-L71

TS

```
get(token: any, notFoundValue?: any, optional?: InjectOptions): any {
  if (optional && optional.self) {
    if (notFoundValue !== undefined) {
      return notFoundValue;
    }
    throw new Error(
      "CombinedInjector should be used as a parent injector / doesn't support self dependencies"
    );
  }

  for (const injector of this.complementaryInjectors) {
    // First we are resolving providers provided at Self level
    // in all complementary injectors...
    const service = injector.get(token, NOT_FOUND_SYMBOL, { self: true });
    if (service !== NOT_FOUND_SYMBOL) {
      return service;
    }
  }

  for (const injector of this.complementaryInjectors) {
    // next we try to resolve tokens from all levels
    const service = injector.get(token, NOT_FOUND_SYMBOL);
    if (service !== NOT_FOUND_SYMBOL) {
      return service;
    }
  }
  // ...and then fallback to main injector
  return this.mainInjector.get(token, notFoundValue);
}
```

5.12 Spartacus: modulo lazy come environment injector del componente CMS

core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L56-L70

```ts
    const factory = this.getComponentFactory(
      injector,
      componentMapping.component
    );

    if (factory) {
      componentRef = viewContainerRef.createComponent(
        factory,
        undefined,
        injector,
        undefined,
        module
      );
      subscriber.next({ elementRef: componentRef.location, componentRef });
    }
```

5.13 Spartacus: guard di route risolti con `UnifiedInjector`

core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L170-L181

```ts
  return (
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot
  ): Observable<GuardResult> => {
    const classGuard = getLastValueSync(
      this.unifiedInjector.get<CanActivate>(
        guard as InjectionToken<CanActivate>
      )
    );
    const canActivate = classGuard ?? { canActivate: guard as CanActivateFn };
    return this.guardsComposer.canActivate([canActivate], route, state);
  };
```

5.14 Spartacus: `import()` di un componente come `PendingTasks`

core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71

```ts
  return defer(() => {
    // Register the dynamic import as a pending task so Angular's stability
    // counter stays > 0 until the chunk is loaded. This prevents SSR from
    // serializing and destroying the app while the import() is still in flight
    // (which would cause NG0205 — destroyed injector). The task is removed by
    // finalize(), which fires on complete, error, and unsubscribe.
```

```
      const removeTaskTracking = this.pendingTasks.add();
      return from(componentMapping.component()).pipe(
        finalize(removeTaskTracking)
      );
    }).pipe(
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`Route.loadChildren`	`LoadChildren` (callback)	nessuno	angular:packages/router/src/models.ts:L738-L738	angular:packages/router/src/router_config_loader.ts:L127-L127, angular:packages/router/src/recognize.ts:L504-L504, angular:packages/router/src/router_preloader.ts:L148-L148
`Route.loadComponent`	`() => Type | Observable | Promise`	nessuno	angular:packages/router/src/models.ts:L626-L629	angular:packages/router/src/router_config_loader.ts:L62-L62, angular:packages/router/src/navigation_transition.ts:L723-L727
`Route.providers`	`Array<Provider | EnvironmentProviders>`	nessuno	angular:packages/router/src/models.ts:L769-L769	angular:packages/router/src/utils/config.ts:L35-L40, angular:packages/router/src/router_preloader.ts:L124-L129
`Route.canLoad` (deprecato)	array di guard	nessuno	angular:packages/router/src/models.ts:L715-L715	angular:packages/router/src/operators/check_guards.ts:L237-L240, angular:packages/router/src/router_preloader.ts:L148-L148
`PreloadingStrategy` (via `ExtraOptions.preloadingStrategy` / `withPreloading`)	classe	nessuna (senza la chiave `ROUTER_PRELOADER` non è registrato; il JSDoc lo chiama `NoPreloading`)	angular:packages/router/src/router_preloader.ts:L26-L28	angular:packages/router/src/provide_router.ts:L629-L633, angular:packages/router/src/router_module.ts:L159-L159, angular:packages/router/src/provide_router.ts:L335-L335; Spartacus non lo configura

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`withExperimentalAutoCleanupInjectors()`	feature opt-in, `@experimental 21.1`	disattivo	angular:packages/router/src/provide_router.ts:L806-L810	angular:packages/router/src/router.ts:L148-L148
`RouterModule.forRoot` options (Spartacus)	`ExtraOptions`	`anchorScrolling: 'enabled', initialNavigation: 'enabledBlocking'`	core-libs/storefront/router/app-routing.module.ts:L15-L18	router Angular
`featureModules` (Spartacus)	`{ [name]: FeatureModuleConfig | string }`	assente	core-libs/core/src/cms/config/cms-config.ts:L129-L129	core-libs/core/src/lazy-loading/feature-modules.service.ts:L83-L97, core-libs/storefront/cms-structure/services/cms-features.service.ts:L57-L60
`FeatureModuleConfig.module` / `.dependencies` / `.cmsComponents`	funzione `import()` / array / `string[]`	assenti	core-libs/core/src/cms/config/cms-config.ts:L113-L121	core-libs/core/src/lazy-loading/feature-modules.service.ts:L58-L63, core-libs/storefront/cms-structure/services/cms-features.service.ts:L66-L72

Per la configurazione completa di `featureModules` (alias, `cmsComponents`, convenzioni delle feature lib) vedi capitolo 12 e 11 — Configurazione.

7. Commenti e note notevoli dal sorgente

- angular:packages/core/src/linker/compiler.ts:L46-L49 — `Compiler` è `@deprecated`: "Ivy JIT mode doesn't require accessing this symbol." Eppure il router stesso lo inietta (angular:packages/router/src/router_config_loader.ts:L47-L47).
- angular:packages/core/src/linker/ng_module_factory.ts:L62-L71 — `NgModuleFactory` deprecata, invita a usare API che accettano la classe (`createNgModule`).
- angular:packages/core/src/render3/ng_module_ref.ts:L44-L50 — `createNgModuleRef` `@deprecated Use createNgModule instead.`
- angular:packages/core/src/linker/ng_module_factory_loader.ts:L16-L22 — `getModuleFactory` `@deprecated Use getNgModuleById instead.`
- angular:packages/core/src/di/initializer_token.ts:L15-L20 — `ENVIRONMENT_INITIALIZER` deprecato dalla v19; le funzioni "are not awaited, hence they should not be `async`".
- angular:packages/router/src/router_config_loader.ts:L153-L156 — il commento sul loop infinito del preloader che giustifica `self: true` nella lettura di `ROUTES`.
- angular:packages/router/src/router_config_loader.ts:L116-L118 — la funzione `loadChildren` è un contratto con l'estrattore di route del prerendering.
- angular:packages/router/src/router_preloader.ts:L139-L146 — "Code splitting and lazy loading is separate from client-side authorization checks and should not be used as a security measure to prevent loading of code."

- angular:packages/router/src/route_reuse_strategy.ts:L146-L146 — typo `@xperimental 21.1`; angular:packages/router/src/models.ts:L789-L789 — typo "retruns" nel JSDoc di `_loadedNgModuleFactory`.
- angular:packages/core/src/linker/view_container_ref.ts:L556-L558 — "DO NOT REFACTOR" sulla ricerca `injector.get(EnvironmentInjector, null)`.
- core-libs/core/src/lazy-loading/lazy-modules.service.ts:L228-L229 — "using compiler here is for jit compatibility, there is no overhead for aot production builds as it will be stubbed". DA VERIFICARE: in Angular v21 `Compiler` non viene "stubbato" dal bundler ma è già un'implementazione banale (§3.6); il commento sembra risalire a View Engine. Non verificabile senza una build (`node_modules` assente).
- core-libs/core/src/lazy-loading/lazy-modules.service.ts:L73-L73 — "New module instance will be created with each call.": la cache non è qui ma in `FeatureModulesService`.
- core-libs/core/src/lazy-loading/tokens.ts:L14-L17 — fallback di `MODULE_INITIALIZER` ad `APP_INITIALIZER` se il modulo è eager (typo "Therfore").
- core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L130-L135 — i guard sono risolti con `UnifiedInjector` "instead of only root injector".
- core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L61-L65 — `PendingTasks` per evitare NG0205 in SSR.
- feature-libs/order/root/order-root.module.ts:L67-L67 — `// TODO: Inline this factory when we start releasing Ivy compiled libraries.`

---

8. Extension point

Angular

- `ROUTES` (`InjectionToken` multi) — angular:packages/router/src/router_config_loader.ts:L37-L39, esportato da `@angular/router` (angular:packages/router/src/index.ts:L118-L118). In Spartacus si contribuisce a `ROUTES` con `RouterModule.forChild` nei moduli eager (§3.11).
- `PreloadingStrategy` / `PreloadAllModules` / `NoPreloading` — angular:packages/router/src/router_preloader.ts:L55-L78. Inutili per le feature Spartacus, che non passano dal router.
- `RouteReuseStrategy.shouldDestroyInjector` (sperimentale) — angular:packages/router/src/route_reuse_strategy.ts:L55-L57, angular:packages/router/src/route_reuse_strategy.ts:L148-L150.
- `createNgModule`, `createEnvironmentInjector`, `provideEnvironmentInitializer` — angular:packages/core/src/core.ts:L101-L103, angular:packages/core/src/di/provider_collection.ts:L81-L88.

Spartacus (pubblici via core-libs/core/public_api.ts:L24-L24)

- `CmsConfig.featureModules` per dichiarare una nuova feature lazy o ridefinire `module`/`dependencies` di una esistente — core-libs/core/src/cms/config/cms-config.ts:L129-L129. Ricetta: 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni.
- `MODULE_INITIALIZER` per inizializzazione (anche asincrona) di un modulo lazy — core-libs/core/src/lazy-loading/tokens.ts:L19-L20. Ricetta: 50 — Ricette: configurazione, DI, interceptor, eventi.
- `LazyModulesService`, `FeatureModulesService` (metodi `protected` `getFeatureConfig`, `resolveFeatureAlias`, `resolveDependencies` — core-libs/core/src/lazy-loading/feature-modules.service.ts:L80-L123), `UnifiedInjector`, `ModuleInitializedEvent`, `facadeFactory` — sostituibili/estendibili via DI.
- Mapping CMS con `component: () => import(...)` (lazy di un singolo componente, §3.12) e guard nei mapping risolti via `UnifiedInjector` — ricetta: 51 — Ricette: componenti CMS, outlet, route e layout.

- `CmsFeaturesService`, `CmsComponentsService` — esportati via core-libs/storefront/cms-structure/services/index.ts:L7-L8.
- **Non** sono API pubbliche: `CombinedInjector` (core-libs/core/src/util/index.ts:L7-L28) e `CmsRoutesImplService` (core-libs/storefront/cms-structure/services/index.ts:L7-L12). Vedi 54 — API pubblica vs interna, upgrade, anti-pattern, checklist sul perché non dipendere da `Compiler`/`NgModuleFactory`/`CombinedInjector` nel codice applicativo.

Si può comunque usare `loadChildren`/`loadComponent` nell'**app** per pagine non CMS (Spartacus non lo impedisce: le route dell'app finiscono nello stesso `ROUTES` root). DA VERIFICARE: l'interazione con la catch-all CMS e con `ConfigurableRoutesService` non è esercitata da nessun esempio nel repo; vedi 05 — Il Router di Angular per l'ordine delle route.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `inject()` **nella callback.** Il router esegue `loadChildren`/`loadComponent` in un contesto di iniezione; `FeatureModuleConfig.module` viene invocata senza contesto: un `inject()` dentro la callback lancerebbe. — angular:packages/router/src/router_config_loader.ts:L127-L127 vs core-libs/core/src/lazy-loading/lazy-modules.service.ts:L223-L223.
2. **Export** `default` **non supportato da Spartacus.** `module: () => import('x')` senza `.then(m => m.X)` passerebbe il namespace del modulo ES a `compileModuleAsync`, che non valida il tipo; l'errore emergerebbe nel costruttore di `NgModuleRef` (assert solo in dev). Usa sempre `.then`, come nell'app demo. — angular:packages/router/src/router_config_loader.ts:L172-L176; core-libs/core/src/lazy-loading/lazy-modules.service.ts:L225-L232; angular:packages/core/src/render3/ng_module_ref.ts:L74-L79; projects/storefrontapp/src/app/spartacus/features/quote-feature.module.ts:L29-L29.
3. **Solo NgModule.** Spartacus accetta solo classi NgModule o factory: niente `Routes` né componenti standalone come unità lazy di feature (a differenza di `loadChildren` con array). — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L225-L232; angular:packages/router/src/router_config_loader.ts:L141-L149.
4. **Moduli re-istanziati.** La deduplicazione di `importProvidersFrom` è locale alla singola chiamata: un modulo importato sia nel root sia nel modulo lazy viene rielaborato nell'injector lazy, con servizi duplicati e costruttori di modulo rieseguiti. È uno dei motivi delle `dependencies` condivise di Spartacus (create una volta). — angular:packages/core/src/di/provider_collection.ts:L151-L156, angular:packages/core/src/di/provider_collection.ts:L340-L343; core-libs/core/src/lazy-loading/lazy-modules.service.ts:L123-L126.
5. `CombinedInjector` **e** `self`**.** `R3Injector` delega al parent senza `InjectOptions`: quando un modulo lazy ha come parent un `CombinedInjector`, il ramo `self` di quest'ultimo non viene raggiunto per delega, e `optional` arriva solo come `notFoundValue = null`. Chiamato direttamente con `self` e senza `notFoundValue`, `CombinedInjector` lancia sempre. — angular:packages/core/src/di/r3_injector.ts:L376-L383; core-libs/core/src/util/combined-injector.ts:L44-L51.
6. **Initializer sincroni vs asincroni.** `ENVIRONMENT_INITIALIZER` non è atteso; per un initializer asincrono in un modulo lazy Spartacus usa `MODULE_INITIALIZER`. Un `MODULE_INITIALIZER` che lancia in modo sincrono viene loggato e rilanciato. — angular:packages/core/src/di/initializer_token.ts:L19-L20; core-libs/core/src/lazy-loading/lazy-modules.service.ts:L187-L207.
7. **Nessuna distruzione delle feature.** Il router non distrugge gli injector lazy salvo opt-in sperimentale; Spartacus in `ngOnDestroy` distrugge solo i dependency module, non i feature module, che restano vivi per tutta la vita dell'app. — angular:packages/router/src/provide_router.ts:L799-L806; core-libs/core/src/lazy-loading/lazy-modules.service.ts:L243-L244.

8. `RouterModule.forChild` **in un modulo lazy Spartacus è ignorato.** `Router.config` legge `ROUTES` solo dal root e il router legge `ROUTES` di un modulo lazy solo nel flusso `loadChildren`: route dichiarate nel modulo caricato da `LazyModulesService` non vengono mai viste. Dichiarale nel modulo `root` eager. — angular:packages/router/src/router.ts:L163-L163; angular:packages/router/src/router_config_loader.ts:L157-L157; feature-libs/order/root/order-root.module.ts:L82-L90.
9. **Guard forniti solo nel modulo lazy.** Su route root il router non li troverebbe: Spartacus li risolve via `UnifiedInjector`, e `getLastValueSync` prende l'**ultima** istanza emessa (l'injector lazy più recente prevale sul root). Se la feature non è ancora caricata, `classGuard` è `undefined` e il token viene trattato come `CanActivateFn`. — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L174-L179; core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23; core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67.
10. `getModule` **può restituire** `undefined`**.** Per feature non ancora risolte `CmsFeaturesService.getModule` restituisce `undefined` e `createComponent` usa prima il `ngModule` della factory, se presente, e solo in sua assenza cerca l'`EnvironmentInjector` nel DI tree. — angular:packages/core/src/linker/view_container_ref.ts:L536-L537; core-libs/storefront/cms-structure/services/cms-features.service.ts:L119-L126; angular:packages/core/src/render3/component_ref.ts:L284-L284.
11. `ngModuleRef` **ed** `environmentInjector` **insieme.** Con la firma a classe di `createComponent` è un errore in dev mode (in produzione non c'è controllo e `environmentInjector` prevale silenziosamente su `ngModuleRef`); Spartacus usa la firma deprecata con factory e passa il modulo come quinto argomento posizionale. — angular:packages/core/src/linker/view_container_ref.ts:L518-L526; core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L62-L68.
12. **Il preloader ignora le route con** `canLoad`, anche se il guard è innocuo; e Spartacus non ha preloading delle feature. — angular:packages/router/src/router_preloader.ts:L147-L152.
13. `createNgModule` **senza parent** usa il `NullInjector`: il modulo non vedrebbe i servizi root. Spartacus passa sempre un parent. — angular:packages/core/src/render3/ng_module_ref.ts:L31-L41; core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L94.
14. **Euristica** `toString()`**.** `LazyComponentHandler` riconosce il lazy import dai primi caratteri del sorgente della funzione: funzioni con nome o forme prodotte da trasformazioni diverse da `function()`/`()=>` non sarebbero riconosciute. — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L45-L48.
15. **SSR e feature module.** `LazyComponentHandler` registra un `PendingTasks`, mentre in `core-libs/core/src/lazy-loading/` non c'è alcun uso di `PendingTasks` (grep allo SHA). DA VERIFICARE: come l'SSR attenda il caricamento dei feature module; probabilmente la stabilità è mantenuta indirettamente da altri task (HTTP, NgRx), ma non è verificabile senza eseguire l'app (`node_modules` assente). Vedi 07 — SSR, hydration e TransferState. — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L67-L70.
16. **Facade non fornita dal modulo lazy.** `FacadeFactoryService` fa `moduleRef.injector.get(facadeClass)`; per la delega al parent (§3.6) se il modulo non fornisce la classe la ricerca risale verso il root. DA VERIFICARE: se ciò restituisca il proxy `providedIn: 'root'` stesso (con possibile ricorsione) non è verificato su un caso reale. — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L56; angular:packages/core/src/di/r3_injector.ts:L374-L383.

---

10. Come fare debug

- **Router.** Abilita `withDebugTracing()` (esportato da `@angular/router`, angular:packages/router/src/index.ts:L85-L101) o filtra `router.events` su `RouteConfigLoadStart` / `RouteConfigLoadEnd`, emessi dal loader —

angular:packages/router/src/navigation_transition.ts:L389-L393. In Spartacus questi eventi non compaiono mai per le feature (nessun `loadChildren`): se li vedi, provengono da route dell'app.

- **Feature Spartacus.** Sottoscrivi `EventService.get(ModuleInitializedEvent)`: l'evento porta `feature` (assente per i dependency module) e `moduleRef` — core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L31. In alternativa `LazyModulesService.modules$`, che rigioca anche i moduli già caricati — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L55.
- **"No module defined for Feature Module X"**: la feature è referenziata (da `cmsComponents`, da una dipendenza o da una facade) ma manca `module` nella config — core-libs/core/src/lazy-loading/feature-modules.service.ts:L47-L52.
- **Componente che non vede un servizio della feature**: verifica che `getModule(flexType)` non sia `undefined` al momento del rendering (§9.10) e che il servizio sia fornito nel modulo lazy e non solo in un modulo eager con scope diverso.
- **Breakpoint utili**: `LazyModulesService.resolveModuleInstance` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L108) per vedere il parent scelto; `NgModuleRef` constructor (angular:packages/core/src/render3/ng_module_ref.ts:L82-L108) per vedere quali moduli vengono istanziati (via `ENVIRONMENT_INITIALIZER`); `DefaultComponentHandler.launcher` per verificare il quinto argomento di `createComponent`.
- **Network tab**: ogni `import()` produce una richiesta a un chunk JS. Un chunk richiesto per una feature in una pagina che non ne mostra componenti indica una facade o una dipendenza che l'ha risolta (§3.10 passo 2).
- **Guard su route CMS**: un breakpoint nel wrapper di `CmsRoutesImplService` (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L170-L181) mostra se `classGuard` è stato trovato o se il token viene trattato come funzione.

---

11. Domande di autoverifica

1. **Quando, durante una navigazione, Angular esegue `loadChildren` e quando `loadComponent`?** `loadChildren` nel recognize (`getChildConfig`), perché servono le route figlie per proseguire il matching; `loadComponent` nella fase "LOAD COMPONENTS", che segue recognize, guard e resolver. — angular:packages/router/src/recognize.ts:L514-L527, angular:packages/router/src/navigation_transition.ts:L717-L739.
2. **Perché il router legge `ROUTES` dal modulo lazy con `self: true`?** Senza `self`, un modulo privo di `RouterModule.forChild()` erediterebbe le route del parent e il preloader entrerebbe in loop infinito. — angular:packages/router/src/router_config_loader.ts:L153-L157.
3. **Che differenza c'è, in Ivy, tra `compiler.compileModuleAsync(M).then(f => f.create(p))` e `createNgModule(M, p)`?** Nessuna sostanziale: entrambe eseguono `new NgModuleRef(M, p, [])`; la prima aggiunge solo un `Promise.resolve`. — angular:packages/core/src/linker/compiler.ts:L57-L66, angular:packages/core/src/render3/ng_module_ref.ts:L37-L42, angular:packages/core/src/render3/ng_module_ref.ts:L132-L134.
4. **Cosa fa il router con l'array `providers` di una route e quale API usa?** Crea una volta un `EnvironmentInjector` con `createEnvironmentInjector(route.providers, currentInjector, 'Route: path')`, salvato in `route._injector`; è il parent di un eventuale modulo lazy. — angular:packages/router/src/utils/config.ts:L31-L43, angular:packages/router/src/models.ts:L760-L769.
5. **Perché Spartacus non usa `loadChildren` per le feature?** Le feature vanno caricate quando il CMS richiede un loro componente (potenzialmente in qualunque route) o quando si usa una facade, non entrando in una route; il caricatore è `FeatureModulesService` → `LazyModulesService`, attivato da

`CmsComponentsService.determineMappings`. — core-libs/storefront/cms-structure/services/cms-components.service.ts:L92-L113, core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L72.

6. **Come ottiene un componente CMS di una feature lazy i servizi del modulo lazy?** `DefaultComponentHandler` passa il `NgModuleRef` come quinto argomento di `createComponent`; Angular costruisce `ChainedInjector(elementInjector, module.injector)`. — core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L62-L68, angular:packages/core/src/render3/component_ref.ts:L140-L159.
7. **Una feature lazy ha due `dependencies`. Chi è il parent del suo injector?** Un `CombinedInjector` con main injector il root e come complementari gli injector dei due moduli di dipendenza (cercati prima con `self`, poi a tutti i livelli, infine il root). — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L89-L93, core-libs/core/src/util/combined-injector.ts:L53-L70.
8. **Perché un `RouterModule.forChild` dentro un modulo caricato da `LazyModulesService` non ha effetto?** Il router legge `ROUTES` dal root all'avvio e dai moduli lazy solo nel flusso `loadChildren`; Spartacus quindi dichiara le route nei moduli `root` eager. — angular:packages/router/src/router.ts:L163-L163, feature-libs/order/root/order-root.module.ts:L82-L90.
9. **Che differenza c'è tra `ENVIRONMENT_INITIALIZER` e `MODULE_INITIALIZER`?** Il primo è eseguito in modo sincrono alla creazione dell'injector e non è atteso; il secondo è letto da Spartacus con `self: true` dopo la creazione del modulo e le Promise restituite sono attese con `Promise.all` prima di emettere `ModuleInitializedEvent`. — angular:packages/core/src/di/r3_injector.ts:L426-L448, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L171.
10. **Come vengono risolti i guard di route CMS forniti solo da un modulo lazy?** Ogni `canActivate` è avvolto in una funzione che usa `getLastValueSync(unifiedInjector.get(guard))`; `UnifiedInjector` interroga il root e, con `self`, gli injector lazy. Se non trova un'istanza tratta il token come `CanActivateFn`. — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L174-L180, core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67.

TAVOLA 03.1 **4.1 Navigazione verso una route loadChildren (NgModule)**

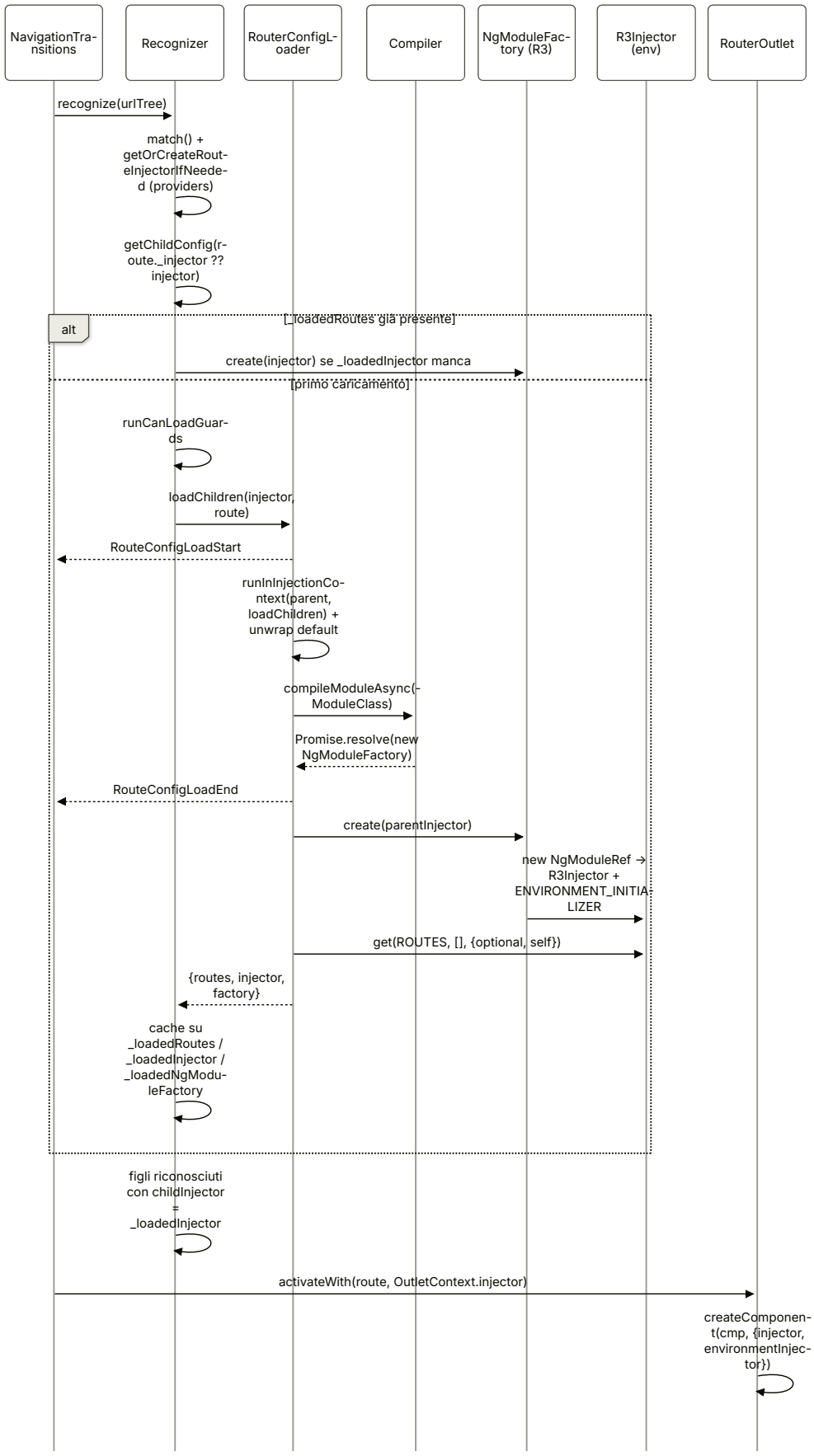


TAVOLA 03.2 **4.2 Gerarchia di injector: router vs Spartacus**

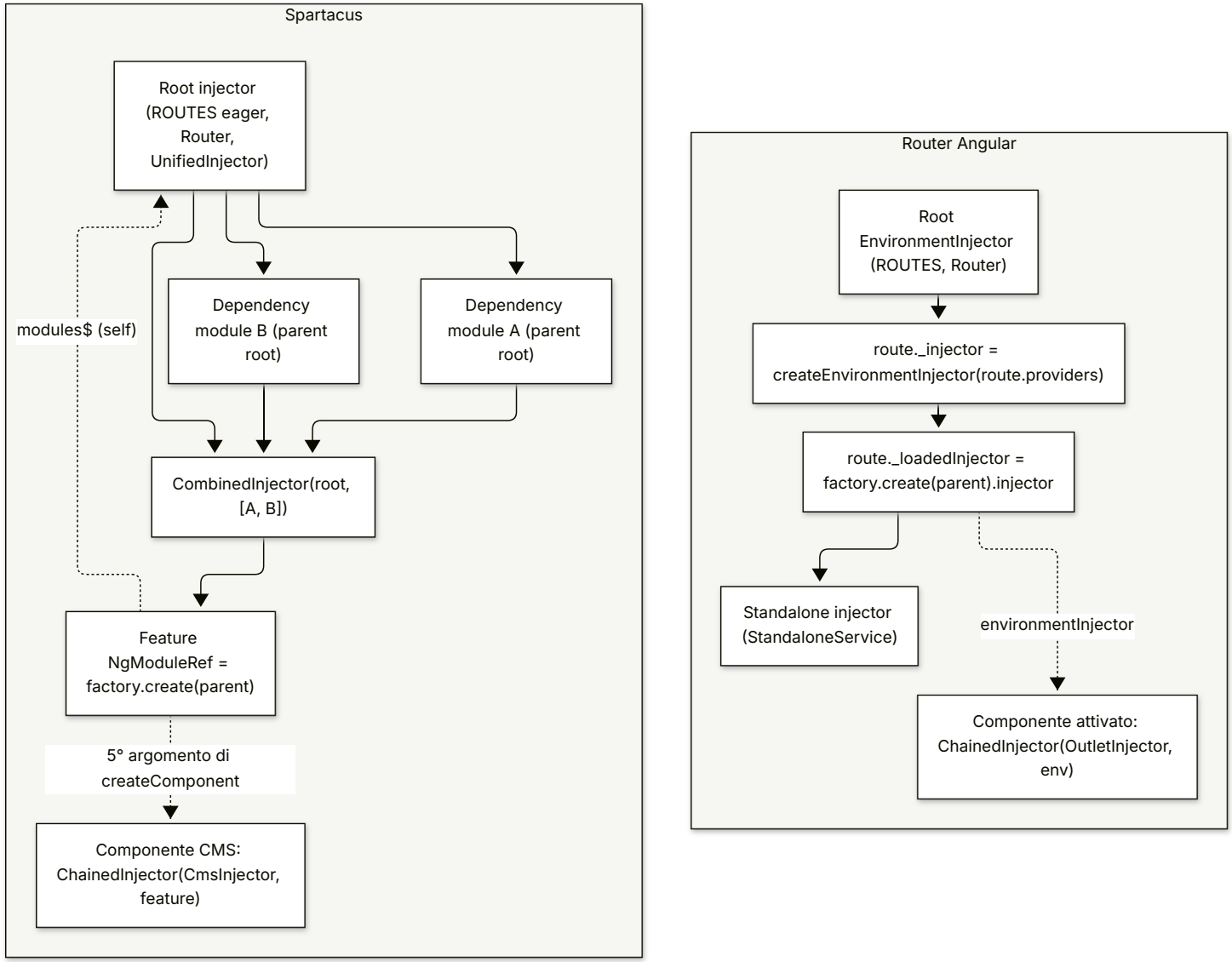
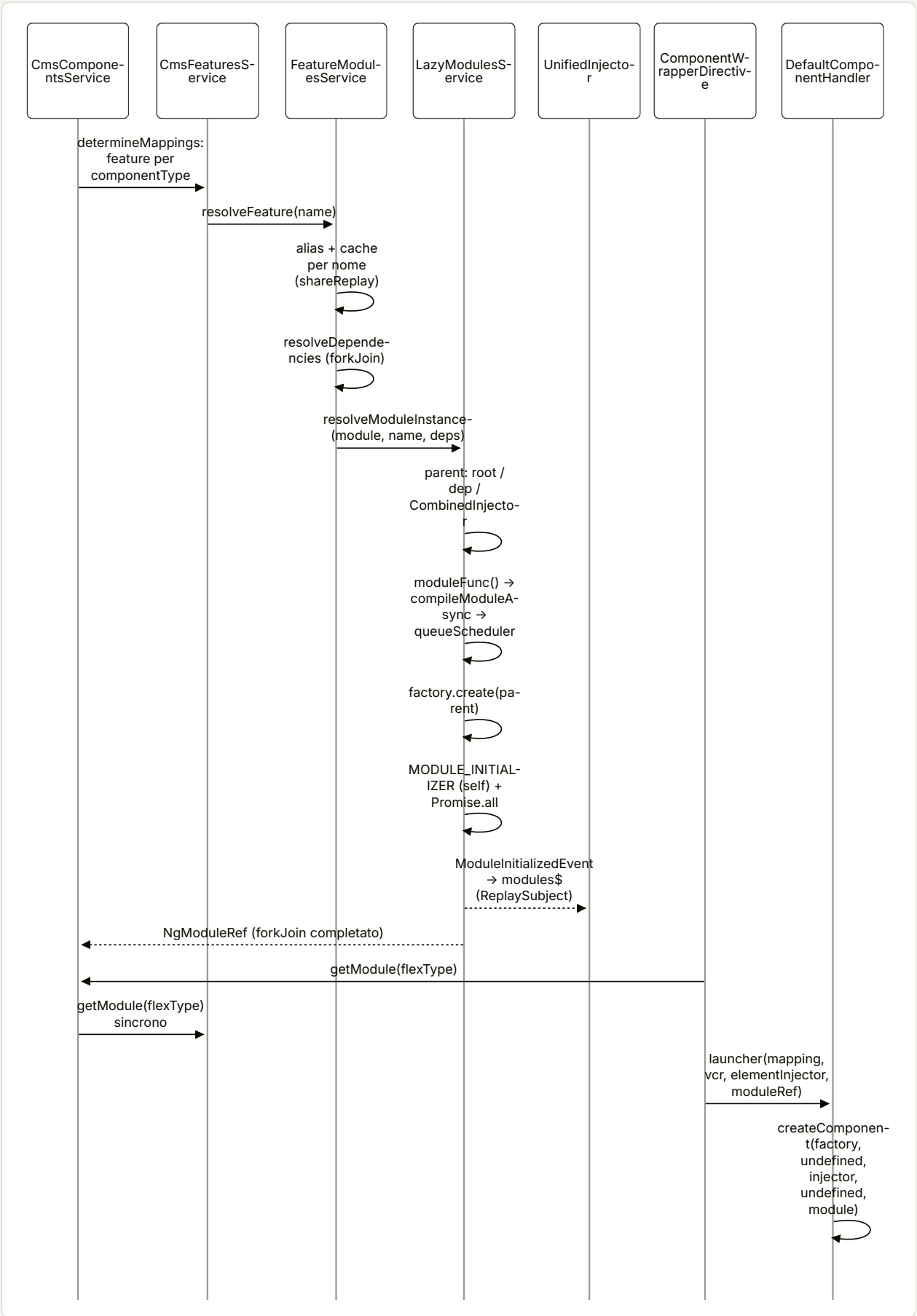


TAVOLA 03.3 **4.3 Componente CMS di una feature lazy (Spartacus)**



04

Change detection, Zone, zoneless e signals

Come Angular riallinea il DOM allo stato, il ruolo di Zone e del modo zoneless, la stabilità per SSR, i signals.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Angular @ `v21.2.23` (prefisso `angular:`), `@angular/ssr` 21.2.23 (prefisso `deps:`).

---

1. A cosa serve e perché esiste

La *change detection* (CD) è il meccanismo con cui Angular riallinea il DOM allo stato dei componenti: percorre l'albero delle view (`LView`), riesegue i template di quelle che ne hanno bisogno e lancia gli hook di ciclo di vita. Il capitolo risponde a quattro domande, tutte con conseguenze dirette per un progetto Spartacus.

1. **Chi decide *quando* parte un giro di CD?** In Angular 21 ci sono due motori:
   - **Zone.js** (`NgZone`): la zona intercetta ogni task asincrono (eventi DOM, timer, XHR, Promise). Quando la coda dei microtask si svuota emette `onMicrotaskEmpty`, e uno scheduler dedicato chiama `ApplicationRef._tick()` — angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L45-L68.
   - **Zoneless**: niente patch del browser. Sono le API Angular (`markForCheck`, `setInput`, listener, signal letti in template, effect…) a *notificare* uno scheduler, che pianifica un tick — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L344-L353, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L138-L224.

   Dalla v21 **il default è zoneless**: `internalCreateApplication` mette `provideZonelessChangeDetectionInternal()` in testa ai provider dell'app — angular:packages/core/src/application/create_application.ts:L64-L71. Lo fa anche il percorso NgModule — angular:packages/core/src/platform/platform_ref.ts:L47-L61. **La demo Spartacus invece è con Zone**: `appConfig` fornisce in modo esplicito `provideZoneChangeDetection({ eventCoalescing: true })` — projects/storefrontapp/src/app/app.config.ts:L26-L33.
2. **Quali view vengono controllate?** Dipende dalla strategia (`OnPush` o `Eager`, l'ex `Default`) e dalla modalità del giro (`Global` o `Targeted`) — angular:packages/core/src/change_detection/constants.ts:L18-L41, angular:packages/core/src/render3/instructions/change_detection.ts:L152-L169. Spartacus dichiara `OnPush` in 400 dei 557 file con `@Component(` delle librerie (F151 del dossier).
3. **Quando l'app è "stabile"?** In v21 `ApplicationRef.isStable` non dipende più da `NgZone`: deriva dal contatore di `PendingTasksInternal` — angular:packages/core/src/application/application_ref.ts:L326-L334. È il punto chiave per l'SSR, perché `renderApplication` serializza l'HTML solo dopo `whenStable()` — angular:packages/platform-server/src/utils.ts:L362-L373. Spartacus usa `PendingTasks` per tenere aperta la stabilità durante l'`import()` dinamico dei componenti CMS lazy — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71.
4. **Che ruolo hanno i signals?** Un signal letto in un template diventa una dipendenza del *consumer reattivo* della view. Quando cambia, Angular rifà solo quella view e si limita ad *attraversare* gli antenati — angular:packages/core/src/render3/reactive_lview_consumer.ts:L43-L53, angular:packages/core/src/render3/util/view_utils.ts:L264-L280. Nelle librerie Spartacus i signals compaiono in pochi componenti recenti (subscription-billing, order, a11y) e in un interceptor (F160–F168).

Il capitolo presuppone 01 — Dependency Injection (injection context, `DestroyRef`) e 02 — NgModule, standalone e bootstrap. Il tema stabilità/SSR prosegue in 07 — SSR, hydration e TransferState e in 39 — SSR ottimizzato. Il caricamento lazy dei componenti CMS è descritto in 16 — CMS rendering.

2. Mappa dei file

Angular (`/tmp/angular`, prefisso `angular:`)

PATH	RUOLO
`angular:packages/core/src/application/create_application.ts`	Bootstrap standalone: antepone i provider zoneless (default v21)
`angular:packages/core/src/platform/platform_ref.ts`	`bootstrapModuleFactory`: zoneless di default anche per NgModule
`angular:packages/core/src/platform/bootstrap.ts`	`bootstrap()`: tutto dentro `ngZone.run`, warning zone+zoneless, pending task del bootstrap
`angular:packages/core/src/zone/ng_zone.ts`	`NgZone`, `NoopNgZone`, zona "angular", `checkStable`, coalescing
`angular:packages/core/src/zone.ts`	Export pubblico: `NgZone` e `NoopNgZone as eNoopNgZone`
`angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts`	`provideZoneChangeDetection`, `NgZoneChangeDetectionScheduler`, `ZoneStablePendingTask`, `NgZoneOptions`
`angular:packages/core/src/change_detection/scheduling/zoneless_scheduling.ts`	`NotificationSource`, `ChangeDetectionScheduler`, token `ZONELESS_ENABLED`, `PROVIDED_ZONELESS`, `SCHEDULE_IN_ROOT_ZONE`
`angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts`	`ChangeDetectionSchedulerImpl`, `provideZonelessChangeDetection(Internal)`
`angular:packages/core/src/change_detection/scheduling/flags.ts`	`SCHEDULE_IN_ROOT_ZONE_DEFAULT`
`angular:packages/core/src/util/callback_scheduler.ts`	Gara `setTimeout`/`requestAnimationFrame`
`angular:packages/core/src/application/application_ref.ts`	`tick`, `synchronize`, `ApplicationRefDirtyFlags`, `isStable`, `whenStable`, `attachView`
`angular:packages/core/src/render3/instructions/change_detection.ts`	`detectChangesInternal`, `detectChangesInView`, `refreshView`, `ChangeDetectionMode`
`angular:packages/core/src/render3/instructions/mark_view_dirty.ts`	`markViewDirty`
`angular:packages/core/src/render3/util/view_utils.ts`	`markAncestorsForTraversal`
`angular:packages/core/src/render3/reactive_lview_consumer.ts`	Consumer reattivo del template
`angular:packages/core/src/render3/view_ref.ts`	`ViewRef.markForCheck`, `detectChanges`
`angular:packages/core/src/render3/view/listeners.ts`	I listener marcano dirty la view
`angular:packages/core/src/render3/component_ref.ts`	`ComponentRef.setInput`
`angular:packages/core/src/render3/view/construction.ts`	Flag iniziali della `LView` (`CheckAlways`, `Dirty`, `SignalView`)
`angular:packages/core/src/change_detection/constants.ts`	`ChangeDetectionStrategy`
`angular:packages/core/src/change_detection/change_detector_ref.ts`	`ChangeDetectorRef` astratta
`angular:packages/core/src/metadata/directives.ts`	Default `changeDetection: Eager` del decoratore `@Component`
`angular:packages/core/src/pending_tasks.ts`	`PendingTasks` pubblico

PATH	RUOLO
angular:packages/core/src/pending_tasks_internal.ts	PendingTasksInternal (Set + BehaviorSubject)
angular:packages/common/src/pipes/async_pipe.ts	AsyncPipe e markForCheck
angular:packages/common/http/src/backend.ts	HttpInterceptorHandler: pending task per ogni richiesta HttpClient
angular:packages/common/http/src/interceptor.ts	Token REQUESTS_CONTRIBUTE_TO_STABILITY e pending task della catena di interceptor legacy (DI)
angular:packages/platform-server/src/utils.ts	renderApplication attende whenStable()
angular:packages/core/primitives/signals/src/signal.ts, computed.ts, graph.ts, equality.ts, untracked.ts, linked_signal.ts	Grafo reattivo push/pull
angular:packages/core/src/render3/reactivity/signal.ts, computed.ts, linked_signal.ts, effect.ts, root_effect_scheduler.ts, view_effect_runner.ts	API pubbliche signal, computed, linkedSignal, effect
angular:packages/core/rxjs-interop/src/to_signal.ts, to_observable.ts, take_until_destroyed.ts	Interop RxJS
angular:packages/core/testing/src/test_bed_compiler.ts	TestBed zoneless di default

Dipendenze (/tmp/deps, prefisso deps:)

PATH	RUOLO
deps:angular-ssr-21.2.23/package/fesm2022/node.mjs	CommonEngine.renderApplication → renderApplication/renderModule

Spartacus

PATH	RUOLO
projects/storefrontapp/src/main.ts	bootstrapApplication(AppComponent, appConfig)
projects/storefrontapp/src/app/app.config.ts	provideZoneChangeDetection({ eventCoalescing: true })
projects/storefrontapp/src/app/app.config.server.ts	mergeApplicationConfig(appConfig, serverConfig): Zone anche in SSR
projects/storefrontapp/src/main.server.ts	Bootstrap server con BootstrapContext
projects/storefrontapp/project.json	Polyfill zone.js (build e test)
core-libs/core/src/test.ts, projects/storefrontapp/src/test.ts	Workaround Karma: provideZoneChangeDetection() nel TestBed
testing/setup-vitest.ts	Setup vitest (zone.js importato, nessun provider zone)
core-libs/schematics/src/migrations/221121_7/move-zone-change-detection-to-main/index.ts	Migrazione provideZoneChangeDetection da main.ts ad app.module.ts
core-libs/schematics/src/modernize-app-migrated-from-2211_32-to-2211_36/csr-and-ssr/update-main-ts.ts	Aggiunge ngZoneEventCoalescing a bootstrapModule()
core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts	Unico uso di PendingTasks nelle librerie

PATH	RUOLO
core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts	`reloadInterval: false` per non bloccare l'SSR
core-libs/setup/ssr/engine/cx-common-engine.ts	`CxCommonEngine extends CommonEngine`
core-libs/storefront/cms-structure/page/slot/page-slot.component.ts	Subscribe manuale + `markForCheck()`
core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts	`markForCheck()` dopo il lancio di un componente CMS
core-libs/storefront/cms-structure/outlet/outlet.directive.ts	`markForCheck()` su ogni embedded view di outlet
core-libs/core/src/i18n/translate.pipe.ts	Pipe impure `cxTranslate` con `markForCheck()`
core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts	`ApplicationRef.attachView` per i dialog
core-libs/storefront/cms-structure/page/component/services/cx-api.service.ts, core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts	`NgZone` esposto ai web component (`cxApi`)
feature-libs/smartedit/core/services/smart-edit.service.ts	`zone.run` per le callback di SmartEdit
integration-libs/cdc/root/service/cdc-js.service.ts, integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts	`zone.run` per le callback Gigya
integration-libs/opf/global-functions/core/services/domains/global/opf-global-functions-global-domain.service.ts, integration-libs/opf/global-functions/core/services/domains/checkout/opf-global-functions-checkout-domain.service.ts, integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts	`ngZone.run` attorno alle global functions OPF
integration-libs/opf/base/root/services/opf-resource-loader.service.ts	Unico `runOutsideAngular`
core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts	`toSignal` in un servizio root
feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts, feature-libs/subscription-billing/components/list/subscription-list.component.ts	`signal/computed/effect/toSignal/toObservable`
feature-libs/subscription-billing/root/components/cart/price-body/subscription-cart-price-body.component.ts, feature-libs/subscription-billing/root/components/cart/price-heading/subscription-cart-price-heading.component.ts, feature-libs/subscription-billing/components/product/price/subscription-product-price.component.ts	`toSignal` su `context$` degli outlet o su facade
core-libs/storefront/shared/components/ng-select-a11y/ng-select-a11y.directive.ts, feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts	`effect` dietro feature toggle a11y

3. Come funziona passo per passo

3.1 Bootstrap: chi fornisce `NgZone` e lo scheduler

1. `bootstrapApplication()` costruisce la configurazione e delega a `internalCreateApplication` — angular:packages/platform-browser/src/browser.ts:L125-L140. La demo lo invoca con

`bootstrapApplication(AppComponent, appConfig)` — projects/storefrontapp/src/main.ts:L17-L17.

2. `internalCreateApplication` compone `allAppProviders = [provideZonelessChangeDetectionInternal(), errorHandlerEnvironmentInitializer, …, ...appProviders]` — angular:packages/core/src/application/create_application.ts:L64-L71. I provider zoneless registrano `ChangeDetectionScheduler → ChangeDetectionSchedulerImpl`, `NgZone → NoopNgZone` e `ZONELESS_ENABLED = true` — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L388-L394.
3. In `appProviders` la demo ha `provideZoneChangeDetection({ eventCoalescing: true })` — projects/storefrontapp/src/app/app.config.ts:L30-L30. Questa funzione crea un vero `NgZone` con le opzioni, imposta `PROVIDED_NG_ZONE=true` e include `internalProvideZoneChangeDetection` — angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L153-L167. Quest'ultimo:
   - sovrascrive `ZONELESS_ENABLED=false` e `NgZone`;
   - aggiunge due `ENVIRONMENT_INITIALIZER`, uno per `NgZoneChangeDetectionScheduler` e uno per `ZoneStablePendingTask`;
   - **non** sovrascrive `ChangeDetectionScheduler`, che resta `ChangeDetectionSchedulerImpl` — angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L84-L131.

   Con Zone quindi lavorano **due** scheduler: quello di zona (guidato da `onMicrotaskEmpty`) e quello "zoneless", che in questa modalità resta per lo più inattivo (§3.3).
4. Le opzioni vengono mappate da `getNgZoneOptions`: `eventCoalescing` → `shouldCoalesceEventChangeDetection`, `runCoalescing` → `shouldCoalesceRunChangeDetection`, entrambi `false` di default — angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L223-L229.
5. Gli environment initializer non vengono eseguiti alla creazione dell'injector, perché devono girare dentro `NgZone` — angular:packages/core/src/application/create_application.ts:L72-L79. `bootstrap()` ottiene `NgZone` dall'injector ed esegue dentro `ngZone.run` gli initializer, gli `APP_INITIALIZER` e il bootstrap del root component — angular:packages/core/src/platform/bootstrap.ts:L84-L93, angular:packages/core/src/platform/bootstrap.ts:L140-L141, angular:packages/core/src/platform/bootstrap.ts:L168-L171.
6. In dev mode, se l'injector contiene sia `PROVIDED_ZONELESS` sia `PROVIDED_NG_ZONE`, compare il warning `PROVIDED_BOTH_ZONE_AND_ZONELESS` — angular:packages/core/src/platform/bootstrap.ts:L95-L105. Gli errori di `NgZone.onError` vengono inoltrati all'error handler interno con una sottoscrizione fatta fuori zona — angular:packages/core/src/platform/bootstrap.ts:L107-L112.
7. Per tutta la durata degli initializer e fino al bootstrap del root component, `bootstrap()` tiene aperto un pending task: prima della fine del bootstrap l'app non può essere stabile — angular:packages/core/src/platform/bootstrap.ts:L137-L178.
8. Il bootstrap del root (`_loadComponent`) esegue `attachView` e poi un `tick()` sincrono, che è il primo render — angular:packages/core/src/application/application_ref.ts:L775-L782.
9. **Lato server** `internalCreateApplication` lancia errore se manca il `BootstrapContext` (`platformRef`) — angular:packages/core/src/application/create_application.ts:L47-L54. Per questo il `main.server.ts` della demo lo inoltra — projects/storefrontapp/src/main.server.ts:L23-L26. La config server è `mergeApplicationConfig(appConfig, serverConfig)` e quindi eredita `provideZoneChangeDetection` — projects/storefrontapp/src/app/app.config.server.ts:L17-L30.

Percorso NgModule. Per le app migrate da versioni con `AppModule` bootstrappato via `bootstrapModule`, `bootstrapModuleFactory` antepone anch'esso `provideZonelessChangeDetectionInternal()` — angular:packages/core/src/platform/platform_ref.ts:L47-L61. Poi `createInjectorWithoutInjectorInstances` mette questi provider aggiuntivi *prima* di quelli del modulo, e `R3Injector` li processa in ordine sovrascrivendo i token non-multi. Di conseguenza un `provideZoneChangeDetection` nei `providers` di `AppModule` prevale — angular:packages/core/src/di/create_injector.ts:L49-L49, angular:packages/core/src/di/r3_injector.ts:L223-L226, angular:packages/

core/src/di/r3_injector.ts:L521-L521. È esattamente quello che fa la migrazione Spartacus 221121_7: sposta provideZoneChangeDetection da main.ts ad app.module.ts — core-libs/schematics/src/migrations/221121_7/move-zone-change-detection-to-main/index.ts:L11-L22, core-libs/schematics/src/migrations/221121_7/move-zone-change-detection-to-main/add-zone-change-detection-to-app-module/index.ts:L55-L80.

3.2 NgZone: come la zona produce un tick

1. Il costruttore di NgZone richiede Zone.js: se Zone non è definito lancia MISSING_ZONEJS, poi chiama Zone.assertZonePatched() — angular:packages/core/src/zone/ng_zone.ts:L151-L158. La demo carica zone.js come polyfill — projects/storefrontapp/project.json:L23-L23.
2. _outer è la zona corrente al momento della creazione. _inner è una fork di _outer (in dev con AsyncStackTaggingZoneSpec('Angular')), forkata a sua volta con il comportamento Angular — angular:packages/core/src/zone/ng_zone.ts:L162-L187. La zona "angular" marca le proprietà isAngularZone/isAngularZone_ID<n> e intercetta onInvokeTask, onInvoke, onHasTask e onHandleError — angular:packages/core/src/zone/ng_zone.ts:L426-L523.
3. All'ingresso di un task, onEnter incrementa _nesting ed emette onUnstable se la zona era stabile. All'uscita onLeave decrementa e chiama checkStable — angular:packages/core/src/zone/ng_zone.ts:L538-L549.
4. checkStable: con _nesting == 0 e nessun microtask pendente emette onMicrotaskEmpty. Se dopo l'emissione i microtask sono ancora zero, emette onStable fuori zona e imposta isStable = true — angular:packages/core/src/zone/ng_zone.ts:L354-L385. onMicrotaskEmpty può scattare più volte nello stesso VM turn; onStable una volta sola — angular:packages/core/src/zone/ng_zone.ts:L115-L137.
5. **Coalescing** (la demo usa eventCoalescing: true). Alla fine di un event task, o di ogni task se è attivo runCoalescing, la zona chiama delayChangeDetectionForEvents, che rinvia checkStable con scheduleCallbackWithRafRace eseguito fuori zona — angular:packages/core/src/zone/ng_zone.ts:L451-L462, angular:packages/core/src/zone/ng_zone.ts:L401-L423. Nel frattempo la zona dichiara hasPendingMicrotasks = true, e così ritarda onMicrotaskEmpty — angular:packages/core/src/zone/ng_zone.ts:L526-L536. scheduleCallbackWithRafRace mette in gara setTimeout e requestAnimationFrame: vince il primo, che cancella l'altro — angular:packages/core/src/util/callback_scheduler.ts:L37-L66. Con runCoalescing attivo, eventCoalescing viene ignorato — angular:packages/core/src/zone/ng_zone.ts:L180-L184.
6. NgZoneChangeDetectionScheduler.initialize() sottoscrive onMicrotaskEmpty. Se lo scheduler zoneless non sta già tickando, fa zone.run(() => { dirtyFlags |= ViewTreeGlobal; applicationRef._tick(); }) — angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L45-L68. Il flag ViewTreeGlobal è la ragione per cui con Zone le view CheckAlways vengono controllate a ogni giro (§3.4).
7. ZoneStablePendingTask traduce la stabilità della zona in pending task: su onUnstable chiama pendingTasks.add(); su onStable rimuove il task in un microtask, se intanto non sono comparsi macro o microtask — angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L231-L281.

API di NgZone usate dal codice applicativo:

- run(fn) esegue fn in _inner; gli errori sincroni vengono rilanciati e **non** riportati via onError — angular:packages/core/src/zone/ng_zone.ts:L232-L236.
- runGuarded(fn) è come run, ma cattura gli errori sincroni e li inoltra a onError — angular:packages/core/src/zone/ng_zone.ts:L260-L266.
- runOutsideAngular(fn) esegue in _outer: i task schedulati da lì non scatenano CD — angular:packages/core/src/zone/ng_zone.ts:L268-L283.

- `NgZone.isInAngularZone()` legge `isAngularZone` dalla zona corrente e funziona anche senza Zone — angular:packages/core/src/zone/ng_zone.ts:L193-L196.
- I task con `data.__ignore_ng_zone__ === true` girano senza `onEnter`/`onLeave` — angular:packages/core/src/zone/ng_zone.ts:L446-L449, angular:packages/core/src/zone/ng_zone.ts:L581-L583.
- `NoopNgZone` (il default zoneless) implementa le stesse API senza zone: `run` e `runOutsideAngular` chiamano direttamente `fn`, `isStable` vale sempre `true` e gli eventi non vengono mai emessi — angular:packages/core/src/zone/ng_zone.ts:L551-L579. È esportato solo come simbolo privato `ɵNoopNgZone` — angular:packages/core/src/zone.ts:L9-L10.

3.3 Lo scheduler unificato: `ChangeDetectionSchedulerImpl`

`ChangeDetectionScheduler` è la classe astratta che viene notificata quando una `LView` diventa sporca: espone `notify(source)` e `runningTick` — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling.ts:L58-L64. Le possibili sorgenti sono enumerate in `NotificationSource` (`MarkAncestorsForTraversal`, `SetInput`, `MarkForCheck`, `Listener`, `ViewAttached`, `PendingTaskRemoved`, `RootEffect`, `ViewEffect`, …) — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling.ts:L11-L56.

1. **Filtro Zone.** Se zoneless non è attivo, `notify(Listener)` viene ignorata, perché i listener girano già in zona — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L138-L149.
2. **Traduzione in dirty flag.** `MarkForCheck`, `Listener`, `SetInput`, `DeferBlockStateUpdate` e `DebugApplyChanges` producono `ViewTreeCheck`. `MarkAncestorsForTraversal`, `CustomElement` e `ViewEffect` producono `ViewTreeTraversal`. `RootEffect` produce `RootEffects`. `PendingTaskRemoved` non produce nessun flag. Tutte le altre sorgenti producono `AfterRender` — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L151-L194.
3. **Serve un tick?** `shouldScheduleTick` risponde no in tre casi — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L227-L246:
   - l'app è distrutta;
   - un tick è già pendente o in corso;
   - con Zone, la chiamata avviene dentro la zona Angular: la zona stabilizzerà e farà la CD da sola.
4. **Pianificazione.** Lo scheduler registra un pending task (`pendingRenderTaskId`) e pianifica `tick()` fuori dalla zona Angular con `scheduleCallbackWithRafRace`, oppure con un microtask subito dopo un tick — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L214-L224. Dopo ogni tick, per un microtask resta attivo lo scheduler a microtask: così le notifiche ravvicinate (il pattern `Promise.resolve` di NgModel) finiscono nello stesso paint — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L122-L136. Con Zone presente i tick manuali non ricevono questo trattamento — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L92-L104.
5. **Cancellazione da parte della zona.** Se la zona diventa instabile mentre lo scheduler non sta tickando, la CD pianificata viene cancellata: ci penserà la zona — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L110-L119.
6. `tick()`. Se `dirtyFlags === None` (per esempio quando l'unica notifica era la rimozione di un pending task) fa solo cleanup. Con Zone, se c'è un qualunque flag di view, aggiunge `ViewTreeGlobal` (workaround marcato TODO) — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L265-L287. Poi esegue `appRef._tick()` dentro `ngZone.run` con gli argomenti `__scheduler_tick__`, che la zona riconosce per non ritardare la stabilità — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L289-L298, angular:packages/core/src/zone/ng_zone.ts:L478-L493.

7. **Protezione dai loop.** In dev mode, 100 notifiche consecutive con lo scheduler a microtask lanciano `INFINITE_CHANGE_DETECTION` — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L36-L57.

3.4 `ApplicationRef.tick()` e il traversal delle view

1. `ApplicationRef.tick()` pubblico aggiunge `ViewTreeGlobal` solo se zoneless **non** è attivo, poi chiama `_tick()` — angular:packages/core/src/application/application_ref.ts:L570-L575.
2. `tickImpl` lancia `RECURSIVE_APPLICATION_REF_TICK` se viene chiamato in modo ricorsivo. Azzera il consumer reattivo attivo, esegue `synchronize()`, in dev mode lancia `checkNoChanges()` su tutte le view e infine emette `afterTick` — angular:packages/core/src/application/application_ref.ts:L591-L619.
3. `synchronize()` ripete `synchronizeOnce()` finché restano dirty flag, al massimo `MAXIMUM_REFRESH_RERUNS = 10` volte; poi, in dev mode, lancia `INFINITE_CHANGE_DETECTION` — angular:packages/core/src/application/application_ref.ts:L159-L159, angular:packages/core/src/application/application_ref.ts:L625-L649.
4. `synchronizeOnce()` lavora in tre fasi — angular:packages/core/src/application/application_ref.ts:L654-L725:
   1. flush dei root effect;
   2. se ci sono view sporche, CD su ogni root view in modalità `Global` (se è presente `ViewTreeGlobal` **e** non si è in zoneless) oppure `Targeted`;
   3. esecuzione degli hook `afterRender`.

   In zoneless anche le root view `CheckAlways` devono essere marcate in modo esplicito — angular:packages/core/src/application/application_ref.ts:L682-L689. Un `markForCheck()` chiamato *durante* il check non provoca un nuovo giro — angular:packages/core/src/application/application_ref.ts:L694-L697.
5. La modalità determina quali view vengono rifatte: in `Global` le view `Dirty`, `CheckAlways` e `RefreshView`; in `Targeted` solo quelle con `RefreshView` o con signal cambiati — angular:packages/core/src/render3/instructions/change_detection.ts:L152-L169.
6. `detectChangesInView` rifà la view (`refreshView`) se vale almeno una di queste condizioni: `Global && CheckAlways`, `Dirty && Global`, `RefreshView`, oppure consumer reattivo del template sporco con producer effettivamente cambiati. Altrimenti, se la view ha `HasChildViewsToRefresh`, ne attraversa i figli in `Targeted` senza rieseguirne il template — angular:packages/core/src/render3/instructions/change_detection.ts:L458-L517. Le view staccate con `detach()` non vengono visitate — angular:packages/core/src/render3/instructions/change_detection.ts:L436-L446.
7. `refreshView` esegue in ordine — angular:packages/core/src/render3/instructions/change_detection.ts:L227-L354:
   1. template (update);
   2. pre-order hook (`OnInit`/`DoCheck`);
   3. transplanted view;
   4. effetti di view (`runEffectsInView`);
   5. embedded view;
   6. content hook;
   7. host binding;
   8. componenti figli;
   9. view query e view hook.

   Al termine cancella il flag `Dirty`.

Strategie. `ChangeDetectionStrategy` vale `OnPush = 0`, `Eager = 1` e `Default = 1`; `Default` è `@deprecated` a favore di `Eager` — angular:packages/core/src/change_detection/constants.ts:L18-L41. Il decoratore `@Component` usa `Eager` come default — angular:packages/core/src/metadata/directives.ts:L669-L673. La `LView` di un componente nasce con uno di questi flag — angular:packages/core/src/render3/view/construction.ts:L263-L271:

- `CheckAlways` (componente `Eager`);
- `Dirty` (componente `OnPush`: il primo check è garantito);
- `SignalView` (se `def.signals`).

Un input di binding cambiato marca `Dirty` la view figlia non-`CheckAlways` (`markDirtyIfOnPush`) — angular:packages/core/src/render3/instructions/shared.ts:L318-L325.

Chi sporca una view.

- `ViewRef.markForCheck()` chiama `markViewDirty(..., MarkForCheck)` — angular:packages/core/src/render3/view_ref.ts:L163-L165. `markViewDirty` notifica lo scheduler e risale fino alla root, marcando ogni view `Dirty` (più `RefreshView` se non c'è un refresh in corso) — angular:packages/core/src/render3/instructions/mark_view_dirty.ts:L26-L52.
- Ogni listener di template o host marca dirty la view prima di eseguire l'handler — angular:packages/core/src/render3/view/listeners.ts:L62-L68.
- `ComponentRef.setInput` ignora i valori identici (`Object.is`) e per gli altri marca dirty con `SetInput` — angular:packages/core/src/render3/component_ref.ts:L566-L580.
- `ViewRef.detectChanges()` imposta `RefreshView` ed esegue subito `detectChangesInternal`, in modo sincrono — angular:packages/core/src/render3/view_ref.ts:L308-L316.
- `AsyncPipe` è impure e chiama `markForCheck()` solo per i valori che arrivano in modo asincrono — angular:packages/common/src/pipes/async_pipe.ts:L140-L143, angular:packages/common/src/pipes/async_pipe.ts:L178-L191, angular:packages/common/src/pipes/async_pipe.ts:L235-L242. Sottoscrive dentro `untracked()` — angular:packages/common/src/pipes/async_pipe.ts:L40-L52 — e, se cambia il riferimento all'Observable, chiude la vecchia subscription e ne apre una nuova — angular:packages/common/src/pipes/async_pipe.ts:L194-L197.

3.5 Stabilità e `PendingTasks`

1. `PendingTasks` (`@publicApi 20.0`) tiene traccia dei task che bloccano la stabilità; tra gli usi previsti c'è l'SSR, che deve attendere la stabilità prima di serializzare — angular:packages/core/src/pending_tasks.ts:L18-L42.
2. `add()` restituisce una funzione di cleanup idempotente. Quando viene chiamata, notifica lo scheduler con `PendingTaskRemoved`, e questo tiene aperta la stabilità fino al tick successivo — angular:packages/core/src/pending_tasks.ts:L50-L61. `run(fn)` (`@developerPreview 19.0`) blocca la stabilità finché la Promise restituita da `fn` non termina — angular:packages/core/src/pending_tasks.ts:L63-L79.
3. `PendingTasksInternal` mantiene un `Set<number>` e un `BehaviorSubject<boolean>`: emette `true` all'arrivo del primo task e `false` quando il set si svuota — angular:packages/core/src/pending_tasks_internal.ts:L20-L69. Dopo la distruzione, `hasPendingTasksObservable` restituisce un Observable che emette `false` e completa — angular:packages/core/src/pending_tasks_internal.ts:L32-L46.
4. `ApplicationRef.isStable` è `hasPendingTasksObservable.pipe(map(p => !p))` — angular:packages/core/src/application/application_ref.ts:L326-L334. `whenStable()` risolve alla prima emissione `true` — angular:packages/core/src/application/application_ref.ts:L344-L357.
5. Chi contribuisce alla stabilità:
   - il pending task del bootstrap — angular:packages/core/src/platform/bootstrap.ts:L137-L178;

- il pending task di zona (`ZoneStablePendingTask`, solo con Zone);
- il `pendingRenderTaskId` dello scheduler — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L217-L217;
- ogni richiesta `HttpClient`: se `REQUESTS_CONTRIBUTE_TO_STABILITY` vale `true` (default, angular:packages/common/http/src/interceptor.ts:L217-L221), `HttpInterceptorHandler.handle` registra un pending task e lo rimuove con `finalize` — angular:packages/common/http/src/backend.ts:L125-L129. Con `withInterceptorsFromDi` (usato dalla demo, projects/storefrontapp/src/app/app.config.ts:L28-L28) la catena degli interceptor legacy ne aggiunge un secondo — angular:packages/common/http/src/interceptor.ts:L243-L248;
- i task aggiunti a mano con `PendingTasks`.

6. In SSR `renderApplication` esegue `await applicationRef.whenStable()` prima di `renderInternal` — angular:packages/platform-server/src/utils.ts:L362-L373. `CommonEngine` di `@angular/ssr` usa `renderApplication` per le bootstrap function — deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L185-L210. Spartacus estende `CommonEngine` con `CxCommonEngine` — core-libs/setup/ssr/engine/cx-common-engine.ts:L20-L66.
7. Un task ricorrente avviato all'avvio (`setInterval`, `interval`) impedisce per sempre la stabilità — angular:packages/core/src/application/application_ref.ts:L172-L194. Spartacus lo evita nell'inizializzatore i18next, dove forza `reloadInterval: false` — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L62-L69.

3.6 Signals: grafo push/pull

1. `createSignal` crea un `SignalNode` più getter, setter e updater — angular:packages/core/primitives/signals/src/signal.ts:L53-L73. Leggere il signal chiama `producerAccessed(node)` — angular:packages/core/primitives/signals/src/signal.ts:L81-L84. Fuori da un contesto reattivo (`activeConsumer === null`) non viene registrato nulla; leggere un signal durante la fase di notifica lancia un errore — angular:packages/core/primitives/signals/src/graph.ts:L204-L216.
2. `signalSetFn` rifiuta la scrittura se il consumer attivo non la consente, e aggiorna il valore solo se `!equal(old, new)` — angular:packages/core/primitives/signals/src/signal.ts:L86-L95. L'uguaglianza di default è `Object.is` — angular:packages/core/primitives/signals/src/equality.ts:L17-L18. Quando il valore cambia: `version++`, incremento dell'`epoch` globale, notifica ai consumer — angular:packages/core/primitives/signals/src/signal.ts:L120-L125.
3. **Push.** `producerNotifyConsumers` marca dirty i consumer live che non lo sono già — angular:packages/core/primitives/signals/src/graph.ts:L327-L349. `consumerMarkDirty` propaga la marcatura e invoca l'hook `consumerMarkedDirty` del nodo: è lì che view ed effect si agganciano allo scheduler — angular:packages/core/primitives/signals/src/graph.ts:L359-L363.
4. **Pull.** `producerUpdateValueVersion` salta il ricalcolo se il consumer è live e pulito, o se è stato pulito nello stesso epoch. Ricalcola solo se un producer ha cambiato versione — angular:packages/core/primitives/signals/src/graph.ts:L297-L322, angular:packages/core/primitives/signals/src/graph.ts:L448-L472.
5. `computed` è lazy. Memorizza l'errore e lo rilancia a ogni lettura — angular:packages/core/primitives/signals/src/computed.ts:L70-L82. Rileva i cicli e, se il nuovo valore è `equal` al precedente, non incrementa la versione — angular:packages/core/primitives/signals/src/computed.ts:L132-L171. `untracked(fn)` esegue `fn` con `activeConsumer = null` — angular:packages/core/primitives/signals/src/untracked.ts:L15-L24.
6. **API pubbliche.**
   - `signal()` aggiunge `set`, `update` e `asReadonly` al getter — angular:packages/core/src/render3/reactivity/signal.ts:L72-L89. `asReadonly` non impedisce la mutazione profonda del valore —

angular:packages/core/src/render3/reactivity/signal.ts:L35-L40.

- `computed()` avvolge `createComputed` — angular:packages/core/src/render3/reactivity/computed.ts:L32-L42.
- `linkedSignal` (`@publicApi 20.0`) è un signal scrivibile, reinizializzato da una computazione che riceve `previous = {source, value}` — angular:packages/core/src/render3/reactivity/linked_signal.ts:L23-L47, angular:packages/core/primitives/signals/src/linked_signal.ts:L159-L169. Il suo `set()` vale finché la sorgente non cambia — angular:packages/core/primitives/signals/src/linked_signal.ts:L106-L110.

7. **Template.** Il template di una view gira dentro un consumer reattivo (`getOrBorrowReactiveLViewConsumer`) — angular:packages/core/src/render3/instructions/change_detection.ts:L204-L225. Quando un signal letto cambia, il consumer chiama `markAncestorsForTraversal`, non `markViewDirty` — angular:packages/core/src/render3/reactive_lview_consumer.ts:L43-L53. `markAncestorsForTraversal` notifica lo scheduler e imposta `HasChildViewsToRefresh` sugli antenati — angular:packages/core/src/render3/util/view_utils.ts:L264-L280. Al tick, gli antenati vengono attraversati e viene rifatta solo la view con il consumer sporco (§3.4, punto 6).
8. **Effect.** `effect()` (`@publicApi 20.0`), in dev mode, asserisce di non essere chiamato in un contesto reattivo e di trovarsi in un injection context (salvo `options.injector`) — angular:packages/core/src/render3/reactivity/effect.ts:L141-L150. In produzione queste asserzioni non girano: resta solo il requisito implicito di `inject(Injector)`, che senza injection context lancia comunque — angular:packages/core/src/render3/reactivity/effect.ts:L158-L158. Se nell'injector trova un `ViewContext` crea un *view effect*, altrimenti un *root effect* — angular:packages/core/src/render3/reactivity/effect.ts:L158-L182. I due tipi si comportano così:
   - **root effect**: quando diventa sporco si accoda in `EffectScheduler` e notifica `RootEffect`; viene eseguito nel flush iniziale di `synchronizeOnce` — angular:packages/core/src/render3/reactivity/effect.ts:L254-L260, angular:packages/core/src/application/application_ref.ts:L656-L659;
   - **view effect**: marca `HasChildViewsToRefresh` e notifica `ViewEffect`; viene eseguito da `runEffectsInView` durante il refresh o il traversal — angular:packages/core/src/render3/reactivity/effect.ts:L275-L282, angular:packages/core/src/render3/reactivity/view_effect_runner.ts:L11-L43.

   Entrambi vengono schedulati subito alla creazione — angular:packages/core/src/render3/reactivity/effect.ts:L308-L312, angular:packages/core/src/render3/reactivity/effect.ts:L325-L327. Ogni effect memorizza la `Zone.current` in cui è stato creato e gira in quella zona — angular:packages/core/src/render3/reactivity/effect.ts:L304-L304, angular:packages/core/src/render3/reactivity/view_effect_runner.ts:L28-L34.

3.7 Interop RxJS

- `toSignal()` richiede un injection context salvo `injector` o `manualCleanup`, e non va chiamato in un contesto reattivo — angular:packages/core/rxjs-interop/src/to_signal.ts:L132-L148. Sottoscrive subito (eager); senza `initialValue` il valore iniziale è `undefined` — angular:packages/core/rxjs-interop/src/to_signal.ts:L161-L188. Con `requireSync: true`, se l'Observable non emette in modo sincrono lancia `REQUIRE_SYNC_WITHOUT_SYNC_EMIT` — angular:packages/core/rxjs-interop/src/to_signal.ts:L190-L196. Il signal restituito è un `computed` che rilancia l'errore dell'Observable; la subscription viene chiusa su `DestroyRef.onDestroy` — angular:packages/core/rxjs-interop/src/to_signal.ts:L198-L225.
- `toObservable()` usa internamente un `effect` che spinge i valori in un `ReplaySubject(1)`: le emissioni seguono i tempi degli effect, non sono sincrone al `set` — angular:packages/core/rxjs-interop/src/to_observable.ts:L48-L75.

- `takeUntilDestroyed(destroyRef?)` usa `inject(DestroyRef)` se il parametro manca; se il `DestroyRef` è già distrutto, completa subito — angular:packages/core/rxjs-interop/src/take_until_destroyed.ts:L25-L43.

3.8 Come lo usa Spartacus

Zone. Tutta la catena demo + test Karma assume Zone.js:

- polyfill `zone.js` nel build e nei test — projects/storefrontapp/project.json:L23-L23, projects/storefrontapp/project.json:L234-L234;
- dipendenza `zone.js ^0.16.0` nel monorepo — package.json:L197-L197;
- `provideZoneChangeDetection({ eventCoalescing: true })` nella config — projects/storefrontapp/src/app/app.config.ts:L30-L30.

Nessun file di libreria o della demo usa `provideZonelessChangeDetection` (F140). Poiché il TestBed di Angular v21 è zoneless di default — angular:packages/core/testing/src/test_bed_compiler.ts:L932-L947 — i `test.ts` Karma aggiungono `provideZoneChangeDetection()` per evitare gli errori NG0100 — core-libs/core/src/test.ts:L21-L35, projects/storefrontapp/src/test.ts:L30-L35. Il setup vitest invece importa `zone.js` ma non fornisce il provider — testing/setup-vitest.ts:L7-L19.

NgZone. Gli usi sono pochi e servono tutti a rientrare in zona da callback esterne:

- SmartEdit — feature-libs/smartedit/core/services/smart-edit.service.ts:L126-L145;
- Gigya/CDC — integration-libs/cdc/root/service/cdc-js.service.ts:L165-L172, integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L97-L101;
- global functions OPF chiamate da script di terze parti — integration-libs/opf/global-functions/core/services/domains/global/opf-global-functions-global-domain.service.ts:L73-L85, integration-libs/opf/global-functions/core/services/domains/checkout/opf-global-functions-checkout-domain.service.ts:L49-L49 (ripetuto a L58, L70, L85, L95), integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L111-L111 (ripetuto a L177, L227, L256).

C'è un solo `runOutsideAngular`, per eseguire script esterni — integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L232-L250. `CxApiService` espone `ngZone` ai web component tramite `webElement.cxApi` — core-libs/storefront/cms-structure/page/component/services/cx-api.service.ts:L73-L75, core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L75-L78.

OnPush e notifiche manuali. Il motore CMS combina `OnPush` con chiamate esplicite:

- `PageSlotComponent` aggiorna `components` in una subscribe e chiama `cd.markForCheck()` — core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L124-L131;
- `ComponentWrapperDirective` marca il proprio `ChangeDetectorRef` dopo il lancio del componente — core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L105-L113;
- `OutletDirective` marca ogni embedded view — core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L175-L185;
- `cxTranslate` è una pipe impure che, come `AsyncPipe`, chiama `markForCheck()` — core-libs/core/src/i18n/translate.pipe.ts:L25-L29, core-libs/core/src/i18n/translate.pipe.ts:L80-L89.

I dialog inline vengono agganciati ad `ApplicationRef.attachView` — core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L32-L55 — e quindi controllati a ogni tick (`attachView` li aggiunge a `_views`, angular:packages/core/src/application/application_ref.ts:L753-L773).

PendingTasks. È usato in un solo punto: `LazyComponentHandler` registra un task intorno all'`import()` del chunk del componente CMS e lo rimuove con `finalize` — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L30-L31, core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71. Riconosce i mapping lazy guardando la stringa della funzione (`function()` o `()=>`) e ha priorità `LOW` — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L38-L52.

Signals. Gli usi sono pochi e recenti:

- `toSignal` in `HttpErrorHandlerInterceptor` — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L40-L46;
- `signal` alimentato da `loading$` in `ConsentManagementComponent` — core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L119-L124;
- `effect` in `NgSelectA11yDirective` (toggle `a11yVocalizeDropdownItemCount`, default `true`) — core-libs/storefront/shared/components/ng-select-a11y/ng-select-a11y.directive.ts:L98-L106, core-libs/storefront/shared/components/ng-select-a11y/ng-select-a11y.directive.ts:L189-L190, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1063-L1063;
- `viewChild()` + `effect` in `StoreFinderListComponent` (toggle `a11yStoreFinderFocusOnBackButton`, default `false`) — feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L68-L83, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1020-L1020;
- `viewChild.required` in `SortingComponent` — core-libs/storefront/shared/components/list-navigation/sorting/sorting.component.ts:L38-L38;
- `toSignal` + `computed` + `effect` in `SubscriptionActionsModalComponent` — feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L67-L96, feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L120-L121;
- altri `toSignal` nella subscription-billing — feature-libs/subscription-billing/root/components/cart/price-body/subscription-cart-price-body.component.ts:L24-L24, feature-libs/subscription-billing/root/components/cart/price-heading/subscription-cart-price-heading.component.ts:L25-L25, feature-libs/subscription-billing/components/product/price/subscription-product-price.component.ts:L33-L35;
- `toObservable` → `switchMap` → `toSignal` in `SubscriptionListComponent` — feature-libs/subscription-billing/components/list/subscription-list.component.ts:L44-L71;
- `toSignal` con `initialValue` in `OrderAttachmentsDialogComponent` — feature-libs/order/components/order-details/order-attachments/attachments-dialog/order-attachments-dialog.component.ts:L86-L90;
- un unico `afterNextRender`, in `ConfiguratorOverviewFormComponent` — feature-libs/product-configurator/rulebased/components/overview-form/configurator-overview-form.component.ts:L434-L443.

Nelle librerie non compaiono `linkedSignal`, `untracked`, `input()`/`output()`/`model()` signal-based né `selectSignal` (F169).

---

4. Diagrammi

4.1 Scelta del motore al bootstrap (demo Spartacus)

■ **TAVOLA 04.1** diagramma a tutta pagina, in fondo al capitolo

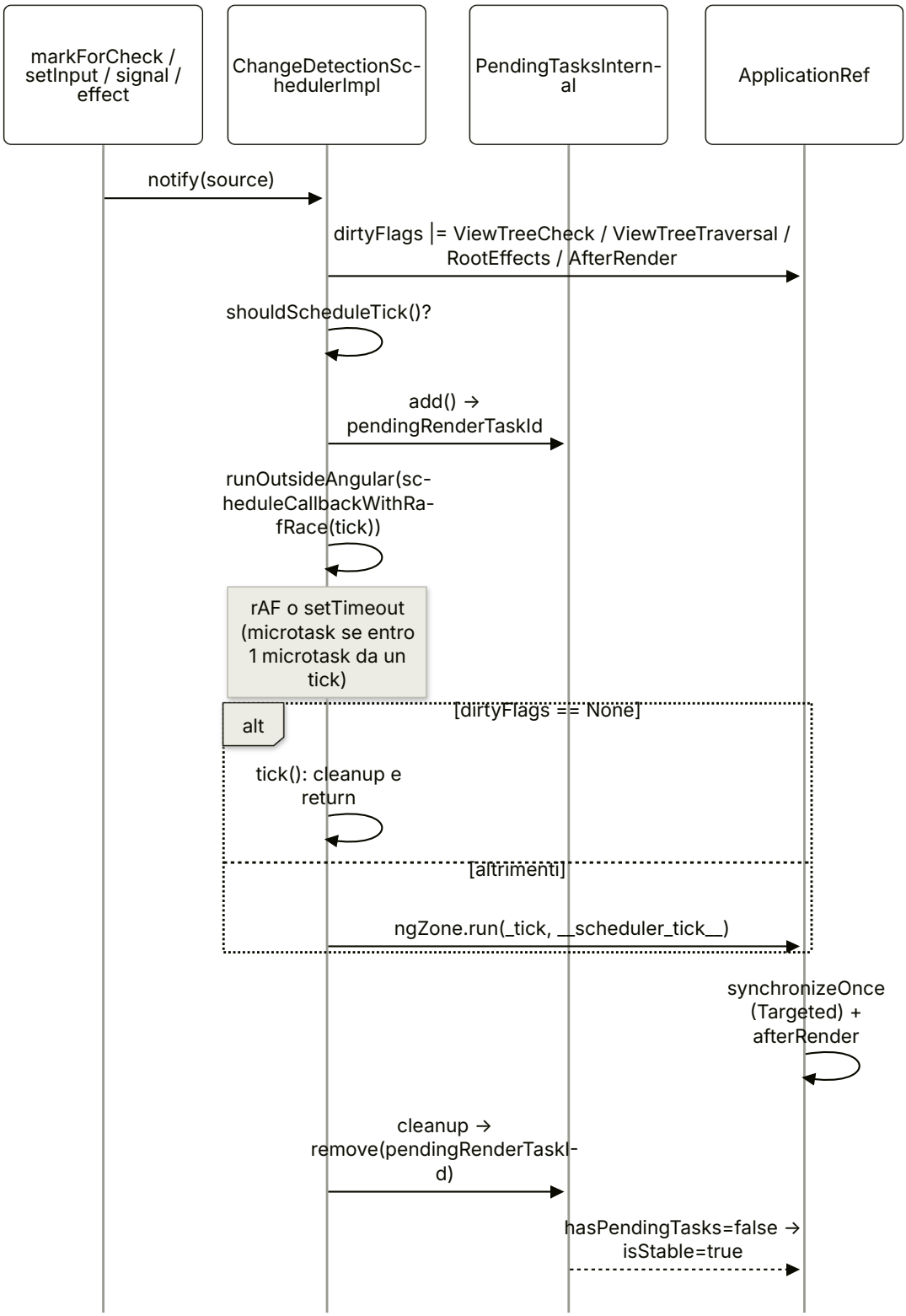
Riferimenti: projects/storefrontapp/src/main.ts:L17-L17, angular:packages/core/src/application/create_application.ts:L64-L71, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L388-L394, angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L84-L131, angular:packages/core/src/platform/bootstrap.ts:L84-L93, angular:packages/core/src/platform/bootstrap.ts:L137-L178, angular:packages/core/src/application/application_ref.ts:L775-L782. Il nodo H viene eseguito solo nel ramo Zone, perché i due initializer sono registrati da `internalProvideZoneChangeDetection`.

4.2 Ciclo con Zone.js ed event coalescing

■ **TAVOLA 04.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: angular:packages/core/src/zone/ng_zone.ts:L538-L549, angular:packages/core/src/render3/view/listeners.ts:L62-L68, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L138-L149, angular:packages/core/src/zone/ng_zone.ts:L451-L462, angular:packages/core/src/zone/ng_zone.ts:L354-L385, angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L45-L68, angular:packages/core/src/application/application_ref.ts:L591-L619, angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L231-L281.

4.3 Ciclo zoneless: dalla notifica al tick



Riferimenti: angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L151-L194, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L214-L246, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L122-L136, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L265-L298,

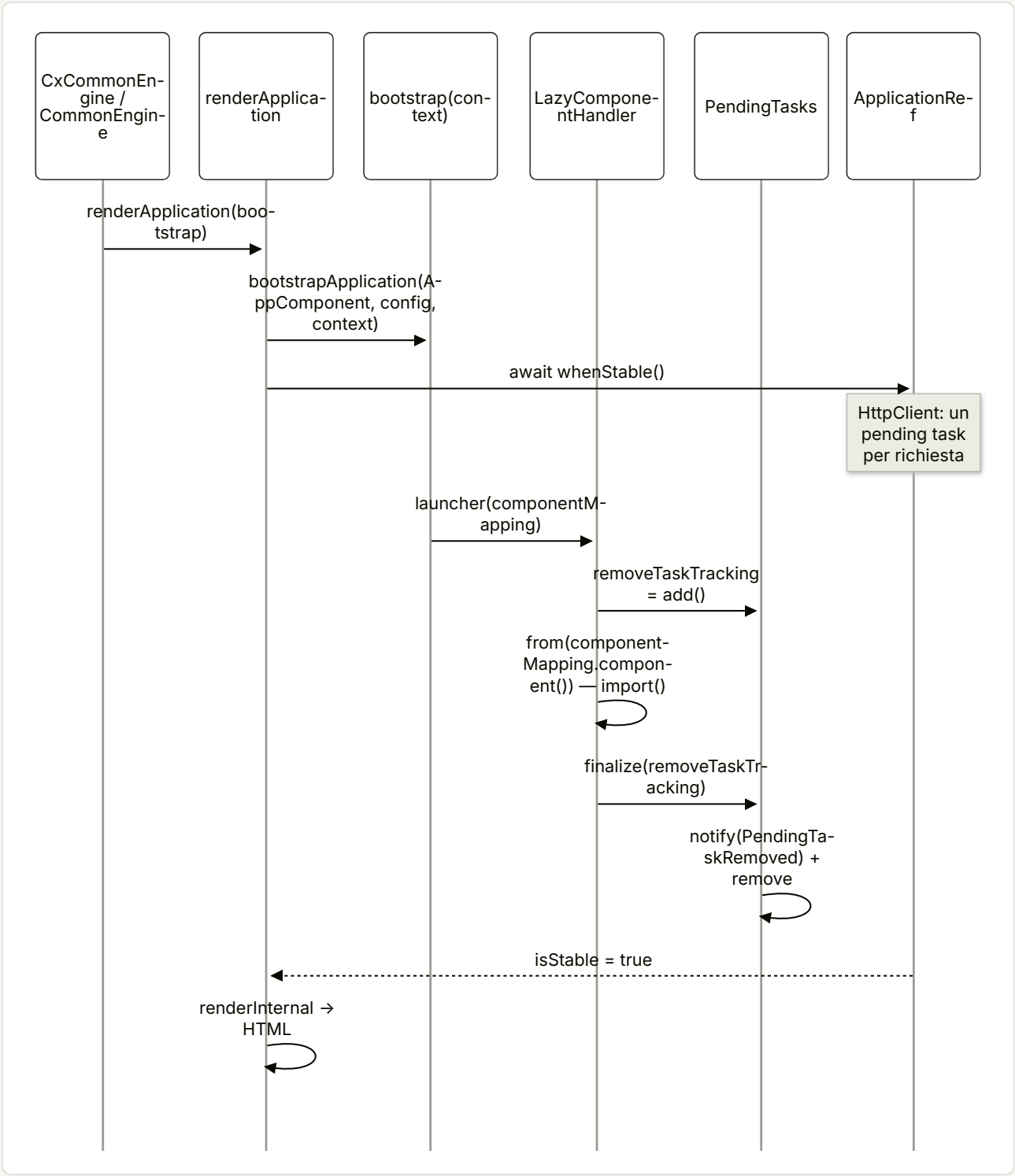
angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L312-L328, angular:packages/core/src/application/application_ref.ts:L682-L690, angular:packages/core/src/application/application_ref.ts:L326-L334.

4.4 Decisione di refresh in `detectChangesInView`

■ **TAVOLA 04.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: angular:packages/core/src/render3/instructions/change_detection.ts:L458-L517, angular:packages/core/src/render3/instructions/change_detection.ts:L227-L354.

4.5 SSR: stabilità e componente CMS lazy



Riferimenti: core-libs/setup/ssr/engine/cx-common-engine.ts:L41-L59, deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L203-L205, projects/storefrontapp/src/main.server.ts:L23-L24, angular:packages/platform-server/src/utils.ts:L367-L371, angular:packages/common/http/src/backend.ts:L125-L129, angular:packages/common/http/src/interceptor.ts:L243-L247, core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L80, angular:packages/core/src/pending_tasks.ts:L57-L59, angular:packages/core/src/pending_tasks_internal.ts:L63-L69.

5. Estratti di codice reale

5.1 Zoneless in testa ai provider (Angular)

angular:packages/core/src/application/create_application.ts:L66-L71

```ts
const allAppProviders = [
  provideZonelessChangeDetectionInternal(),
  errorHandlerEnvironmentInitializer,
  ...(ngDevMode ? [validAppIdInitializer] : []),
  ...(appProviders || []),
];
```

5.2 La demo Spartacus riattiva Zone

projects/storefrontapp/src/app/app.config.ts:L26-L35

```ts
export const appConfig: ApplicationConfig = {
  providers: [
    provideHttpClient(withFetch(), withInterceptorsFromDi()),
    provideClientHydration(withEventReplay(), withNoHttpTransferCache()),
    provideZoneChangeDetection({ eventCoalescing: true }),
    provideBrowserGlobalErrorListeners(),

    importProvidersFrom(AppModule),
  ],
};
```

5.3 `onMicrotaskEmpty` → tick globale

angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L45-L68

```ts
initialize(): void {
  if (this._onMicrotaskEmptySubscription) {
    return;
  }

  this._onMicrotaskEmptySubscription = this.zone.onMicrotaskEmpty.subscribe({
    next: () => {
      // `onMicroTaskEmpty` can happen _during_ the zoneless scheduler change detection because
      // zone.run(() => {}) will result in `checkStable` at the end of the `zone.run` closure
      // and emit `onMicrotaskEmpty` synchronously if run coalsecing is false.
      if (this.changeDetectionScheduler.runningTick) {
        return;
      }
      this.zone.run(() => {
        try {
```

```
          this.applicationRef.dirtyFlags |= ApplicationRefDirtyFlags.ViewTreeGlobal;
          this.applicationRef._tick();
        } catch (e) {
          this.applicationErrorHandler(e);
        }
      });
    },
  });
}
```

5.4 Il filtro dei listener e la mappa sorgente → flag

angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L138-L163

TS

```
notify(source: NotificationSource): void {
  if (!this.zonelessEnabled && source === NotificationSource.Listener) {
    // When the notification comes from a listener, we skip the notification unless the
    // application has enabled zoneless. Ideally, listeners wouldn't notify the scheduler at all
    // automatically. ...
    return;
  }

  switch (source) {
    case NotificationSource.MarkAncestorsForTraversal: {
      this.appRef.dirtyFlags |= ApplicationRefDirtyFlags.ViewTreeTraversal;
      break;
    }
    case NotificationSource.DebugApplyChanges:
    case NotificationSource.DeferBlockStateUpdate:
    case NotificationSource.MarkForCheck:
    case NotificationSource.Listener:
    case NotificationSource.SetInput: {
      this.appRef.dirtyFlags |= ApplicationRefDirtyFlags.ViewTreeCheck;
      break;
    }
    // ...
```

5.5 `Global` solo se non zoneless

angular:packages/core/src/application/application_ref.ts:L676-L697

TS

```
for (let {_lView} of this.allViews) {
  // When re-checking, only check views which actually need it.
  if (!useGlobalCheck && !requiresRefreshOrTraversal(_lView)) {
    continue;
  }
```

```
  const mode =
    useGlobalCheck && !this.zonelessEnabled
      ? // Global mode includes `CheckAlways` views.
        // When using zoneless, all root views must be explicitly marked for refresh, even if they are
        // `CheckAlways`.
        ChangeDetectionMode.Global
      : // Only refresh views with the `RefreshView` flag or views is a changed signal
        ChangeDetectionMode.Targeted;
  detectChangesInternal(_lView, mode);
  ranDetectChanges = true;
}

// If `markForCheck()` was called during view checking, it will have set the `ViewTreeCheck`
// flag. We clear the flag here because, for backwards compatibility, `markForCheck()`
// during view checking doesn't cause the view to be re-checked.
this.dirtyFlags &= ~ApplicationRefDirtyFlags.ViewTreeCheck;
```

5.6 Refresh mirato o solo traversal

angular:packages/core/src/render3/instructions/change_detection.ts:L481-L498

```ts
// Always refresh views marked for refresh, regardless of mode.
shouldRefreshView ||= !!(flags & LViewFlags.RefreshView);

// Refresh views when they have a dirty reactive consumer, regardless of mode.
shouldRefreshView ||= !!(consumer?.dirty && consumerPollProducersForChange(consumer));

shouldRefreshView ||= !!(ngDevMode && isExhaustiveCheckNoChanges());

// Mark the Flags and `ReactiveNode` as not dirty before refreshing the component, so that they
// can be re-dirtied during the refresh process.
if (consumer) {
  consumer.dirty = false;
}
lView[FLAGS] &= ~(LViewFlags.HasChildViewsToRefresh | LViewFlags.RefreshView);

if (shouldRefreshView) {
  refreshView(tView, lView, tView.template, lView[CONTEXT]);
} else if (flags & LViewFlags.HasChildViewsToRefresh) {
```

5.7 `PendingTasks` attorno all'`import()` (Spartacus)

core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L80

```ts
return defer(() => {
  // Register the dynamic import as a pending task so Angular's stability
  // counter stays > 0 until the chunk is loaded. This prevents SSR from
```

```
  // serializing and destroying the app while the import() is still in flight
  // (which would cause NG0205 — destroyed injector). The task is removed by
  // finalize(), which fires on complete, error, and unsubscribe.

  const removeTaskTracking = this.pendingTasks.add();
  return from(componentMapping.component()).pipe(
    finalize(removeTaskTracking)
  );
}).pipe(
  switchMap((component) =>
    this.defaultHandler.launcher(
      { ...componentMapping, component },
      viewContainerRef,
      elementInjector,
      module
    )
  )
);
```

5.8 `setInterval` vietato in SSR (Spartacus)

core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L62-L69

```ts
const backend: HttpBackendOptions = {
  loadPath,
  request: this.i18nextHttpClient,

  // Disable the periodical reloading. Otherwise SSR would not finish due to the pending task `setInterval()`
  // See source code of `i18next-http-backend` : https://github.com/i18next/i18next-http-backend/blob/00b7e8f67abf8372af17529b51190a7e8b17e3d8/lib/index.js#L40-L41
  reloadInterval: false,
};
```

5.9 OnPush + subscribe manuale + `markForCheck` (Spartacus)

core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L124-L131

```ts
ngOnInit() {
  this.subscription.add(
    this.slot$.pipe(tap((slot) => this.decorate(slot))).subscribe((value) => {
      this.components = value?.components || [];
      this.cd.markForCheck();
    })
  );
}
```

5.10 Signals + effect con chiamata HTTP (Spartacus)

feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L120-L140

```ts
protected registerSubscriptionCancellationEffect(): void {
  effect(() => {
    const mode = this.mode();
    const code = this.subscriptionCode();

    if (mode === 'cancel' && code) {
      this.actionsFacade
        .getEffectiveCancellationDate(code)
        .pipe(
          takeUntilDestroyed(this.destroyRef),
          catchError(() => {
            this.actionHandler.onError();
            return of(undefined);
          })
        )
        .subscribe((data) => {
          this.cancelData.set(data);
        });
    }
  });
}
```

5.11 `toObservable` → `switchMap` → `toSignal` (Spartacus)

feature-libs/subscription-billing/components/list/subscription-list.component.ts:L44-L64

```ts
listParams: WritableSignal<{
  sortCode: string | undefined;
  currentPage: number;
}> = signal({
  sortCode: undefined,
  currentPage: 0,
});

subscriptions$ = toObservable(this.listParams).pipe(
  switchMap((params) =>
    this.subscriptionFacade.getSubscriptionList(
      this.PAGE_SIZE,
      params.currentPage,
      params.sortCode
    )
  )
);
```

```
subscriptions = toSignal<SubscriptionList | null | undefined>(
  this.subscriptions$
);
```

5.12 Workaround Karma per Angular 21 (Spartacus)

core-libs/core/src/test.ts:L21-L31

```ts
// Angular 21 introduced a change that causes NG0100 errors in Karma tests.
// See: https://github.com/angular/angular-cli/issues/32047
// Angular fixed this for built-in test.ts: https://github.com/angular/angular-cli/pull/32049
// Since we use a custom test.ts, we must manually provide zone change detection.

//No longer necessary under vitest
@NgModule({
  providers: [provideZoneChangeDetection()],
})
class ZoneChangeDetectionModule {}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
NgZoneOptions.eventCoalescing	boolean	false	angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L197-L197	getNgZoneOptions angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L226-L226
NgZoneOptions.runCoalescing	boolean	false	angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L217-L217	angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L227-L227
scheduleInRootZone (interno)	boolean	false (SCHEDULE_IN_ROOT_ZONE_DEFAULT)	angular:packages/core/src/change_detection/scheduling/flags.ts:L9-L9	angular:packages/core/src/zone/ng_zone.ts:L148-L148, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L73-L76
ZONELESS_ENABLED (token interno)	boolean	true (factory)	angular:packages/core/src/change_detection/scheduling/zoneless_scheduling.ts:L66-L70	angular:packages/core/src/application/application_ref.ts:L269-L269, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L65-L65

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`REQUESTS_CONTRIBUTE_TO_STABILITY` (token interno HttpClient)	`boolean`	`true`	angular:packages/common/http/src/interceptor.ts:L217-L221	angular:packages/common/http/src/backend.ts:L61-L61, angular:packages/common/http/src/interceptor.ts:L244-L244
`BootstrapOptions.ngZone` / `ngZoneEventCoalescing`	deprecated	—	angular:packages/core/src/application/application_ref.ts:L100-L134	non letti da `bootstrapModuleFactory` (angular:packages/core/src/platform/platform_ref.ts:L51-L56)
Demo: `provideZoneChangeDetection({ eventCoalescing: true })`	provider	—	projects/storefrontapp/src/app/app.config.ts:L30-L30	ereditato in SSR projects/storefrontapp/src/app/app.config.server.ts:L30-L30
Demo: `polyfills`	`string[]`	`["zone.js"]`	projects/storefrontapp/project.json:L23-L23	builder
i18next `reloadInterval`	`false`	forzato `false`	core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L68-L68	`i18next-http-backend`
Feature toggle `a11yVocalizeDropdownItemCount`	`boolean`	`true`	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1063-L1063	core-libs/storefront/shared/components/ng-select-a11y/ng-select-a11y.directive.ts:L103-L103
Feature toggle `a11yStoreFinderFocusOnBackButton`	`boolean`	`false`	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1020-L1020	feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L77-L77

Non c'è nessuna chiave di configurazione Spartacus (`provideConfig`) che controlli la change detection: zone o zoneless è una scelta del bootstrap dell'applicazione. I feature toggle sono descritti in 42 — Feature toggle e deprecazioni.

---

7. Commenti e note notevoli dal sorgente

- angular:packages/core/src/change_detection/constants.ts:L37-L37: `@deprecated Use \`Eager\` instead.suChangeDetectionStrategy.Default. In Spartacus 5 componenti lo usano ancora (F152), per esempio` core-libs/storefront/cms-components/product/product-list/product-facet-navigation/active-facets/active-facets.component.ts:L31-L31`.
- angular:packages/core/src/metadata/directives.ts:L671-L671: `// TODO(jeanmeche): remove the ts-ignore when OnPush is the default`. OnPush non è ancora il default.
- angular:packages/core/src/change_detection/scheduling/zoneless_scheduling.ts:L78-L78: `// TODO(atscott): Remove in v19. Scheduler should be done with runOutsideAngular`. In v21 il TODO è ancora lì.

- angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L98-L100: TODO sul fatto che lo scheduler a microtask non andrebbe *mai* usato con ZoneJS attivo.
- angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L284-L284: `// TODO(alxhub): clean up and remove this workaround as a breaking change.` (forzatura di `ViewTreeGlobal` con Zone).
- angular:packages/core/src/zone/ng_zone.ts:L355-L369: TODO sul rientro di `checkStable`.
- angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L52-L57: spiega perché `NgZoneChangeDetectionScheduler` controlla `runningTick` (estratto 5.3).
- angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L75-L82: `PROVIDED_NG_ZONE` è documentato come controllo contro `bootstrapModule`, ma in v21 lo legge solo il warning zone+zoneless di `bootstrap()` (angular:packages/core/src/platform/bootstrap.ts:L96-L96).
- angular:packages/core/src/render3/reactivity/root_effect_scheduler.ts:L132-L132: `// TODO: what happens if this throws an error?`
- angular:packages/common/http/src/interceptor.ts:L214-L216: TODO che si chiede se `HttpClient` debba contribuire automaticamente alla stabilità.
- angular:packages/core/src/render3/reactivity/effect.ts:L152-L156: warning su `allowSignalWrites`, deprecato e ignorato.
- core-libs/core/src/test.ts:L21-L26: workaround NG0100 per Angular 21, con la nota "No longer necessary under vitest".
- core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L61-L65: spiegazione di NG0205 in SSR (estratto 5.7).
- core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L183-L184: "we do not know if content is created dynamically or not so we apply change detection anyway".
- integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L236-L236: "Run outside Angular zone since script execution doesn't need change detection".
- core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L155-L157: lo sblocco del focus viene ritardato con `setTimeout` "in case the host is using `ChangeDetectionStrategy.Default`".

8. Extension point

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`provideZoneChangeDetection(options)` / `NgZoneOptions`	Nei provider dell'app (`appConfig` o `AppModule`): Zone con `eventCoalescing`/`runCoalescing`	angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L153-L167; uso nella demo projects/storefrontapp/src/app/app.config.ts:L30-L30	50 — Ricette: config, DI, interceptor, eventi
`provideZonelessChangeDetection()`	Passaggio esplicito a zoneless (in v21 equivale a rimuovere `provideZoneChangeDetection`, più i controlli dev). Non usato da Spartacus	angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L362-L386	54 — Upgrade-safe
`PendingTasks` (`add`, `run`)	Tenere aperta la stabilità (SSR) durante lavoro asincrono non tracciato, per esempio `import()` o SDK esterni	angular:packages/core/src/pending_tasks.ts:L42-L79	51 — Ricette: CMS, outlet, route e layout

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`LazyComponentHandler` (`pendingTasks` protected, `providedIn: 'root'`)	Sostituzione via DI o estensione; esportato da `@spartacus/storefront`	`core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L27-L52`; export `core-libs/storefront/cms-structure/page/component/handlers/index.ts:L9-L9`	51
`NgZone.run` / `runOutsideAngular`	Rientrare in zona da callback di SDK esterni (modello CDC/OPF/SmartEdit), o uscirne per lavoro che non tocca la UI	`angular:packages/core/src/zone/ng_zone.ts:L232-L283`	50
`ChangeDetectorRef.markForCheck()`	In componenti `OnPush` che aggiornano lo stato in subscribe manuali (modello `PageSlotComponent`)	`angular:packages/core/src/change_detection/change_detector_ref.ts:L61-L119`	51
`signal` / `computed` / `effect` / `toSignal` / `toObservable`	Stato locale dei componenti custom; interop con le facade Spartacus (che espongono Observable)	`angular:packages/core/src/render3/reactivity/signal.ts:L72-L89`, `angular:packages/core/rxjs-interop/src/to_signal.ts:L132-L225`	53 — Ricette: feature, carrello, checkout, stili, i18n
`REQUESTS_CONTRIBUTE_TO_STABILITY`	Token **interno** di `@angular/common/http`: impostandolo a `false` le richieste non bloccano la stabilità. Non è API pubblica	`angular:packages/common/http/src/interceptor.ts:L217-L221`	54
`CxApiService.ngZone`	Disponibile ai web component CMS tramite `cxApi`; **non** esportato da `page/component/services/index.ts`	`core-libs/storefront/cms-structure/page/component/services/cx-api.service.ts:L73-L75`	51

Esempio d'uso di `PendingTasks.run` in un servizio custom (codice illustrativo, non presente nel repo):

```ts
// Illustrativo: blocca la stabilità SSR finché lo SDK non è pronto
private pendingTasks = inject(PendingTasks);
init(): void {
  this.pendingTasks.run(() => loadExternalSdk());
}
```

9. Trappole, casi limite, comportamenti controintuitivi

1. **Rimuovere** `provideZoneChangeDetection` **attiva zoneless.** In v21 il default è zoneless (`angular:packages/core/src/application/create_application.ts:L64-L71`). Nelle librerie Spartacus restano 152 file con `@Component(` senza `changeDetection` (cioè `Eager`), di cui 7 sono helper di test (per esempio `core-libs/storefront/shared/test/mock-media.component.ts` e i `*.testing.module.ts`): circa 145 componenti di produzione `Eager`, più 5 con `Default` (F151), e in zoneless le root view `CheckAlways` non vengono controllate globalmente (`angular:packages/core/src/application/application_ref.ts:L682-L689`). Uno stato aggiornato in una subscribe senza `markForCheck`, senza signal o senza `async` pipe può quindi non arrivare a schermo — `projects/storefrontapp/src/app/app.config.ts:L30-L30`.

2. **Con Zone,** `notify(Listener)` **non fa nulla.** È la zona a garantire la CD dopo un listener (angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L138-L149). Un componente che "funziona" con Zone perché un evento DOM innesca un tick globale può smettere di aggiornarsi in zoneless.
3. `zone.run` **e** `runOutsideAngular` **sono no-op in zoneless.** `NoopNgZone` chiama solo `fn()` (angular:packages/core/src/zone/ng_zone.ts:L564-L574). I `zone.run` di CDC, OPF e SmartEdit (§3.8) servono solo con Zone; in zoneless la UI si aggiornerebbe soltanto se il codice chiamato produce notifiche (`markForCheck`, signal, `async` pipe).
4. **TestBed è zoneless di default.** Il setup vitest non fornisce `provideZoneChangeDetection` (testing/setup-vitest.ts:L16-L19), mentre Karma lo aggiunge (core-libs/core/src/test.ts:L27-L31). Gli stessi componenti girano quindi in modalità diverse nei due runner e nella demo (angular:packages/core/testing/src/test_bed_compiler.ts:L932-L947).
5. **Opzioni di bootstrap deprecate senza effetto.** Lo schematic `modernize-app-migrated-from-2211_32-to-2211_36` aggiunge `{ ngZoneEventCoalescing: true }` a `bootstrapModule()` (core-libs/schematics/src/modernize-app-migrated-from-2211_32-to-2211_36/csr-and-ssr/update-main-ts.ts:L70-L73), ma in v21 `bootstrapModuleFactory` legge solo `options.applicationProviders` (angular:packages/core/src/platform/platform_ref.ts:L51-L56). Per avere coalescing serve `provideZoneChangeDetection`, che è ciò che fa la migrazione `221121_7`.
6. **Ordine dei provider nel percorso NgModule.** `provideZoneChangeDetection` messo nei provider di `AppModule` vince su quello zoneless anteposto da `bootstrapModuleFactory`, perché i provider del modulo vengono processati dopo (angular:packages/core/src/di/create_injector.ts:L49-L49, angular:packages/core/src/di/r3_injector.ts:L521-L521). Se invece si fornissero entrambi `provideZoneChangeDetection` e `provideZonelessChangeDetection`, in dev mode comparirebbe il warning `PROVIDED_BOTH_ZONE_AND_ZONELESS` (angular:packages/core/src/platform/bootstrap.ts:L95-L105).
7. `setInterval` **e polling bloccano l'SSR.** Un task ricorrente impedisce la stabilità (angular:packages/core/src/application/application_ref.ts:L172-L194), e `renderApplication` resta in attesa su `whenStable()`. Spartacus forza `reloadInterval: false` in i18next per questo motivo (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L66-L68).
8. `import()` **non è tracciato.** Senza il pending task manuale, l'SSR potrebbe serializzare e distruggere l'app mentre il chunk è ancora in volo, con errore NG0205 (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L61-L70). Un handler custom che carica codice in modo asincrono deve fare lo stesso.
9. **Due** `MAXIMUM_REFRESH_RERUNS`. Valgono 10 in `ApplicationRef` (angular:packages/core/src/application/application_ref.ts:L159-L159) e 100 nel traversal render3 (angular:packages/core/src/render3/instructions/change_detection.ts:L78-L78). Entrambi portano a `INFINITE_CHANGE_DETECTION`, ma in punti diversi.
10. `markForCheck()` **durante il check non provoca un nuovo giro** (angular:packages/core/src/application/application_ref.ts:L694-L697).
11. **I root effect non sono "microtask".** Il JSDoc dice che girano come microtask (angular:packages/core/src/render3/reactivity/effect.ts:L128-L129), ma il codice li esegue nel flush `RootEffects` di `synchronizeOnce`, dopo una notifica allo scheduler (angular:packages/core/src/application/application_ref.ts:L656-L659).
12. `effect` **con chiamate HTTP.** In `SubscriptionActionsModalComponent` ogni cambio di `mode` o `code` avvia una nuova richiesta. La subscription è legata a `takeUntilDestroyed(this.destroyRef)`, cioè alla distruzione del componente, e non al rerun dell'effect (non c'è `onCleanup`): le richieste precedenti non vengono annullate (feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L120-L140).

13. `effect()` **dentro un metodo.** `NgSelectA11yDirective.vocalizeItemCount()` funziona solo perché il costruttore lo invoca in injection context (core-libs/storefront/shared/components/ng-select-a11y/ng-select-a11y.directive.ts:L103-L105, core-libs/storefront/shared/components/ng-select-a11y/ng-select-a11y.directive.ts:L189-L190). Chiamato altrove senza `injector`, `effect()` lancia un errore: in dev mode dall'asserzione, in produzione da `inject(Injector)` (angular:packages/core/src/render3/reactivity/effect.ts:L148-L150, angular:packages/core/src/render3/reactivity/effect.ts:L158-L158). Lo stesso schema c'è in `SubscriptionActionsModalComponent` (feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L106-L107).
14. `toSignal` **in un servizio root.** La subscription vive quanto l'environment injector root, e senza `initialValue` `userId()` può valere `undefined` prima della prima emissione (core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L46-L46, angular:packages/core/rxjs-interop/src/to_signal.ts:L161-L166).
15. `toObservable` **non è sincrono.** In `SubscriptionListComponent` la prima richiesta parte al primo flush dell'effect interno, non alla costruzione (angular:packages/core/rxjs-interop/src/to_observable.ts:L55-L67, feature-libs/subscription-billing/components/list/subscription-list.component.ts:L52-L60).
16. `AsyncPipe` **e valori sincroni.** Se l'Observable emette in modo sincrono alla subscribe, la pipe non chiama `markForCheck` perché il valore è già nel primo render (angular:packages/common/src/pipes/async_pipe.ts:L182-L189).
17. `isStable` **omonimi.** `ConfigInitializerService.isStable` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L42-L42) e l'`isStable` del carrello usato in checkout (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L95-L95) non hanno niente a che vedere con la stabilità di `ApplicationRef`.
18. `ChangeDetectionStrategy.Default` **è deprecato** (angular:packages/core/src/change_detection/constants.ts:L34-L40), ma è solo un alias di `Eager`: i 5 componenti Spartacus che lo usano funzionano come prima.
19. **Zone.js lato server.** DA VERIFICARE: non è confermato che il bundle server riceva `zone.js` attraverso il builder. `main.server.ts` non importa zone.js (projects/storefrontapp/src/main.server.ts:L23-L26) e il polyfill è dichiarato nel target browser (projects/storefrontapp/project.json:L23-L23), ma `node_modules` non è installato e la logica del builder non si può leggere. Se Zone mancasse sul server, la config SSR, che eredita `provideZoneChangeDetection` (projects/storefrontapp/src/app/app.config.server.ts:L17-L30), fallirebbe con `MISSING_ZONEJS` (angular:packages/core/src/zone/ng_zone.ts:L151-L158).

---

10. Come fare debug

- **Capire se l'app gira con Zone.** Cercare `provideZoneChangeDetection` nei provider (projects/storefrontapp/src/app/app.config.ts:L30-L30) e `zone.js` nei `polyfills` (projects/storefrontapp/project.json:L23-L23). A runtime, `NgZone.isInAngularZone()` restituisce `false` se Zone non c'è (angular:packages/core/src/zone/ng_zone.ts:L193-L196). Se l'`NgZone` iniettato è un `NoopNgZone` (privato, `ɵNoopNgZone`), l'app è zoneless (angular:packages/core/src/zone.ts:L9-L10).
- **Warning in console.** `PROVIDED_BOTH_ZONE_AND_ZONELESS` indica provider in conflitto (angular:packages/core/src/platform/bootstrap.ts:L95-L105). Il messaggio "The application is using zoneless change detection, but is still loading Zone.js" indica un polyfill zone.js superfluo (angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L362-L386).
- `MISSING_ZONEJS`. Si è chiesto `provideZoneChangeDetection` senza caricare zone.js (angular:packages/core/src/zone/ng_zone.ts:L151-L158).

- **SSR che non termina o va in timeout.** Sottoscrivere `inject(ApplicationRef).isStable` e registrare le emissioni; l'Observable deriva da `PendingTasksInternal` (angular:packages/core/src/application/application_ref.ts:L326-L334). Cercare `setInterval`, `interval()` o polling avviati all'avvio (angular:packages/core/src/application/application_ref.ts:L172-L194), richieste HTTP che non completano (angular:packages/common/http/src/backend.ts:L125-L129, angular:packages/common/http/src/interceptor.ts:L243-L247) e `PendingTasks.add()` senza la rimozione corrispondente.
- **NG0205 (injector distrutto) in SSR.** Lavoro asincrono non tracciato che riprende dopo la serializzazione: aggiungere un pending task come in `LazyComponentHandler` (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71).
- **NG0100 (espressione cambiata dopo il check).** In dev mode `tickImpl` esegue `checkNoChanges()` su tutte le view (angular:packages/core/src/application/application_ref.ts:L591-L619). Nei test Karma di Angular 21 vedi il workaround di core-libs/core/src/test.ts:L21-L35.
- `INFINITE_CHANGE_DETECTION`.
  - Messaggio "Infinite change detection while refreshing application views": più di 10 giri in `synchronize` (angular:packages/core/src/application/application_ref.ts:L625-L649).
  - Messaggio "endless change notifications": 100 notifiche consecutive con lo scheduler a microtask (angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L36-L57).

  In entrambi i casi cercare scritture di stato (o di signal) in hook o effect che a loro volta sporcano la view.
- **Componente OnPush che non si aggiorna.** Verificare che lo stato passi per `async` pipe, `markForCheck()` (angular:packages/core/src/render3/view_ref.ts:L163-L165), un input cambiato per riferimento (angular:packages/core/src/render3/instructions/shared.ts:L318-L325) o un signal letto nel template (angular:packages/core/src/render3/reactive_lview_consumer.ts:L43-L53). Mutare in profondità un oggetto non cambia il riferimento, e un signal non notifica se `Object.is` dà vero (angular:packages/core/primitives/signals/src/equality.ts:L17-L18).
- **Effect che non parte.** Controllare il feature toggle: `a11yStoreFinderFocusOnBackButton` vale `false` di default (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1020-L1020). Controllare anche che il signal sia letto *dentro* la funzione dell'effect e non in un callback asincrono, dove `producerAccessed` non registra nulla (angular:packages/core/primitives/signals/src/graph.ts:L204-L216).

---

11. Domande di autoverifica

1. **In Angular 21, cosa succede se un'app non fornisce né** `provideZoneChangeDetection` **né** `provideZonelessChangeDetection`**?** L'app è zoneless: `internalCreateApplication` antepone `provideZonelessChangeDetectionInternal()`, che fornisce `NoopNgZone` e `ZONELESS_ENABLED=true` — angular:packages/core/src/application/create_application.ts:L64-L71, angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L388-L394.
2. **Perché la demo Spartacus controlla ancora i componenti** `Eager` **a ogni evento?** Perché fornisce `provideZoneChangeDetection({ eventCoalescing: true })` — projects/storefrontapp/src/app/app.config.ts:L30-L30. Su `onMicrotaskEmpty` lo scheduler di zona imposta `ViewTreeGlobal` — angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L58-L64 — e `synchronizeOnce` usa la modalità `Global` solo se non è zoneless — angular:packages/core/src/application/application_ref.ts:L682-L689.
3. **Da cosa dipende** `ApplicationRef.isStable` **in v21, e perché è importante per l'SSR?** Dipende da `PendingTasksInternal.hasPendingTasksObservable` — angular:packages/core/src/application/

application_ref.ts:L326-L334. renderApplication attende whenStable() prima di serializzare l'HTML — angular:packages/platform-server/src/utils.ts:L362-L373.

4. **Perché** LazyComponentHandler **chiama** pendingTasks.add() **e quando rimuove il task?** Per evitare che l'SSR serializzi e distrugga l'app durante l'import() del chunk (NG0205). Il task viene rimosso da finalize, cioè su complete, error o unsubscribe — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71.
5. **Cosa fa Angular quando cambia un signal letto nel template di un componente OnPush figlio di altri OnPush non sporchi?** Il consumer della view chiama markAncestorsForTraversal, che imposta HasChildViewsToRefresh sugli antenati e notifica lo scheduler — angular:packages/core/src/render3/reactive_lview_consumer.ts:L43-L53, angular:packages/core/src/render3/util/view_utils.ts:L264-L280. Al tick gli antenati vengono solo attraversati, mentre la view con il consumer sporco viene rifatta — angular:packages/core/src/render3/instructions/change_detection.ts:L481-L517.
6. **Con Zone attiva, perché** notify(NotificationSource.Listener) **non pianifica un tick?** Lo scheduler la ignora quando zoneless non è attivo, perché i listener girano in zona e la zona produrrà la CD — angular:packages/core/src/change_detection/scheduling/zoneless_scheduling_impl.ts:L138-L149.
7. **Che differenza c'è tra un view effect e un root effect, e dove vengono eseguiti?** Il view effect si crea quando nell'injector c'è un ViewContext e viene eseguito da runEffectsInView durante refresh o traversal della view. Il root effect viene accodato in EffectScheduler ed eseguito nel flush RootEffects di synchronizeOnce — angular:packages/core/src/render3/reactivity/effect.ts:L158-L182, angular:packages/core/src/render3/reactivity/view_effect_runner.ts:L11-L43, angular:packages/core/src/application/application_ref.ts:L656-L659.
8. **Perché lo schematic che aggiunge** ngZoneEventCoalescing: true **a** bootstrapModule() **non ha effetto in v21?** Perché bootstrapModuleFactory usa solo options.applicationProviders, e BootstrapOptions.ngZoneEventCoalescing è deprecato — angular:packages/core/src/platform/platform_ref.ts:L51-L56, angular:packages/core/src/application/application_ref.ts:L100-L134, core-libs/schematics/src/modernize-app-migrated-from-2211_32-to-2211_36/csr-and-ssr/update-main-ts.ts:L70-L73.
9. **Perché i test vitest di Spartacus girano in una modalità di CD diversa dai test Karma?** Il TestBed di v21 antepone provideZonelessChangeDetectionInternal() — angular:packages/core/testing/src/test_bed_compiler.ts:L932-L947. I test.ts Karma aggiungono provideZoneChangeDetection() — core-libs/core/src/test.ts:L27-L31 — mentre il setup vitest non lo fa — testing/setup-vitest.ts:L16-L19.
10. **Un** toSignal(obs$) **senza opzioni, in un servizio** providedIn: 'root'**: quanto vale prima della prima emissione e quando si chiude la subscription?** Vale undefined — angular:packages/core/rxjs-interop/src/to_signal.ts:L161-L166. La subscription si chiude su DestroyRef.onDestroy dell'injector, cioè l'environment injector root, quindi alla distruzione dell'app — angular:packages/core/rxjs-interop/src/to_signal.ts:L198-L200, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L46-L46.

TAVOLA 04.1 **4.1 Scelta del motore al bootstrap (demo Spartacus)**

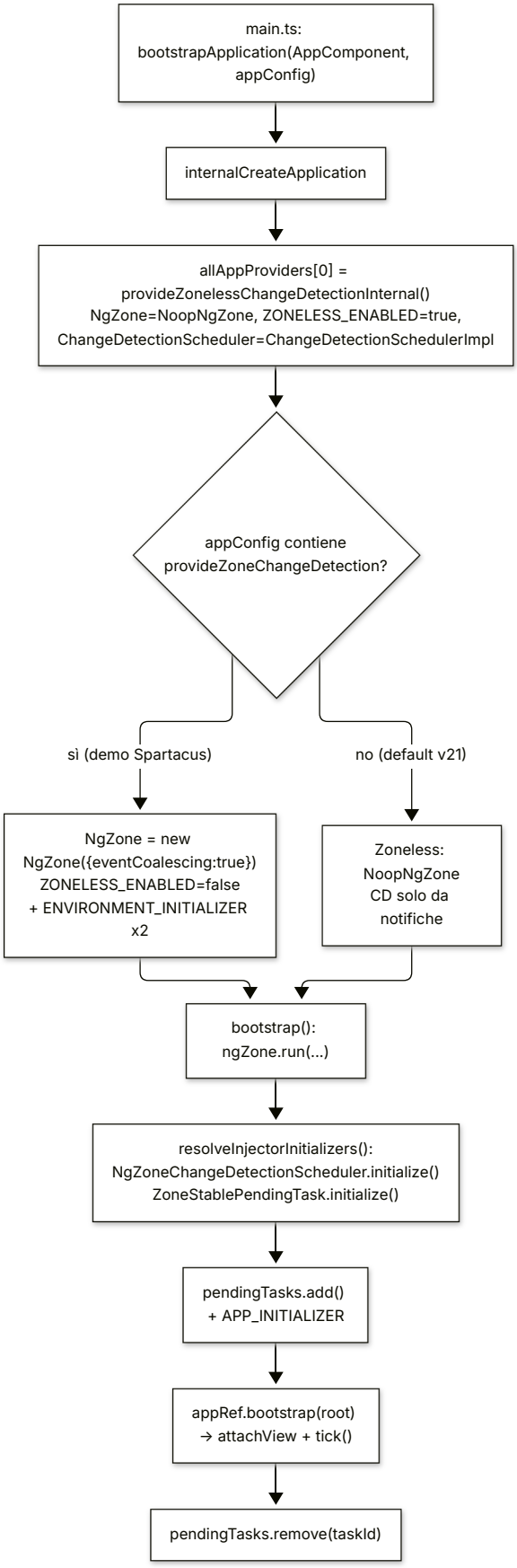


TAVOLA 04.2 **4.2 Ciclo con Zone.js ed event coalescing**

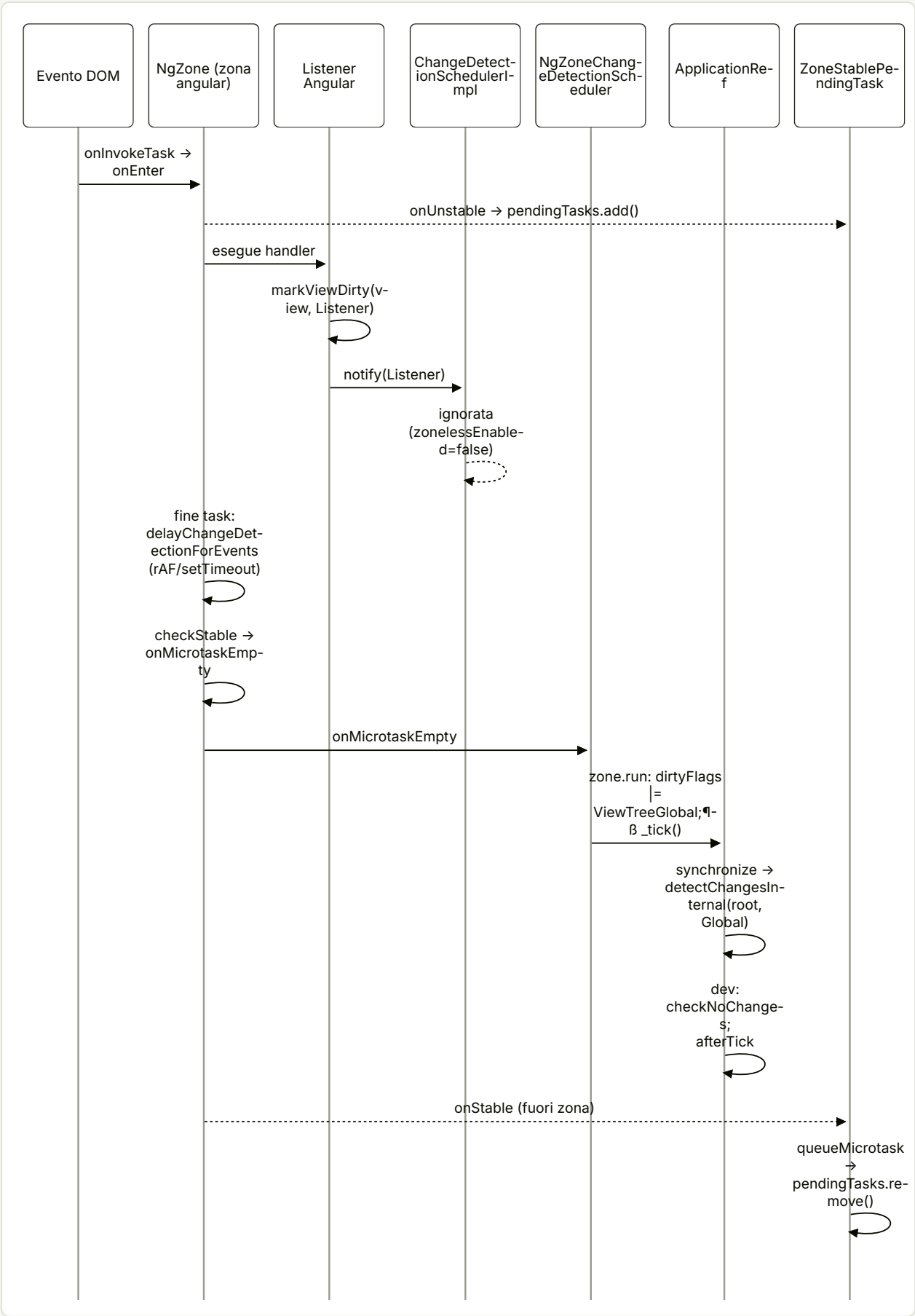
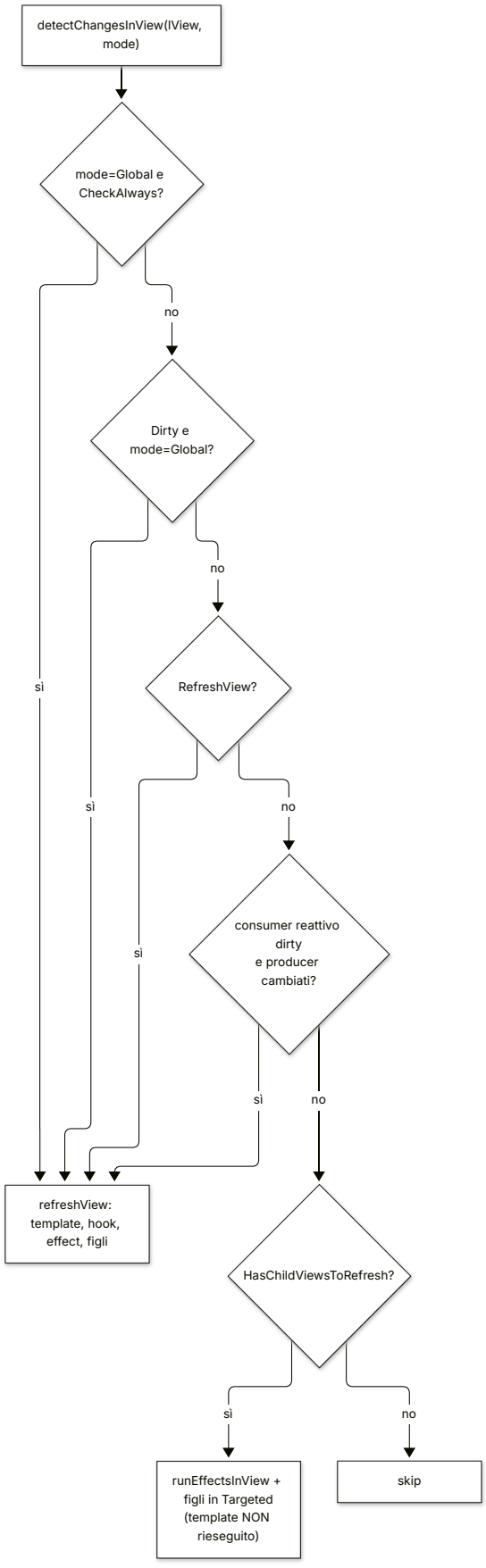


TAVOLA 04.3 **4.4 Decisione di refresh in detectChangesInView**



05

Il Router di Angular

La pipeline URL → `UrlTree` → recognize → guard → resolver → attivazione, e i suoi eventi.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Angular @ `v21.2.23` (prefisso `angular:`), `@ngrx/router-store` / `@ngrx/store` 21.0.1 (prefisso `deps:`).

---

1. A cosa serve e perché esiste

Il Router di Angular trasforma un URL in un albero di componenti attivati. Lo fa con una pipeline fissa: parsing dell'URL in un `UrlTree`, *recognize* (abbinamento dei segmenti alla configurazione `Routes`), esecuzione dei guard, esecuzione dei resolver, attivazione dei componenti. Ogni fase emette eventi osservabili su `router.events`.

Spartacus usa questo router in modo poco convenzionale, e il capitolo esiste per spiegare proprio queste differenze:

1. **Nessuna route statica nell'app.** `AppRoutingModule` chiama `RouterModule.forRoot([])` con un array vuoto — core-libs/storefront/router/app-routing.module.ts:L13-L19. Le route arrivano dalle feature (`RouterModule.forChild`) con `path: null` e un nome semantico in `data.cxRoute` — feature-libs/cart/base/root/cart-base-root.module.ts:L62-L74. Il path reale viene scritto a runtime da `ConfigurableRoutesService` leggendo `routing.routes` dalla configurazione — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L48-L52.
2. **Una route catch-all governa quasi tutto.** In fondo alla config viene aggiunta `{ path: '**', canActivate: [CmsPageGuard], component: PageLayoutComponent }` — core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25. Qualsiasi URL che non corrisponde a una route di feature diventa una "content page" del CMS.
3. **Il guard carica i dati.** `CmsPageGuard` non si limita a dire sì/no: carica la pagina CMS dal backend, risolve i mapping dei componenti (anche in lazy loading), esegue i guard dei componenti, carica le traduzioni e, se serve, riscrive la configurazione del router — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84, core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L98.
4. **Lo stato del router vive anche in NgRx.** `@ngrx/router-store` replica ogni navigazione nello store `router`, serializzata da un `CustomSerializer` che calcola il `PageContext` (tipo e id della pagina CMS) — core-libs/core/src/routing/routing.module.ts:L73-L92, core-libs/core/src/routing/store/reducers/router.reducer.ts:L112-L167. `CmsPageGuard` legge il contesto della navigazione *in corso* da lì — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L58-L60.
5. **La navigazione iniziale è bloccante e trattenuta.** `initialNavigation: 'enabledBlocking'` fa attendere il bootstrap alla prima navigazione — angular:packages/router/src/provide_router.ts:L441-L475 — e Spartacus fornisce `LOCATION_INITIALIZED` per rimandarne l'avvio finché configurazione, autenticazione e altri initializer non sono pronti — core-libs/core/src/routing/routing.module.ts:L54-L71.

Per seguire il capitolo servono i concetti di DI (01 — Dependency Injection) e di bootstrap (02 — NgModule, standalone e bootstrap). Il routing "semantico" di Spartacus visto dal lato delle feature (URL, page context, contenuti) è approfondito in 14 — Routing; qui l'obiettivo è il punto di contatto con il Router di Angular.

2. Mappa dei file

Angular (`/tmp/angular`, prefisso `angular:`)

PATH	RUOLO
`angular:packages/router/src/models.ts`	Tipi `Route`, `UrlMatcher`, `UrlMatchResult`, `CanActivateFn`, `GuardResult`, `DeprecatedGuard`
`angular:packages/router/src/utils/config.ts`	`validateConfig` (dev mode), `getOrCreateRouteInjectorIfNeeded`
`angular:packages/router/src/utils/config_matching.ts`	`match`, `matchWithChecks` (matching + `canMatch`)
`angular:packages/router/src/shared.ts`	`defaultUrlMatcher`, `matchParts`
`angular:packages/router/src/recognize.ts`	`recognize()` e classe `Recognizer`
`angular:packages/router/src/operators/check_guards.ts`	`checkGuards`, `runCanActivate`, `runCanMatchGuards`
`angular:packages/router/src/operators/prioritized_guard_value.ts`	Priorità dei risultati dei guard paralleli
`angular:packages/router/src/utils/preactivation.ts`	`getTokenOrFunctionIdentity` (guard di classe vs funzione)
`angular:packages/router/src/operators/resolve_data.ts`	`resolveData`
`angular:packages/router/src/navigation_transition.ts`	`NavigationTransitions.setupNavigations`: la pipeline e i suoi eventi
`angular:packages/router/src/router.ts`	`Router`: `config`, `resetConfig`, `parseUrl`/`serializeUrl`, inoltro di `events`
`angular:packages/router/src/url_tree.ts`	`UrlTree`, `UrlSegmentGroup`, `UrlSerializer`, `DefaultUrlSerializer`
`angular:packages/router/src/router_module.ts`	`RouterModule.forRoot/forChild`, `ROUTER_PROVIDERS`, scroller
`angular:packages/router/src/provide_router.ts`	`withEnabledBlockingInitialNavigation`
`angular:packages/router/src/router_scroller.ts`	`RouterScroller` ed eventi `Scroll`
`angular:packages/router/src/events.ts`	`EventType`, classi evento

NgRx (`/tmp/deps`, prefisso `deps:`)

PATH	RUOLO
`deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs`	`StoreRouterConnectingModule`, `provideRouterStore`, servizio che traduce gli eventi del router in azioni
`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs`	`State`: applicazione delle azioni ai reducer

Spartacus

PATH	RUOLO
`core-libs/storefront/router/app-routing.module.ts`	`AppRoutingModule`: `RouterModule.forRoot([], …)` + initializer di `OnNavigateService`
`core-libs/storefront/router/on-navigate.service.ts`	Reset di scroll e focus a ogni navigazione

PATH	RUOLO
core-libs/storefront/router/config/default-on-navigate-config.ts	Default di `enableResetViewOnNavigate`
core-libs/core/src/routing/routing.module.ts	`RoutingModule` core: NgRx router-store, `CustomSerializer`, initializer route, `LOCATION_INITIALIZED`
core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts	Token `LOCATION_INITIALIZED_MULTI` e helper
core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts	Applica `routing.routes` alle `Route` (path/matcher) e chiama `resetConfig`
core-libs/core/src/routing/configurable-routes/routes-config.ts	`RouteConfig` (`paths`, `matchers`, `paramsMapping`, `disabled`, `protected`, `authFlow`)
core-libs/core/src/routing/configurable-routes/config/routing-config.ts	`RoutingConfig`, `RouteLoadStrategy`
core-libs/core/src/routing/configurable-routes/routing-config.service.ts	Lettura config, mappa inversa path→nome, load strategy
core-libs/core/src/routing/services/url-matcher.service.ts	Fabbrica di `UrlMatcher` (falsy, da path, combinato, opposto, glob)
core-libs/core/src/routing/url-matcher/default-url-matcher.ts	Token `DEFAULT_URL_MATCHER`
core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts	`getSuffixUrlMatcher` (`**/marker/:param`)
core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts	`PRODUCT_DETAILS_URL_MATCHER`
core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts	Route PLP e il loro ordine
core-libs/storefront/cms-structure/routing/routing.module.ts	`RoutingModule` storefront: config di default, `BEFORE_CMS_PAGE_GUARD`
core-libs/storefront/cms-structure/routing/default-routing-config.ts	`defaultRoutesConfigFactory`
core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts	Route catch-all `**`
core-libs/storefront/cms-structure/guards/cms-page.guard.ts	`CmsPageGuard`
core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts	Attivazione pagina CMS / not found
core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts	Token `BEFORE_CMS_PAGE_GUARD`
core-libs/storefront/cms-structure/services/guards-composer.ts	Composizione sequenziale dei guard
core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts	Route CMS dinamiche (`childRoutes`)
core-libs/core/src/routing/protected-routes/protected-routes.guard.ts	Storefront chiuso (`routing.protected`)
core-libs/core/src/routing/external-routes/external-routes.service.ts	Route verso sistemi esterni (`routing.internal`)

PATH	RUOLO
core-libs/core/src/routing/store/reducers/router.reducer.ts	Reducer `router` + `CustomSerializer`
core-libs/core/src/routing/store/effects/router.effect.ts	Pulizia delle route CMS dinamiche
core-libs/core/src/routing/facade/routing.service.ts	Facade di navigazione
core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts	Comandi semantici `{ cxRoute, params }` → segmenti
core-libs/core/src/site-context/services/site-context-url-serializer.ts	`UrlSerializer` con prefisso di site context

3. Come funziona passo per passo

3.1 Il modello `Route` di Angular in quattro regole

- Una route corrisponde ai segmenti con `path` (+ `pathMatch: 'prefix' | 'full'`) **oppure** con un `matcher` di tipo `UrlMatcher` — angular:packages/router/src/models.ts:L606-L613. Il JSDoc di `UrlMatcher` dice che "Cannot be used together with `path` and `pathMatch`" — angular:packages/router/src/models.ts:L158-L185 — e in dev mode `validateConfig` lancia un errore se una route ha sia `path` (truthy) sia `matcher` — angular:packages/router/src/utils/config.ts:L167-L172.
- Una route deve avere almeno uno dei due; il controllo è `route.path === void 0 && route.matcher === void 0` — angular:packages/router/src/utils/config.ts:L185-L190. Quindi `path: null` supera la validazione: è il trucco su cui si reggono le route di feature di Spartacus (§3.4).
- Un `path` non può iniziare con `/` — angular:packages/router/src/utils/config.ts:L191-L196.
- Un `UrlMatcher` riceve `(segments, group, route)` e ritorna `UrlMatchResult | null`, dove il risultato contiene i segmenti `consumed` e i `posParams` come `UrlSegment` — angular:packages/router/src/models.ts:L153-L156.

I guard sono funzioni `CanActivateFn (route, state) => MaybeAsync<GuardResult>` — angular:packages/router/src/models.ts:L915-L918 — dove `GuardResult` è `boolean | UrlTree | RedirectCommand` — angular:packages/router/src/models.ts:L85-L85. `canActivate` accetta ancora anche `DeprecatedGuard` (token di classe o stringa) — angular:packages/router/src/models.ts:L669-L681 — tipo marcato `@deprecated` — angular:packages/router/src/models.ts:L50-L64. Spartacus usa proprio questa forma deprecata: `canActivate: [CmsPageGuard]` con `CmsPageGuard` classe `@Injectable` — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L23-L27.

3.2 Matching: `match()`, `defaultUrlMatcher`, ordine delle route

1. `Recognizer.processSegment` prova le route **nell'ordine della configurazione** e ritorna la prima che corrisponde; un `NoMatch` fa passare alla successiva — angular:packages/router/src/recognize.ts:L235-L267. L'ordine è dunque semantico: ne dipendono la posizione della catch-all `**` e l'ordine delle route PLP (§9).
2. Se la route ha `redirectTo` il redirect viene espanso, altrimenti si passa a `matchSegmentAgainstRoute` — angular:packages/router/src/recognize.ts:L298-L321.
3. `matchSegmentAgainstRoute` invoca `matchWithChecks` — angular:packages/router/src/recognize.ts:L416-L437 — che esegue `match()` e, solo se c'è un match, crea l'eventuale injector della route (`route.`

`providers`) ed esegue i guard `canMatch`; un risultato non-`true` diventa `noMatch` — angular:packages/router/src/utils/config_matching.ts:L54-L80, angular:packages/router/src/utils/config.ts:L31-L43.

4. `match()` gestisce direttamente `path: ''`: con `pathMatch: 'full'` fallisce se restano segmenti o figli, altrimenti consuma zero segmenti — angular:packages/router/src/utils/config_matching.ts:L82-L99.
5. Per gli altri path usa `route.matcher || defaultUrlMatcher` — angular:packages/router/src/utils/config_matching.ts:L101-L103 — e converte i `posParams` in stringhe, fondendovi i matrix params dell'ultimo segmento consumato — angular:packages/router/src/utils/config_matching.ts:L105-L112.
6. `defaultUrlMatcher` divide `route.path!` per `/`; senza wildcard rifiuta URL più corti della config e, con `pathMatch: 'full'`, quelli più lunghi o con figli — angular:packages/router/src/shared.ts:L153-L181. In v21 supporta **un solo** `**` anche in mezzo al path, consumando tutti i segmenti — angular:packages/router/src/shared.ts:L183-L217. Le parti che iniziano con `:` diventano parametri, le altre sono confrontate letteralmente — angular:packages/router/src/shared.ts:L120-L136.
7. Per una route `'**'` i figli del segment group vengono svuotati (issue #40089) — angular:packages/router/src/recognize.ts:L416-L437; per le altre, i segmenti rimanenti sono abbinati ricorsivamente ai `children` — angular:packages/router/src/recognize.ts:L440-L492.
8. `createSnapshot` costruisce l'`ActivatedRouteSnapshot` con params, `getData(route)`, `routeConfig` e applica l'ereditarietà di params/data secondo `paramsInheritanceStrategy` (default `'emptyOnly'`) — angular:packages/router/src/recognize.ts:L377-L400, angular:packages/router/src/recognize.ts:L46-L66.
9. Alla fine l'URL dello stato viene serializzato con l'`UrlSerializer` **iniettato** — angular:packages/router/src/recognize.ts:L97-L114. Se nulla corrisponde, l'errore è `Cannot match any routes. URL Segment: ...` — angular:packages/router/src/recognize.ts:L88-L95. In Spartacus questo non accade quasi mai, perché la catch-all `**` corrisponde sempre.

3.3 Pipeline di navigazione ed eventi

`NavigationTransitions.setupNavigations` è una catena RxJS su un `BehaviorSubject` di transizioni, con `switchMap`: una nuova navigazione cancella quella in corso — angular:packages/router/src/navigation_transition.ts:L440-L446. I passi:

1. **Same-URL.** Se il router ha già navigato e `onSameUrlNavigation !== 'reload'`, una navigazione verso lo stesso URL viene saltata con `NavigationSkipped` — angular:packages/router/src/navigation_transition.ts:L494-L513.
2. `NavigationStart`, con l'URL serializzato dal serializer iniettato; poi `recognize` sulla `router.config` corrente — angular:packages/router/src/navigation_transition.ts:L515-L545. L'operatore aggiunge alla transizione `targetSnapshot` e `urlAfterRedirects` — angular:packages/router/src/operators/recognize.ts:L19-L40.
3. `BeforeRoutesRecognized` (interno) e poi `RoutesRecognized` — angular:packages/router/src/navigation_transition.ts:L548-L573. È qui che NgRx dispatcha `ROUTER_NAVIGATION` (§3.8).
4. `GuardsCheckStart`, calcolo dei guard (`getAllRouteGuards`) e `checkGuards` — angular:packages/router/src/navigation_transition.ts:L630-L649.
5. `checkGuards` esegue prima tutti i `canDeactivate` e, solo se tutti `true`, i `canActivate` — angular:packages/router/src/operators/check_guards.ts:L56-L78. Per ogni route da attivare, in sequenza, emette `ChildActivationStart`/`ActivationStart`, esegue `canActivateChild` e `canActivate`, e si ferma al primo risultato non-`true` — angular:packages/router/src/operators/check_guards.ts:L93-L111.
6. Dentro una singola route i guard partono **in parallelo** (`combineLatest`), ma `prioritizedGuardValue` restituisce il primo risultato non-`true` in ordine di dichiarazione, aspettando i guard precedenti — angular:packages/router/src/operators/prioritized_guard_value.ts:L18-L45.

7. Ogni guard viene risolto con `getTokenOrFunctionIdentity` sull'injector della route: se l'oggetto ha un metodo `canActivate` è trattato come classe, altrimenti come funzione eseguita in `runInInjectionContext` — angular:packages/router/src/operators/check_guards.ts:L149-L172. Il token viene cercato con `injector.get`; se manca ed è una funzione non `@Injectable`, viene usato così com'è — angular:packages/router/src/utils/preactivation.ts:L66-L78.
8. **Esito dei guard.** Un `UrlTree`/`RedirectCommand` viene lanciato come `redirectingNavigationError`; `false` cancella con codice `GuardRejected` — angular:packages/router/src/navigation_transition.ts:L651-L671. Il redirect produce `NavigationCancel` seguito da `RedirectRequest` — angular:packages/router/src/navigation_transition.ts:L885-L901 — che il `Router` trasforma in una nuova navigazione che eredita la promise della precedente — angular:packages/router/src/router.ts:L214-L241.
9. **Resolver.** `ResolveStart`/`ResolveEnd` incorniciano `resolveData`; un resolver che non emette cancella con `NoDataFromResolver` — angular:packages/router/src/navigation_transition.ts:L677-L714. Le route vengono processate in sequenza (`concatMap`) — angular:packages/router/src/operators/resolve_data.ts:L55-L58 — mentre i resolver di una stessa route girano in parallelo (`mergeMap` sulle chiavi di `resolve`) — angular:packages/router/src/operators/resolve_data.ts:L111-L113. Spartacus non dichiara alcun `resolve:`, `canMatch:`, `canDeactivate:` né `canActivateChild:` nelle librerie né nella demo app: le uniche occorrenze di `resolve: {` sono nei file `vitest.config.ts` (config di Vite, non route), es. core-libs/storefront/vitest.config.ts:L14-L14, feature-libs/cart/vitest.config.ts:L16-L16.
10. **Attivazione.** Dopo i `loadComponent` viene invocato l'hook `afterPreactivation()` — angular:packages/router/src/navigation_transition.ts:L717-L742 — poi `ActivateRoutes.activate`, `NavigationEnd` e aggiornamento del titolo — angular:packages/router/src/navigation_transition.ts:L782-L810.
11. **Errori.** Gli errori non di cancellazione diventano `NavigationError`, passando prima da `navigationErrorHandler`, che può restituire un `RedirectCommand` — angular:packages/router/src/navigation_transition.ts:L905-L941.

Gli eventi pubblici sono 17 (`EventType`, da `NavigationStart` a `NavigationSkipped`) — angular:packages/router/src/events.ts:L33-L51. Il `Router` li riceve dalla pipeline e li inoltra al `Subject` pubblico `router.events` nella stessa sottoscrizione, sincronamente, dopo aver aggiornato il proprio stato — angular:packages/router/src/router.ts:L195-L195, angular:packages/router/src/router.ts:L245-L250.

3.4 Dalla config Spartacus alla `Router.config`

1. **Registrazione.** `RouterModule.forRoot(routes, config)` registra `ROUTES` (multi), `ROUTER_CONFIGURATION`, location strategy, scroller, initial navigation — angular:packages/router/src/router_module.ts:L133-L166. `forChild` aggiunge solo un altro `ROUTES` multi — angular:packages/router/src/router_module.ts:L184-L189. `Router.config` è `inject(ROUTES).flat()` — angular:packages/router/src/router.ts:L163-L163.
2. **Route di feature senza path.** Le feature dichiarano `path: null` (con `// @ts-ignore`), `canActivate: [CmsPageGuard]`, `component: PageLayoutComponent` e `data.cxRoute` — feature-libs/cart/base/root/cart-base-root.module.ts:L62-L74. Il path viene dalla config: `cart: { paths: ['cart'] }` — feature-libs/cart/base/root/config/default-cart-routing-config.ts:L12-L12.
3. **Config delle route.** `RouteConfig` offre `paths`, `matchers` (funzioni o token di `UrlMatcherFactory`), `paramsMapping`, `disabled`, `protected`, `authFlow` — core-libs/core/src/routing/configurable-routes/routes-config.ts:L15-L57. Lo storefront registra i default con `provideDefaultConfigFactory(defaultRoutesConfigFactory)` — core-libs/storefront/cms-structure/routing/routing.module.ts:L22-L26 — tra cui `home: ['']`, `notFound: ['not-found']`, `product: ['product/:productCode/:name', 'product/:productCode']` — core-libs/storefront/cms-structure/routing/default-routing-config.ts:L17-L73.
4. **Initializer.** `RoutingModule.forRoot()` (core) registra con `provideAppInitializer` la chiamata a `ConfigurableRoutesService.init()` — core-libs/core/src/routing/routing.module.ts:L93-L95, core-

libs/core/src/routing/routing.module.ts:L37-L42. `init()` è idempotente (`initCalled`) — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L31-L43.

5. **`resetConfig`.** `configure()` ottiene il `Router` dall'`Injector` (non dal costruttore, per la dipendenza ciclica con `APP_INITIALIZER`) e chiama `router.resetConfig(this.configureRoutes(router.config))` — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L48-L52. `resetConfig` valida (dev mode), normalizza e reimposta `navigated = false` — angular:packages/router/src/router.ts:L389-L393. `configureRoutes` scende anche nei `children` — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L59-L68.
6. **Decisione per route.** Il nome è `route.data.cxRoute`; le route senza nome passano invariate — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L80, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L149-L151. Poi, in quest'ordine — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L83-L106:
   - `disabled` → `matcher = getFalsy()` (non combacia mai);
   - `matchers` → matcher combinato da `resolveUrlMatchers` (le funzioni passano così come sono, i token vengono risolti con `injector.get(token)(route)`) — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L118-L143;
   - esattamente un path → `path` nativo di Angular (quindi `defaultUrlMatcher`);
   - altrimenti → `matcher` da `getFromPaths(paths || [])`. In ogni ramo `path` o `matcher` viene cancellato, per non violare la regola "path e matcher insieme" di §3.1.
7. **Validazione morbida.** In dev mode `validateRouteConfig` considera valide la config `null`, `paths: null` o `matchers` definiti, e avvisa se config o `paths` sono `undefined` — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L153-L179.

3.5 I matcher di Spartacus

`UrlMatcherService` è una piccola fabbrica di `UrlMatcher`:

- `getFalsy()` → sempre `null` — core-libs/core/src/routing/services/url-matcher.service.ts:L24-L28.
- `getFromPaths(paths)` → un matcher per path, combinati; in dev mode aggiunge `_paths` — core-libs/core/src/routing/services/url-matcher.service.ts:L33-L40.
- `getCombined(matchers)` → OR in cortocircuito: il primo risultato non nullo vince — core-libs/core/src/routing/services/url-matcher.service.ts:L45-L63.
- `getFromPath(path)` riproduce `defaultUrlMatcher` di Angular, ma legge il path dall'argomento e gestisce internamente il path vuoto; il commento di licenza punta a una vecchia revisione di `shared.ts` — core-libs/core/src/routing/services/url-matcher.service.ts:L65-L88. Rispetta `pathMatch: 'full'` e tratta `:param` come parametro — core-libs/core/src/routing/services/url-matcher.service.ts:L90-L121; il path vuoto replica la regola di `match()` — core-libs/core/src/routing/services/url-matcher.service.ts:L128-L142.
- `getOpposite(matcher)` accetta ciò che l'originale rifiuta, consumando tutti i segmenti — core-libs/core/src/routing/services/url-matcher.service.ts:L162-L176.
- `getFromGlob(patterns)` ricostruisce `/a/b` e lo valida con `GlobService` — core-libs/core/src/routing/services/url-matcher.service.ts:L181-L197; un URL è valido se almeno un pattern positivo e nessun pattern `!` corrisponde — core-libs/core/src/util/glob.service.ts:L17-L32.

Le factory di matcher (`UrlMatcherFactory = (route: Route) => UrlMatcher` — core-libs/core/src/routing/url-matcher/url-matcher-factory.ts:L9-L9) si registrano come `InjectionToken`. `DEFAULT_URL_MATCHER` crea un matcher dai `paths` della route nominata da `data.cxRoute` — core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L13-L45.

Il matcher più interessante è `getSuffixUrlMatcher({ marker, paramName })`: cerca l'**ultima** occorrenza del marker, prende il segmento successivo come parametro e numera i precedenti `param0`, `param1`…, consumando fino al parametro — core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L18-L48. Su di esso sono costruiti:

- `PRODUCT_DETAILS_URL_MATCHER` = default matcher + suffisso `**/p/:productCode` — core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L16-L50, impostato come `routing.routes.product.matchers` di default — core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L14-L36;
- `PRODUCT_LISTING_URL_MATCHER` = default matcher + suffisso `**/c/:categoryCode`; la route `category` è dichiarata per ultima per non oscurare `brand` — core-libs/storefront/cms-pages/product-listing-page/product-listing-url-matcher.ts:L20-L27, core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L30-L38.

3.6 Route CMS: catch-all e route dinamiche

- **Catch-all.** `CmsRouteModule` registra un `APP_INITIALIZER` con `deps: [Injector]` — core-libs/storefront/cms-structure/routing/cms-route/cms-route.module.ts:L10-L20 — che fa `router.config.push` della route `**` (mutazione diretta, senza `resetConfig`) — core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25.
- **Route dinamiche con `childRoutes`.** Un componente CMS può dichiarare `childRoutes` (array di `Route` o `{ parent, children }`) e `guards` nella sua config — core-libs/core/src/cms/config/cms-config.ts:L48-L57, core-libs/core/src/cms/config/cms-config.ts:L74-L77. Durante il guard, `CmsRoutesImplService.handleCmsRoutesInGuard` ritorna `true` se la route CMS per la label esiste già; altrimenti, se i componenti della pagina hanno `childRoutes`, aggiorna la config e ri-naviga allo stesso URL ritornando `false` — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L91. `updateRouting` (solo content page con label `/…`) antepone con `resetConfig` una route `path = label senza '/'`, `PageLayoutComponent`, `children` e `data.cxCmsRouteContext` — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L93-L124. I guard dei `children` vengono avvolti in un `CanActivateFn` che li risolve tramite `UnifiedInjector` (anche se forniti solo da un modulo lazy) — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L126-L182.
- **Pulizia.** `RouterEffects.clearCmsRoutes$` rimuove le route con `data.cxCmsRouteContext` su cambio lingua, login o logout — core-libs/core/src/routing/store/effects/router.effect.ts:L19-L37.

3.7 `CmsPageGuard` e la composizione dei guard

1. `CmsPageGuard` è un guard di **classe** (`@Injectable({ providedIn: 'root' })`, senza `implements CanActivate`) con proprietà statica `guardName = 'CmsPageGuard'` — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L23-L27. Angular lo istanzia dall'injector perché ha un metodo `canActivate` (§3.3, passo 7).
2. `canActivate` esegue prima i guard **globali** di `BeforeCmsPageGuardService`; solo se il risultato è `true` attende il `nextPageContext` dallo store, carica la pagina CMS con `getPage(ctx, shouldReload)` e delega a `canActivatePage` o `canActivateNotFoundPage` — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84. `shouldReload()` è `true` salvo `loadStrategy === ONCE` — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L89-L91.
3. I guard globali vengono dal token multi `BEFORE_CMS_PAGE_GUARD` (default `[]`), che il JSDoc riserva ai guard globali — core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L10-L24. Lo storefront vi registra `ProtectedRoutesGuard` e `FederatedLoginGuard` — core-libs/storefront/cms-structure/routing/routing.module.ts:L27-L36; punchout e OPPS ne aggiungono altri — integration-

libs/punchout/root/guards/punchout-navigation.module.ts:L15-L15, integration-libs/opps/root/login-required/opps-login-required.module.ts:L14-L14.

4. `BeforeCmsPageGuardService` delega a `GuardsComposer.canActivate` — core-libs/storefront/cms-structure/guards/before-cms-page-guard.service.ts:L17-L32 — che invoca subito il metodo `canActivate()` di ogni guard (`guards.map(...)`) — core-libs/storefront/cms-structure/services/guards-composer.ts:L44-L48 — ma si sottoscrive agli Observable risultanti **in sequenza** (`concat`) e si ferma al primo non-`true`; un oggetto senza `canActivate` causa `Error('Invalid CanActivate guard')` — core-libs/storefront/cms-structure/services/guards-composer.ts:L59-L86. `GuardsComposer` definisce il proprio tipo `CanActivate = { canActivate: CanActivateFn }` come sostituto di quello deprecato di Angular — core-libs/storefront/cms-structure/services/guards-composer.ts:L16-L19.
5. `ProtectedRoutesGuard` mappa `route.url` a stringhe (root → `['']`) e, se l'URL è protetto, delega ad `AuthGuard` — core-libs/core/src/routing/protected-routes/protected-routes.guard.ts:L23-L33 — che salva l'URL corrente e ritorna un `UrlTree` verso `login` — core-libs/core/src/auth/user-auth/guards/auth.guard.ts:L30-L42. Solo `protected === false` (non `undefined`) rende pubblica una route — core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L82.
6. `canActivatePage`: tipi di componenti → `determineMappings` → guard dei componenti (`CmsGuardsService`) → chunk i18n → `handleCmsRoutesInGuard` se la route non è già CMS-driven — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L98. I guard dei componenti (`cmsComponents[].guards`) sono risolti via `UnifiedInjector` e composti con `GuardsComposer` — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48.
7. `canActivateNotFoundPage` carica la pagina `notFound`, registra il fallimento sulla pagina richiesta, cambia il `nextPageContext` e applica `canActivatePage`; se manca la route `notFound` ritorna `false` — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L107-L143. L'URL nel browser resta quello richiesto.

3.8 Il router in NgRx: quando arriva `ROUTER_NAVIGATION`

1. `RoutingModule` (core) importa lo store feature `router`, gli effetti e `StoreRouterConnectingModule.forRoot({ routerState: RouterState.Minimal, stateKey: 'router' })` — core-libs/core/src/routing/routing.module.ts:L73-L82. `RoutingModule.forRoot()` fornisce `RouterStateSerializer = CustomSerializer` — core-libs/core/src/routing/routing.module.ts:L84-L92.
2. `StoreRouterConnectingModule.forRoot` usa `provideRouterStore`, che fornisce a sua volta un `RouterStateSerializer` (il serializer minimale di NgRx, MinimalRouterStateSerializer, se non specificato) e istanzia il servizio di connessione con un environment initializer — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L351-L370, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L415-L420. Vince `CustomSerializer`: in `walkProviderTree` i provider degli import vengono visitati prima di quelli del `ModuleWithProviders` — angular:packages/core/src/di/provider_collection.ts:L293-L320 — e un provider non-multi sovrascrive il precedente — angular:packages/core/src/di/r3_injector.ts:L497-L521.
3. La config NgRx di default ha `navigationActionTiming: PreActivation` — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L102-L109 — e Spartacus non la cambia.
4. Il servizio si sottoscrive a `router.events`: su `NavigationStart` dispatcha `'@ngrx/router-store/request'`; su `RoutesRecognized`, con `PreActivation`, dispatcha **subito** `ROUTER_NAVIGATION`; su `NavigationCancel`/`NavigationError` dispatcha `ROUTER_CANCEL`/`ROUTER_ERROR`; su `NavigationEnd` dispatcha `ROUTER_NAVIGATED` — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L217-L256. Il JSDoc lo dichiara: "before any guards or resolvers run, the router will dispatch a ROUTER_NAVIGATION action" — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L373-L377.

5. ROUTER_NAVIGATION porta routerState = serializzazione dello snapshot **target** (RoutesRecognized.state); ROUTER_NAVIGATED serializza invece router.routerState.snapshot, già attivato — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L260-L282. Con routerState non Full, payload.event è ridotto a { id, url, urlAfterRedirects } — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L283-L305.
6. Poiché RoutesRecognized è emesso prima del blocco che emette GuardsCheckStart ed esegue checkGuards — angular:packages/router/src/navigation_transition.ts:L564-L573, angular:packages/router/src/navigation_transition.ts:L630-L649 — il nextState dello store Spartacus è già popolato quando CmsPageGuard si mette in ascolto. Lo State di NgRx applica le azioni via observeOn(queueScheduler) + scan — deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L487-L496; in ogni caso CmsPageGuard attende con filter(isNotUndefined) — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L58-L60.
7. Il reducer Spartacus salva il serializzato in nextState su ROUTER_NAVIGATION, lo azzera su ROUTER_ERROR/ROUTER_CANCEL e su ROUTER_NAVIGATED lo promuove a state preservando il context eventualmente cambiato durante la navigazione — core-libs/core/src/routing/store/reducers/router.reducer.ts:L47-L95. ChangeNextPageContext modifica il context del nextState solo se c'è una navigazione in corso — core-libs/core/src/routing/store/reducers/router.reducer.ts:L68-L75.
8. CustomSerializer. Scende lungo firstChild fino alla foglia, raccogliendo data.cxRoute e data.cxCmsRouteContext da qualsiasi antenato — core-libs/core/src/routing/store/reducers/router.reducer.ts:L112-L133. cmsRequired diventa true se c'è un contesto CMS o se un guard in canActivate è una funzione con guardName === 'CmsPageGuard' — core-libs/core/src/routing/store/reducers/router.reducer.ts:L135-L149. Il PageContext deriva da: preview SmartEdit; productCode → PRODUCT_PAGE; categoryCode/brandCode → CATEGORY_PAGE; data.pageLabel → CONTENT_PAGE — core-libs/core/src/routing/store/reducers/router.reducer.ts:L175-L196; in mancanza, CONTENT_PAGE con id '/' + segmenti della foglia o HOME_PAGE_CONTEXT per l'URL vuoto — core-libs/core/src/routing/store/reducers/router.reducer.ts:L198-L217.

3.9 Bootstrap: navigazione iniziale bloccante e trattenuta

1. AppRoutingModule → RouterModule.forRoot([], { anchorScrolling: 'enabled', initialNavigation: 'enabledBlocking' }) — core-libs/storefront/router/app-routing.module.ts:L13-L19. La demo app lo importa in AppModule prima di SpartacusModule — projects/storefrontapp/src/app/app.module.ts:L15-L22; lo schematic add-spartacus rimuove l'eventuale AppRoutingModule locale generato da ng new — core-libs/schematics/src/add-spartacus/update-app-module.ts:L70-L75.
2. initialNavigation: 'enabledBlocking' → withEnabledBlockingInitialNavigation() — angular:packages/router/src/router_module.ts:L237-L244 — che registra un app initializer: attende la promise di LOCATION_INITIALIZED, lancia router.initialNavigation() e sblocca il bootstrap in afterPreactivation (o a fine/annullamento della navigazione) — angular:packages/router/src/provide_router.ts:L441-L475.
3. ApplicationInitStatus.runInitializers invoca **sincronamente** tutti gli initializer in ordine e attende solo quelli asincroni — angular:packages/core/src/application/application_init.ts:L239-L275. Così ConfigurableRoutesService.init(), addCmsRoute, ExternalRoutesService.addRoutes() e OnNavigateService.initializeWithConfig() sono già stati eseguiti quando parte la navigazione iniziale.
4. Spartacus fornisce LOCATION_INITIALIZED con una factory che invoca tutti i LOCATION_INITIALIZED_MULTI e ritorna Promise.all — core-libs/core/src/routing/routing.module.ts:L54-L71, core-libs/core/src/routing/routing.module.ts:L102-L105. Tra i contributori: ConfigInitializerModule (config stabile) — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64 — e UserAuthModule — core-libs/core/src/auth/user-auth/user-auth.module.ts:L119-L120. Il JSDoc del to-

ken vieta di usare il Router nelle implementazioni — core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L22-L38.

5. **Ordine degli initializer di routing.** Lo storefront `RoutingModule` importa `[CoreRoutingModule.forRoot(), CmsRouteModule]` — core-libs/storefront/cms-structure/routing/routing.module.ts:L18-L20. I provider dei `ModuleWithProviders` importati vengono visitati dopo i moduli semplici — angular:packages/core/src/di/provider_collection.ts:L293-L320 — e `provideAppInitializer` è un provider multi di `APP_INITIALIZER` — angular:packages/core/src/application/application_init.ts:L192-L202: quindi `addCmsRoute` è registrato **prima** di `ConfigurableRoutesService.init()`. Il risultato è comunque corretto: la route `**` non ha `data.cxRoute` e `configureRoutes` la lascia invariata in coda — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L80, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L108-L108.

3.10 `UrlSerializer`, URL semantici e scroll

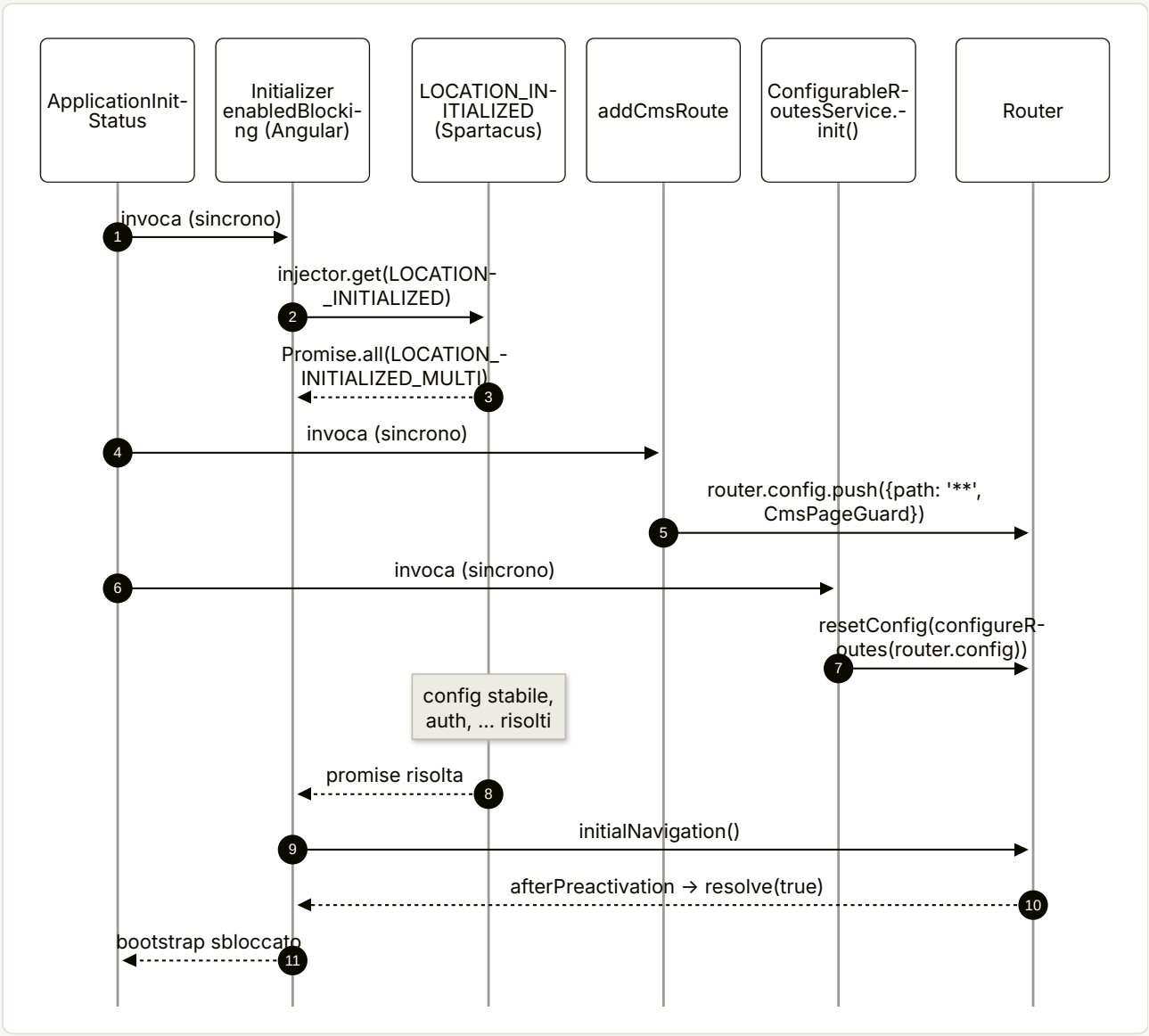
- **Serializer.** `UrlSerializer` è astratto, `providedIn: 'root'`, con factory `DefaultUrlSerializer` — angular:packages/router/src/url_tree.ts:L420-L427 — ma `ROUTER_PROVIDERS` lo ri-registra, con un TODO che avverte che "provider order matters" — angular:packages/router/src/router_module.ts:L65-L81. `SiteContextModule.forRoot()` lo sostituisce con `SiteContextUrlSerializer` (`useExisting`) — core-libs/core/src/site-context/providers/site-context-params-providers.ts:L14-L18 — e `BaseCoreModule` lo importa con il commento "should be imported after RouterModule.forRoot, because it overwrites UrlSerializer" — core-libs/core/src/base-core.module.ts:L38-L38. In `parse` rimuove i segmenti prefisso di site context e li salva in `urlTree.siteContext` — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L31-L71; in `serialize` li riantepone — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L126-L159. Il router usa il serializer iniettato per `parseUrl`/`serializeUrl` — angular:packages/router/src/router.ts:L588-L594 — ma `UrlTree.toString()` usa sempre un `DefaultUrlSerializer` statico — angular:packages/router/src/url_tree.ts:L289-L291.
- **URL semantici.** `RoutingService.go(commands)` traduce i comandi con `SemanticPathService.transform` e chiama `router.navigate` — core-libs/core/src/routing/facade/routing.service.ts:L91-L94. `transform` espande i comandi `{ cxRoute, params }` scegliendo il primo path i cui parametri sono tutti forniti (con `paramsMapping`) — core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L41-L68, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L135-L154. La pipe `cxUrl` delega a `transform` — core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts:L11-L17.
- **Scroll.** `RouterScroller` ha di default `scrollPositionRestoration` e `anchorScrolling` a `'disabled'` e, con hydration, sopprime gli eventi `Scroll` finché l'app non è stabile — angular:packages/router/src/router_scroller.ts:L48-L72, angular:packages/router/src/router_scroller.ts:L132-L132. Su `NavigationEnd` schedula uno `Scroll` con la posizione salvata solo per `popstate` — angular:packages/router/src/router_scroller.ts:L85-L104, angular:packages/router/src/router_scroller.ts:L149-L157. `OnNavigateService` imposta `history.scrollRestoration = 'manual'`, ascolta gli `Scroll` e, salvo filtri, scrolla in cima e mette il focus sull'host — core-libs/storefront/router/on-navigate.service.ts:L80-L90, core-libs/storefront/router/on-navigate.service.ts:L91-L116.

---

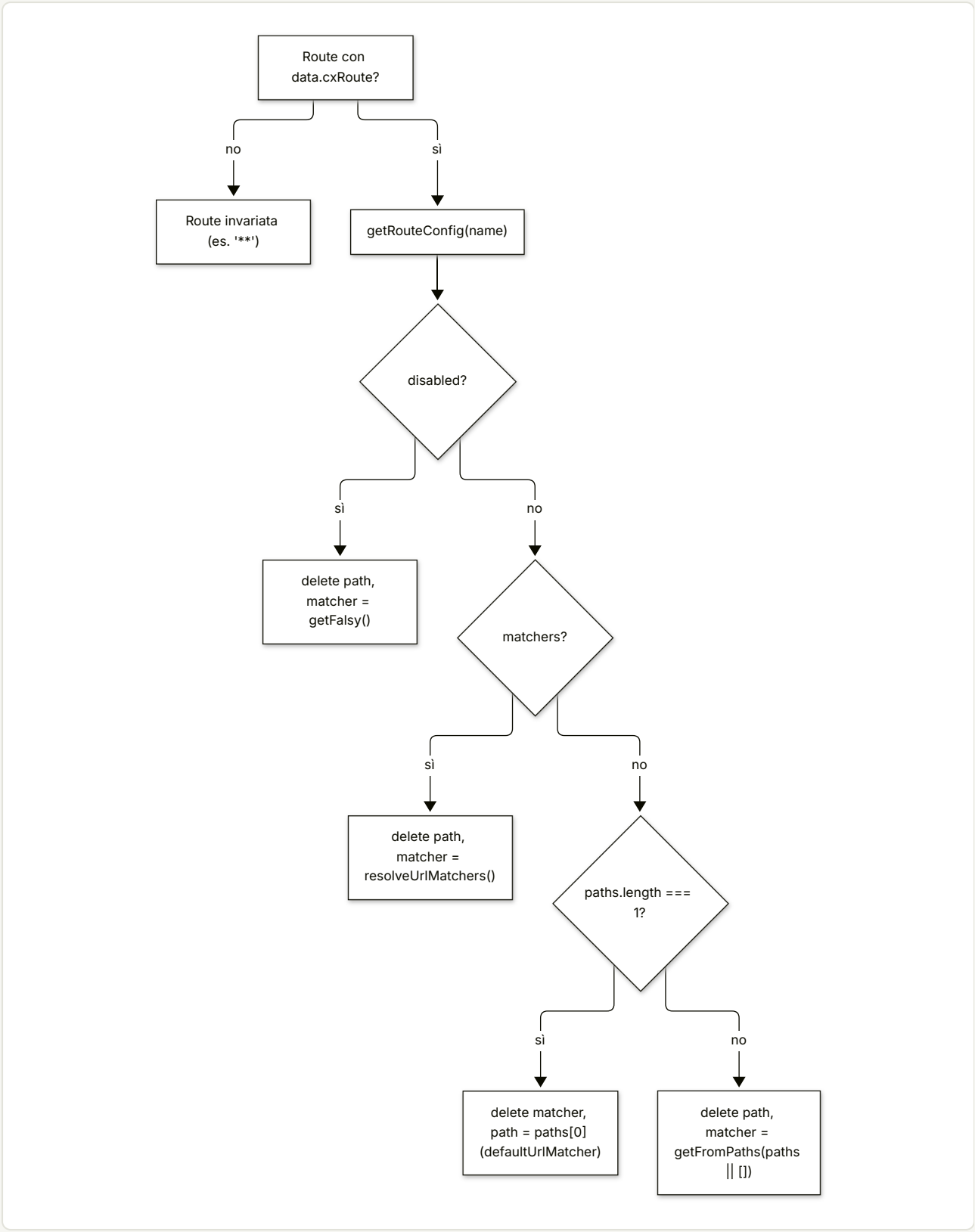
4. Diagrammi

4.1 Bootstrap: dalla config Spartacus alla navigazione iniziale

Ogni passo corrisponde a §3.4 e §3.9.


ApplicationInit-Status
Initializer enabledBlocki-ng (Angular)
LOCATION_IN-ITIALIZED (Spartacus)
addCmsRoute
ConfigurableR-outesService.-init()
Router
1
invoca (sincrono)
2
injector.get(LOCATION-_INITIALIZED)
3
Promise.all(LOCATION_-INITIALIZED_MULTI)
4
invoca (sincrono)
5
router.config.push({path: '**', CmsPageGuard})
6
invoca (sincrono)
7
resetConfig(configureR-outes(router.config))
config stabile, auth, ... risolti
8
promise risolta
9
initialNavigation()
10
afterPreactivation → resolve(true)
11
bootstrap sbloccato

4.2 Decisione di `configureRoute`



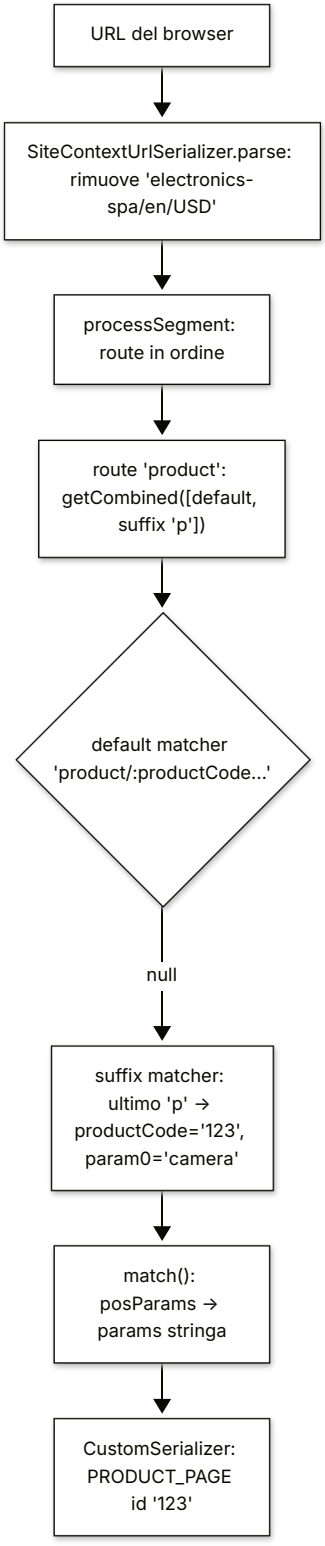
4.3 Una navigazione verso una pagina CMS

Combina §3.3, §3.7 e §3.8. La freccia "dispatch ROUTER_NAVIGATION" avviene durante la gestione sincrona di `RoutesRecognized`, prima di `GuardsCheckStart`.

■ **TAVOLA 05.1** diagramma a tutta pagina, in fondo al capitolo

4.4 Matching di `/electronics-spa/en/USD/camera/p/123`

Combina §3.2, §3.5 e §3.10.



5. Estratti di codice reale

5.1 AppRoutingModule

core-libs/storefront/router/app-routing.module.ts:L13-L37

```ts
@NgModule({
  imports: [
    RouterModule.forRoot([], {
      anchorScrolling: 'enabled',
      initialNavigation: 'enabledBlocking',
    }),
  ],
  providers: [
    provideDefaultConfig(defaultOnNavigateConfig),
    {
      provide: APP_INITIALIZER,
      useFactory: onNavigateFactory,
      deps: [OnNavigateService],
      multi: true,
    },
  ],
})
export class AppRoutingModule {}

export function onNavigateFactory(
  onNavigateService: OnNavigateService
): () => void {
  const isReady = () => onNavigateService.initializeWithConfig();
  return isReady;
}
```

5.2 `match()` di Angular

angular:packages/router/src/utils/config_matching.ts:L82-L112

```ts
export function match(
  segmentGroup: UrlSegmentGroup,
  route: Route,
  segments: UrlSegment[],
): MatchResult {
  if (route.path === '') {
    if (route.pathMatch === 'full' && (segmentGroup.hasChildren() || segments.length > 0)) {
      return {...noMatch};
    }

    return {
```

```
      matched: true,
      consumedSegments: [],
      remainingSegments: segments,
      parameters: {},
      positionalParamSegments: {},
    };
  }

  const matcher = route.matcher || defaultUrlMatcher;
  const res = matcher(segments, segmentGroup, route);
  if (!res) return {...noMatch};

  const posParams: {[n: string]: string} = {};
  Object.entries(res.posParams ?? {}).forEach(([k, v]) => {
    posParams[k] = v.path;
  });
  const parameters =
    res.consumed.length > 0
      ? {...posParams, ...res.consumed[res.consumed.length - 1].parameters}
      : posParams;
```

5.3 `runCanActivate`: classe o funzione

angular:packages/router/src/operators/check_guards.ts:L149-L172

```ts
function runCanActivate(
  futureRSS: RouterStateSnapshot,
  futureARS: ActivatedRouteSnapshot,
): Observable<GuardResult> {
  const canActivate = futureARS.routeConfig ? futureARS.routeConfig.canActivate : null;
  if (!canActivate || canActivate.length === 0) return of(true);

  const canActivateObservables = canActivate.map((canActivate) => {
    return defer(() => {
      const closestInjector = futureARS._environmentInjector;
      const guard = getTokenOrFunctionIdentity<CanActivate>(
        canActivate as ProviderToken<CanActivate>,
        closestInjector,
      );
      const guardVal = isCanActivate(guard)
        ? guard.canActivate(futureARS, futureRSS)
        : runInInjectionContext(closestInjector, () =>
            (guard as CanActivateFn)(futureARS, futureRSS),
          );
      return wrapIntoObservable(guardVal).pipe(first());
    });
```

```
  });
  return of(canActivateObservables).pipe(prioritizedGuardValue());
}
```

5.4 configureRoute

core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L109

```ts
protected configureRoute(route: Route): Route {
  const routeName = this.getRouteName(route);
  if (routeName) {
    const routeConfig = this.routingConfigService.getRouteConfig(routeName);
    this.validateRouteConfig(routeConfig, routeName, route);

    if (routeConfig?.disabled) {
      delete route.path;
      return {
        ...route,
        matcher: this.urlMatcherService.getFalsy(),
      };
    } else if (routeConfig?.matchers) {
      delete route.path;
      return {
        ...route,
        matcher: this.resolveUrlMatchers(route, routeConfig?.matchers),
      };
    } else if (routeConfig?.paths?.length === 1) {
      delete route.matcher;
      return { ...route, path: routeConfig?.paths[0] };
    } else {
      delete route.path;
      return {
        ...route,
        matcher: this.urlMatcherService.getFromPaths(
          routeConfig?.paths || []
        ),
      };
    }
  }
  return route; // if route doesn't have a name, just pass the original route
}
```

5.5 getSuffixUrlMatcher

core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L27-L48

```ts
  precedingParamName = precedingParamName || 'param';
  const matcher = function suffixUrlMatcher(
    segments: UrlSegment[]
  ): UrlMatchResult | null {
    const markerIndex = findLastIndex(segments, ({ path }) => path === marker);
    const isMarkerLastSegment = markerIndex === segments.length - 1;

    if (markerIndex === -1 || isMarkerLastSegment) {
      return null;
    }

    const paramIndex = markerIndex + 1;
    const posParams: { [name: string]: UrlSegment } = {
      [paramName]: segments[paramIndex],
    };

    for (let i = 0; i < markerIndex; i++) {
      posParams[`${precedingParamName}${i}`] = segments[i];
    }

    return { consumed: segments.slice(0, paramIndex + 1), posParams };
  };
```

5.6 La catch-all CMS

core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25

```ts
const cmsRoute: Route = {
  path: '**',
  canActivate: [CmsPageGuard],
  component: PageLayoutComponent,
};

export function addCmsRoute(injector: Injector): () => void {
  const result = () => {
    // Router could not be injected via `deps[]` due to a cyclic dependency with APP_INITIALIZER:
    const router = injector.get(Router);
    router.config.push(cmsRoute);
  };
  return result;
}
```

5.7 `CmsPageGuard.canActivate`

core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84

```ts
canActivate(
  route: CmsActivatedRouteSnapshot,
  state: RouterStateSnapshot
): Observable<GuardResult> {
  return this.beforeCmsPageGuardService.canActivate(route, state).pipe(
    switchMap((canActivate) =>
      canActivate === true
        ? this.routingService.getNextPageContext().pipe(
            filter(isNotUndefined),
            take(1),
            switchMap((pageContext) =>
              this.cmsService.getPage(pageContext, this.shouldReload()).pipe(
                first(),
                switchMap((pageData) =>
                  pageData
                    ? this.service.canActivatePage(
                        pageContext,
                        pageData,
                        route,
                        state
                      )
                    : this.service.canActivateNotFoundPage(
                        pageContext,
                        route,
                        state
                      )
                )
              )
            )
          )
        : of(canActivate)
    )
  );
}
```

5.8 `GuardsComposer`: sottoscrizione in serie

core-libs/storefront/cms-structure/services/guards-composer.ts:L59-L86

```ts
protected canActivateGuard(
  guard: CanActivate,
  route: CmsActivatedRouteSnapshot,
  state: RouterStateSnapshot
): Observable<GuardResult> {
  if (this.isCanActivate(guard)) {
    return wrapIntoObservable(guard.canActivate(route, state)).pipe(first());
  } else {
    throw new Error('Invalid CanActivate guard');
```

```
  }
}

// ...
protected composeCanActivateObservables(
  canActivateObservables: CanActivateObservable[]
): CanActivateObservable {
  return concat(...canActivateObservables).pipe(
    skipWhile((canActivate: GuardResult) => canActivate === true),
    endWith(true),
    first()
  );
}
```

5.9 NgRx: da eventi del router ad azioni

deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L217-L256

JS

```
  setUpRouterEventsListener() {
      const dispatchNavLate = this.config.navigationActionTiming ===
          NavigationActionTiming.PostActivation;
      let routesRecognized;
      this.router.events
          .pipe(withLatestFrom(this.store))
          .subscribe(([event, storeState]) => {
          this.lastEvent = event;
          if (event instanceof NavigationStart) {
              this.routerState = this.serializer.serialize(this.router.routerState.snapshot);
              if (this.trigger !== RouterTrigger.STORE) {
                  this.storeState = storeState;
                  this.dispatchRouterRequest(event);
              }
          }
          else if (event instanceof RoutesRecognized) {
              routesRecognized = event;
              if (!dispatchNavLate && this.trigger !== RouterTrigger.STORE) {
                  this.dispatchRouterNavigation(event);
              }
          }
          // ...
          else if (event instanceof NavigationEnd) {
              if (this.trigger !== RouterTrigger.STORE) {
                  if (dispatchNavLate) {
                      this.dispatchRouterNavigation(routesRecognized);
                  }
                  this.dispatchRouterNavigated(event);
              }
              this.reset();
```

```
          }
        });
    }
```

5.10 Reducer `router` di Spartacus

core-libs/core/src/routing/store/reducers/router.reducer.ts:L52-L90

```ts
    case fromNgrxRouter.ROUTER_NAVIGATION: {
      return {
        ...state,
        nextState: action.payload.routerState,
        navigationId: action.payload.event.id,
      };
    }

    case fromNgrxRouter.ROUTER_ERROR:
    case fromNgrxRouter.ROUTER_CANCEL: {
      return {
        ...state,
        nextState: undefined,
      };
    }
    // ...
    case fromNgrxRouter.ROUTER_NAVIGATED: {
      return {
        state: {
          ...action.payload.routerState,
          context:
            // we want to preserve already resolved context,
            // in case it was changed while navigating
            state.nextState?.context ?? action.payload.routerState.context,
        },
        navigationId: action.payload.event.id,
        nextState: undefined,
      };
    }
```

5.11 `LOCATION_INITIALIZED` di Spartacus

core-libs/core/src/routing/routing.module.ts:L66-L71

```ts
function locationInitializedFactory(): Promise<any> {
  const initializers =
    inject(LOCATION_INITIALIZED_MULTI, { optional: true }) ?? [];
```

```
  const promiseInitializers = initializers.map((initializer) => initializer());
  return Promise.all(promiseInitializers);
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes[name].paths	string[]	da defaultRoutesConfigFactory + config delle feature	core-libs/core/src/routing/configurable-routes/routes-config.ts:L22-L22; default core-libs/storefront/cms-structure/routing/default-routing-config.ts:L17-L89	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L104, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L26-L31
routing.routes[name].matchers	(UrlMatcher | InjectionToken<UrlMatcherFactory>)[]	— (PDP/PLP lo impostano)	core-libs/core/src/routing/configurable-routes/routes-config.ts:L31-L31; core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L27-L35	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L118-L143
routing.routes[name].paramsMapping	{ [param]: string }	es. product: { productCode: 'code' }	core-libs/core/src/routing/configurable-routes/routes-config.ts:L36-L36	core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L135-L154
routing.routes[name].disabled	boolean	—	core-libs/core/src/routing/configurable-routes/routes-config.ts:L41-L41	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L83-L88
routing.routes[name].protected	boolean	— (conta solo false)	core-libs/core/src/routing/configurable-routes/routes-config.ts:L47-L47	core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L82
routing.routes[name].authFlow	boolean	true per login/register/...	core-libs/core/src/routing/configurable-routes/routes-config.ts:L56-L56; core-libs/storefront/cms-structure/routing/default-routing-config.ts:L24-L59	letta da AuthFlowRoutesService — core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts:L23-L36 — usata via isAuthFlow da core-libs/core/src/auth/user-auth/guards/federated-login.guard.ts:L64-L69 e core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L113-L113
routing.protected	boolean	undefined → impostato da SecurePortalConfigInitializer	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L22-L49	core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L28-L48; core-libs/core/src/routing/routing.module.ts:L44-L52
routing.loadStrategy	RouteLoadStrategy ('once'/'always')	ALWAYS	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L11-L20	core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L45-L47; core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L89-L91

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.internal	string[] (glob, ! = esclusione)	—	core-libs/core/src/routing/external-routes/external-routes-config.ts:L10-L24	core-libs/core/src/routing/external-routes/external-routes.service.ts:L35-L70
enableResetViewOnNavigate.active	boolean	true	core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16	core-libs/storefront/router/on-navigate.service.ts:L80-L90
enableResetViewOnNavigate.ignoreQueryString	boolean	false	core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16	core-libs/storefront/router/on-navigate.service.ts:L91-L116
enableResetViewOnNavigate.ignoreRoutes	string[]	[]	core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16	core-libs/storefront/router/on-navigate.service.ts:L171-L179
enableResetViewOnNavigate.selectedHostElement	string (tag)	'body'	core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16	core-libs/storefront/router/on-navigate.service.ts:L45-L57
RouterModule.forRoot → anchorScrolling	'disabled' | 'enabled'	Angular 'disabled'; Spartacus 'enabled'	core-libs/storefront/router/app-routing.module.ts:L16-L16	angular:packages/router/src/router_scroller.ts:L106-L126; core-libs/storefront/router/on-navigate.service.ts:L138-L163
RouterModule.forRoot → initialNavigation	'enabledBlocking' | …	Spartacus 'enabledBlocking'	core-libs/storefront/router/app-routing.module.ts:L17-L17	angular:packages/router/src/router_module.ts:L237-L244
RouterModule.forRoot → scrollPositionRestoration	'disabled' | 'enabled' | 'top'	'disabled' (Spartacus non lo imposta)	angular:packages/router/src/router_scroller.ts:L48-L72	angular:packages/router/src/router_scroller.ts:L106-L126
cmsComponents[type].childRoutes	Route[] | CmsComponentChildRoutesConfig	—	core-libs/core/src/cms/config/cms-config.ts:L74-L74	core-libs/storefront/cms-structure/services/cms-components.service.ts:L205-L248
cmsComponents[type].guards	any[]	—	core-libs/core/src/cms/config/cms-config.ts:L77-L77	core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261
StoreRouterConnectingModule.forRoot → routerState / stateKey / navigationActionTiming	NgRx	Minimal / 'router' / PreActivation (non impostato)	core-libs/core/src/routing/routing.module.ts:L77-L80	deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L102-L109, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L283-L305

7. Commenti e note notevoli dal sorgente

- "Router could not be injected in constructor due to cyclic dependency with APP_INITIALIZER" — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L49-L49; variante analoga in core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L20-L20: "Router could not be injected via deps[] due to a cyclic dependency with APP_INITIALIZER". Il Router dipende da initializer che a loro volta vorrebbero il Router: per questo viene preso pigramente dall'Injector.
- "workaround for AOT compilation (see https://stackoverflow.com/a/51977115)" — core-libs/core/src/routing/routing.module.ts:L40-L40.
- JSDoc di LOCATION_INITIALIZED: "Only when the Promise in this injection token is resolved, then Angular will start the initial navigation in the Router." — core-libs/core/src/routing/routing.module.ts:L54-L65.

- "CAUTION: Don't use Router and don't perform any Router navigation in its implementations." — core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L30-L35.
- `/** since 2211.24 not used anymore, but called indirectly via {@link BeforeCmsPageGuardService} */` sulla dipendenza `ProtectedRoutesGuard` di `CmsPageGuard` — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L32-L33.
- "**CAUTION**: Suitable for global guards ONLY! For route-specific guards, configure them under `cmsComponents`" — core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L13-L20.
- "Replacement for the Angular's deprecated type `CanActivate`." — core-libs/storefront/cms-structure/services/guards-composer.ts:L16-L19.
- "This service should be exposed in public API only after the refactor planned in https://github.com/SAP/spartacus/issues/7070" — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L31; `CmsRoutesService` è il token pubblico per l'implementazione privata — core-libs/storefront/cms-structure/services/cms-routes.service.ts:L11-L23.
- Licenza MIT Google nel matcher copiato da Angular (`shared.ts#L121` di una vecchia revisione) — core-libs/core/src/routing/services/url-matcher.service.ts:L76-L83.
- "Note: Matchers will "match" a route, but do not contribute to the creation of the route, nor do they guard routes." — core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L33-L33.
- Ordine PLP: "The 'category' route may include a greedy suffix url matcher '**/c/:categoryCode' ... the 'category' is the last route in the sequence." — core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L31-L32.
- "should be imported after RouterModule.forRoot, because it overwrites UrlSerializer" — core-libs/core/src/base-core.module.ts:L38-L38.
- Commento in `OnNavigateService` su hydration e `pairwise`: il primo `Scroll` della navigazione iniziale è soppresso, quindi il precedente è tracciato a mano — core-libs/storefront/router/on-navigate.service.ts:L36-L43.
- Nel reducer: "we want to preserve already resolved context, in case it was changed while navigating" — core-libs/core/src/routing/store/reducers/router.reducer.ts:L82-L84.
- Angular, TODO sui `ROUTER_PROVIDERS` mantenuti "to avoid a breaking change whereby the provider order matters" — angular:packages/router/src/router_module.ts:L65-L68.
- Angular, commento su `router.events`: "it's important to have the Router process the events *before* the event is pushed through the public observable" — angular:packages/router/src/router.ts:L245-L250.
- Angular, `@deprecated` su `DeprecatedGuard` — angular:packages/router/src/models.ts:L50-L64.
- NgRx, JSDoc: "before any guards or resolvers run, the router will dispatch a ROUTER_NAVIGATION action" — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L373-L377.

8. Extension point

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`routing.routes[name].paths`	Cambiare o aggiungere path a una route nominata con `provideConfig`	core-libs/core/src/routing/configurable-routes/routes-config.ts:L15-L57	51 — Ricette: CMS, outlet, route e layout

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`routing.routes[name].matchers` + `UrlMatcherFactory`	Matcher custom (funzione o token di factory)	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L118-L143; core-libs/core/src/routing/url-matcher/url-matcher-factory.ts:L9-L9	51
`DEFAULT_URL_MATCHER`, `PRODUCT_DETAILS_URL_MATCHER`, `PRODUCT_LISTING_URL_MATCHER`	Comporre i matcher esistenti in uno nuovo	core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L35-L45; core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L42-L50	51
`getSuffixUrlMatcher`	URL SEO-friendly `…/marker/:param`	core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L18-L55	51
`UrlMatcherService`	Sostituzione via DI o uso diretto (`getCombined`, `getOpposite`, `getFromGlob`)	core-libs/core/src/routing/services/url-matcher.service.ts:L17-L198	50 — Ricette: config, DI, interceptor, eventi
Route Angular aggiuntive	`RouterModule.forChild` con `data.cxRoute` (+ `CmsPageGuard` se la pagina è CMS)	feature-libs/cart/base/root/cart-base-root.module.ts:L62-L74	51
`BEFORE_CMS_PAGE_GUARD`	Guard globale prima della logica CMS (oggetto `{ canActivate }`)	core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L10-L24	51
`cmsComponents[type].guards` / `.childRoutes`	Guard per componente; route figlie della pagina CMS	core-libs/core/src/cms/config/cms-config.ts:L48-L77	51
`CmsPageGuard`, `CmsPageGuardService`, `BeforeCmsPageGuardService`, `GuardsComposer`	Sostituzione via DI (servizi `providedIn: 'root'`)	core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L23-L92; core-libs/storefront/cms-structure/services/guards-composer.ts:L30-L94	50, 54 — Upgrade-safe
`LOCATION_INITIALIZED_MULTI` + `provideLocationInitializer(Factory)`	Trattenere la navigazione iniziale fino a una promise	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L22-L62	50
`ExternalRoutesModule` + `routing.internal`	URL serviti da un altro sistema (full reload)	core-libs/core/src/routing/external-routes/external-routes.service.ts:L35-L70	51
`OnNavigateConfig` (`enableResetViewOnNavigate`)	Disattivare/filtrare il reset di scroll e focus	core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16	50
`UrlSerializer` (Angular)	Override via DI, in competizione con `SiteContextUrlSerializer`	core-libs/core/src/site-context/providers/site-context-params-providers.ts:L14-L18	54
`RouterStateSerializer` (NgRx)	Già occupato da `CustomSerializer` (non esportato)	core-libs/core/src/routing/routing.module.ts:L84-L92	54

Note di superficie pubblica (dal dossier): `CustomSerializer`, `reducer` e `RouterEffects` non sono esportati da `@spartacus/core` (le cartelle `store/reducers` ed `store/effects` sono escluse dal barrel di `core-libs/core/src/routing/index.ts`); `CmsRoutesImplService` non è esportato, `CmsRoutesService` sì. Il dettaglio delle ricette, con verifica dei simboli nei `public_api`, è nella Parte III.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `path: null` **è una mina.** Supera la validazione Angular (`=== void 0`) — angular:packages/router/src/utils/config.ts:L185-L185 — ma se `ConfigurableRoutesService` non la configurasse, `defaultUrlMatcher` farebbe `route.path!.split('/')` su `null` — angular:packages/router/src/shared.ts:L158-L158, feature-libs/cart/base/root/cart-base-root.module.ts:L64-L65. Le route con `path: null` funzionano solo se hanno `data.cxRoute`.
2. **Mutazione dell'oggetto originale.** `configureRoute` fa `delete route.path` / `delete route.matcher` sull'oggetto registrato in `ROUTES` prima dello spread — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L84-L97.
3. **Route nominata senza config = disabilitata in silenzio.** Finisce nel ramo `getFromPaths([])`, un matcher che non combacia mai, con warning solo in dev mode — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L98-L105, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L169-L176.
4. **Uno vs due path cambiano il matcher.** Con un solo path si usa `defaultUrlMatcher` di Angular, che in v21 supporta `**` in mezzo al path; con due o più si usa `getFromPath` di Spartacus, che confronta `**` letteralmente — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L97, core-libs/core/src/routing/services/url-matcher.service.ts:L90-L119, angular:packages/router/src/shared.ts:L183-L217.
5. **L'ordine conta.** `processSegment` prende la prima route che corrisponde — angular:packages/router/src/recognize.ts:L235-L267. Il suffix matcher di `category` (`**/c/:categoryCode`) catturerebbe `Brands/:brandName/c/:brandCode`, per questo `category` è dichiarata dopo `brand` — core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L23-L38. Le route anteposte con `resetConfig([nuova, ...config])` (route esterne, route CMS dinamiche) hanno precedenza su tutte — core-libs/core/src/routing/external-routes/external-routes.service.ts:L35-L70, core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L93-L124.
6. `resetConfig` **azzera** `navigated`**.** Dopo `clearCmsRoutes$` o `updateRouting`, una navigazione allo stesso URL non viene saltata (`urlTransition = !router.navigated || …`) — angular:packages/router/src/router.ts:L389-L393, angular:packages/router/src/navigation_transition.ts:L494-L513.
7. `false` **+** `navigateByUrl`**.** `handleCmsRoutesInGuard` avvia una nuova navigazione e poi ritorna `false`. La corrente viene cancellata con codice `SupersededByNewNavigation`, perché la nuova navigazione parte in modo sincrono prima che il guard restituisca `false`: `transitions.next(...)` fa scattare lo `switchMap` di `setupNavigations`, il cui `finalize` cancella la transizione in corso; il `false` non viene mai letto. Ne seguono `ROUTER_CANCEL` e l'azzeramento di `nextState`, e la nuova navigazione sostituisce la vecchia — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L83-L88, angular:packages/router/src/navigation_transition.ts:L419-L425, angular:packages/router/src/navigation_transition.ts:L445-L446, angular:packages/router/src/navigation_transition.ts:L847-L865, core-libs/core/src/routing/store/reducers/router.reducer.ts:L60-L66. Chi osserva `NavigationCancel` vedrà un annullamento "spurio".
8. **Serie vs parallelo.** `GuardsComposer` invoca subito il metodo `canActivate()` di ogni guard — core-libs/storefront/cms-structure/services/guards-composer.ts:L44-L48, core-libs/storefront/cms-structure/

services/guards-composer.ts:L64-L65 — ma si sottoscrive agli Observable risultanti in serie (`concat`) e si ferma al primo risultato non-`true` — core-libs/storefront/cms-structure/services/guards-composer.ts:L81-L85. Angular invece li avvia in parallelo e ne prioritizza solo il risultato — angular:packages/router/src/operators/prioritized_guard_value.ts:L18-L45. Stesso risultato, tempi ed effetti collaterali diversi: il lavoro differito (Observable cold) di un guard in `BEFORE_CMS_PAGE_GUARD` non parte finché i precedenti non hanno emesso `true`, mentre il codice sincrono del suo `canActivate()` (o una Promise già avviata) è già stato eseguito.

9. `cmsRequired` **riconosce** `CmsPageGuard` **per nome.** `CustomSerializer` cerca una funzione con `guardName === 'CmsPageGuard'` in `canActivate` — core-libs/core/src/routing/store/reducers/router.reducer.ts:L140-L146. Un guard funzionale che fa `inject(CmsPageGuard)` non rende la route `cmsRequired`. Una sottoclasse di `CmsPageGuard` eredita la proprietà statica `guardName` — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L27-L27 — e quindi viene riconosciuta, a meno che non la ridefinisca.
10. **Navigazioni innescate dallo store non passano dal reducer.** Se lo stato `router` cambia senza che l'abbia innescato il router, NgRx naviga verso `state.url` (`navigateIfNeeded`) e, per quella navigazione, non dispatcha né `'@ngrx/router-store/request'` né `ROUTER_NAVIGATION`/`ROUTER_NAVIGATED` (`trigger === STORE`) — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L198-L216, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L217-L256. Ne segue (dal reducer, §3.8) che in quel caso `nextState` non verrebbe popolato e `CmsPageGuard` attenderebbe un contesto che non arriva. Non scrivere nello slice `router` con reducer propri.
11. **Il contesto può sovrascrivere** `cxCmsRouteContext`**.** In `getPageContext`, se è presente un param `productCode`/`categoryCode`/`brandCode` o `data.pageLabel`, il contesto calcolato vince anche su quello della route CMS — core-libs/core/src/routing/store/reducers/router.reducer.ts:L173-L196.
12. `UrlTree.toString()` **non conosce il site context.** Usa un `DefaultUrlSerializer` statico — angular:packages/router/src/url_tree.ts:L289-L291 — mentre `router.serializeUrl` usa `SiteContextUrlSerializer` — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L126-L131. Per stringhe da mettere nel browser usare `router.serializeUrl` (come fa `RoutingService.getUrl` — core-libs/core/src/routing/facade/routing.service.ts:L101-L133).
13. **Override di** `UrlSerializer`**.** Vince l'ultimo provider non-multi visitato — angular:packages/core/src/di/r3_injector.ts:L497-L521. Un `UrlSerializer` custom fornito in `AppModule.providers` verrebbe visitato dopo gli import e quindi vincerebbe su `SiteContextUrlSerializer` (i provider del modulo seguono gli import — angular:packages/core/src/di/provider_collection.ts:L346-L354), perdendo il prefisso di site context se non estende quella classe.
14. **Doppio scroll all'ancora.** Con `anchorScrolling: 'enabled'` `RouterScroller` chiama `scrollToAnchor` — angular:packages/router/src/router_scroller.ts:L106-L126 — e `OnNavigateService` lo richiama dopo 100 ms — core-libs/storefront/router/on-navigate.service.ts:L138-L163.
15. **Nessuno** `Scroll` **alla prima navigazione con hydration.** `RouterScroller` non emette finché l'app non è stabile — angular:packages/router/src/router_scroller.ts:L132-L132 — motivo del `previousScroll` esplicito — core-libs/storefront/router/on-navigate.service.ts:L36-L43.
16. `getSuffixUrlMatcher` **non consuma la coda.** Usa l'ultima occorrenza del marker, richiede che non sia l'ultimo segmento e lascia non consumati i segmenti dopo il parametro (restano per eventuali figli o fanno fallire il match) — core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L31-L47.
17. **Matrix params finali diventano route params.** `match()` fonde i matrix params dell'ultimo segmento consumato: con `getOpposite`/`getFromGlob` (che consumano tutto) finiscono nei params — angular:packages/router/src/utils/config_matching.ts:L109-L112, core-libs/core/src/routing/services/url-matcher.service.ts:L168-L170.
18. `ExternalRoutesGuard` **in SSR.** Ritorna sempre `false` e fa il full reload solo nel browser: in SSR la navigazione è rifiutata senza redirect — core-libs/core/src/routing/external-routes/external-routes.guard.

ts:L22-L41.

19. **Path duplicati.** Se due route nominate hanno lo stesso path, `getRouteName` restituisce l'ultima; l'errore è loggato solo in dev mode — core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L63-L92.
20. **Un comando non traducibile manda alla home.** `SemanticPathService.transform` ritorna `['/']` se anche un solo comando semantico non è risolvibile — core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L41-L68.
21. `ProtectedRoutesService` **legge** `routing.protected` **nel costruttore** — core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L28-L48. DA VERIFICARE: se il servizio fosse istanziato prima che `SecurePortalConfigInitializer` imposti il flag, i path non protetti non verrebbero pre-calcolati. Dipende da quando il servizio viene creato rispetto ai config initializer, cosa non tracciata nel dossier.

---

10. Come fare debug

- **Tracciare la pipeline.** `RouterModule.forRoot` accetta `enableTracing`, che in dev mode attiva `withDebugTracing()` e logga in console tutti gli eventi interni — angular:packages/router/src/router_config.ts:L205-L210, angular:packages/router/src/router_module.ts:L136-L141 — ma `AppRoutingModule` non lo imposta e le sue opzioni sono fisse — core-libs/storefront/router/app-routing.module.ts:L13-L19. La via più semplice è sottoscriversi a `router.events` e loggare `NavigationStart`, `RoutesRecognized`, `GuardsCheckStart`, `GuardsCheckEnd`, `NavigationCancel` (campo `code`) e `NavigationEnd`: gli eventi pubblici sono quelli di `EventType` — angular:packages/router/src/events.ts:L33-L51. Una `NavigationCancel` con `code` `SupersededByNewNavigation`, subito seguita da una `NavigationStart` verso lo stesso URL, indica `handleCmsRoutesInGuard` (§9.7).
- **Ispezionare la config effettiva.** Dopo il bootstrap, `inject(Router).config` contiene le route già trasformate. In dev mode i matcher hanno proprietà di debug: `_paths` (`getFromPaths`), `_matchers` (`getCombined`), `_suffixRouteConfig` (`getSuffixUrlMatcher`) — core-libs/core/src/routing/services/url-matcher.service.ts:L33-L40, core-libs/core/src/routing/services/url-matcher.service.ts:L59-L61, core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L50-L52. Se una route ha `matcher: falsyUrlMatcher`, è `disabled` — core-libs/core/src/routing/services/url-matcher.service.ts:L24-L28. L'ultima route deve essere la catch-all `**`.
- **Warning in console (dev mode).** "No path was configured for the named route '...'!" — core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L26-L34; "Could not configure the named route '...'" — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L153-L179; path duplicati — core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L83-L90.
- **Store** `router` **con Redux DevTools.** Sequenza attesa: `@ngrx/router-store/request` → `@ngrx/router-store/navigation` (popola `nextState`) → eventuale `[Router] Change Next PageContext` (pagina not found) → `@ngrx/router-store/navigated` oppure `cancel`/`error` — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L217-L256, core-libs/core/src/routing/store/actions/router.action.ts:L10-L15. Controllare `nextState.context` (tipo e id della pagina) e `cmsRequired`.
- **Guard che non rispondono.** Se la navigazione resta appesa dopo `GuardsCheckStart`, mettere breakpoint in `CmsPageGuard.canActivate` — core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84 — e in `GuardsComposer.composeCanActivateObservables` — core-libs/storefront/cms-structure/services/guards-composer.ts:L78-L86: un guard in `BEFORE_CMS_PAGE_GUARD` che non emette blocca tutti i successivi.

- **Navigazione iniziale che non parte.** Con `enabledBlocking` il bootstrap attende `LOCATION_INITIALIZED`: un `LOCATION_INITIALIZED_MULTI` la cui promise non si risolve blocca l'app prima della prima navigazione — core-libs/core/src/routing/routing.module.ts:L66-L71, angular:packages/router/src/provide_router.ts:L441-L475.
- **Serializer.** Confrontare `router.serializeUrl(tree)` con `tree.toString()`: se differiscono per il prefisso di site context, il serializer attivo è `SiteContextUrlSerializer` (§3.10).

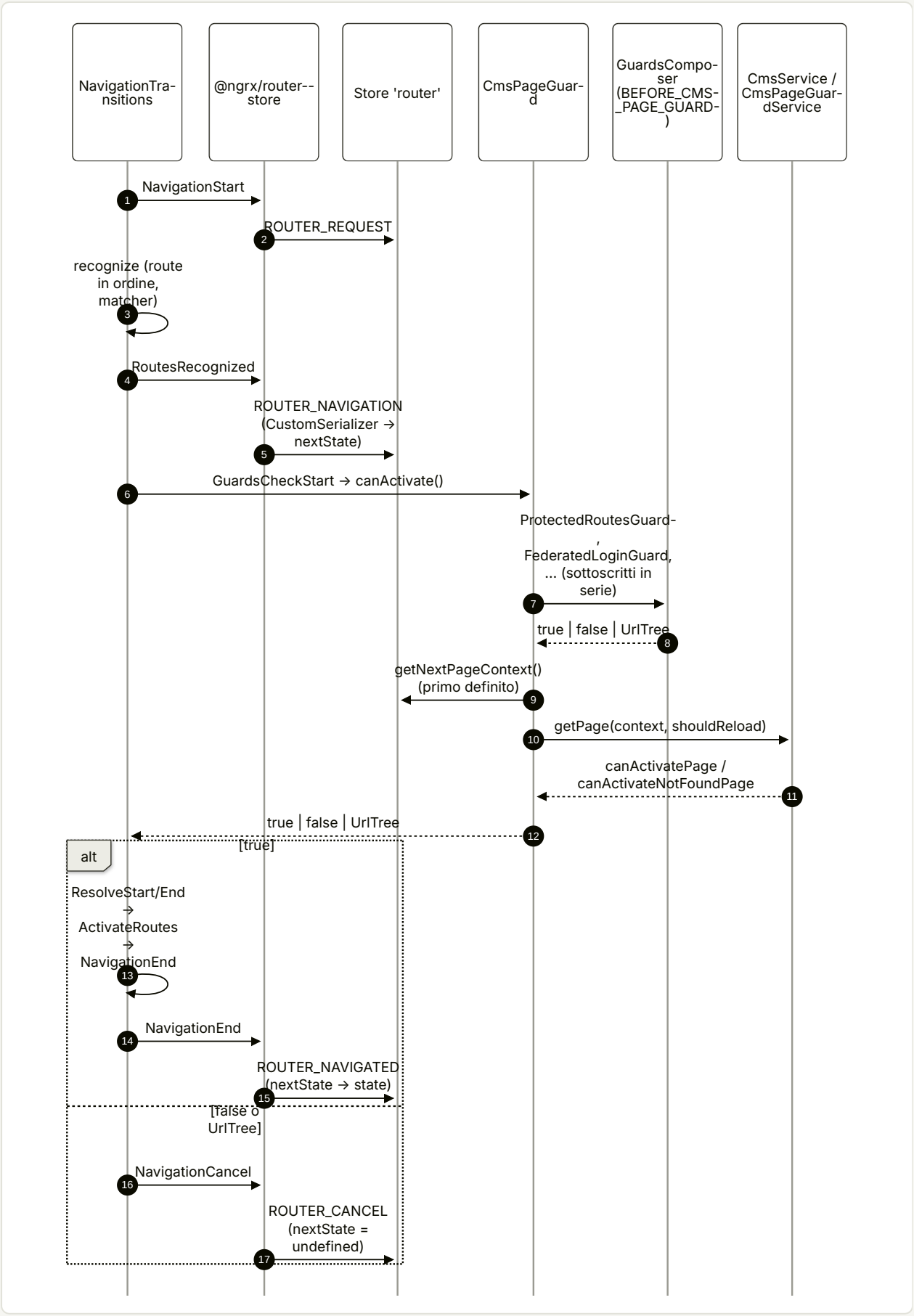
---

11. Domande di autoverifica

1. **Perché le route di feature possono avere** `path: null` **senza errori di validazione?** Perché `validateConfig` controlla `route.path === void 0 && route.matcher === void 0`, e `null` non è `undefined`; il path reale viene poi impostato da `ConfigurableRoutesService` — angular:packages/router/src/utils/config.ts:L185-L190, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L83-L106.
2. **Una route ha in config** `paths: ['a', 'b']`**. Viene usato** `path` **o** `matcher`**?** `matcher`, creato da `getFromPaths`; solo con esattamente un path viene usato `path` nativo — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L105.
3. **Quando viene dispatchata** `ROUTER_NAVIGATION` **rispetto ai guard?** Sull'evento `RoutesRecognized`, perché il timing di default è `PreActivation`; `RoutesRecognized` precede `GuardsCheckStart` e `checkGuards` — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L102-L109, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L232-L237, angular:packages/router/src/navigation_transition.ts:L564-L573, angular:packages/router/src/navigation_transition.ts:L630-L649.
4. **Perché vince** `CustomSerializer` **e non il serializer minimale di NgRx (MinimalRouterStateSerializer)?** Perché `StoreRouterConnectingModule.forRoot` è negli `imports` di `RoutingModule`, mentre `CustomSerializer` è nei provider di `RoutingModule.forRoot()`, visitati dopo gli import; l'ultimo provider non-multi vince — core-libs/core/src/routing/routing.module.ts:L73-L92, angular:packages/core/src/di/provider_collection.ts:L293-L320, angular:packages/core/src/di/r3_injector.ts:L497-L521.
5. **In che ordine girano i guard registrati in** `BEFORE_CMS_PAGE_GUARD`**, e cosa succede se il primo ritorna un** `UrlTree`**?** Tutti i `canActivate()` vengono invocati subito, ma gli Observable risultanti sono sottoscritti in serie (`concat`), fermandosi al primo risultato non-`true` (§9.8); l'`UrlTree` diventa il risultato di `CmsPageGuard`, e Angular lo trasforma in redirect (`NavigationCancel` + `RedirectRequest`) — core-libs/storefront/cms-structure/services/guards-composer.ts:L78-L86, core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L55-L57, angular:packages/router/src/navigation_transition.ts:L651-L671.
6. **Perché** `addCmsRoute` **può fare** `push` **invece di** `resetConfig` **e restare comunque l'ultima route?** Perché `ConfigurableRoutesService` trasforma la config con un `map` che lascia invariate le route senza `data.cxRoute`, mantenendone la posizione; le route anteposte (esterne, CMS dinamiche) vanno in testa — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L59-L68, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L108-L108, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L18-L25.
7. **Come fa Spartacus a ritardare la navigazione iniziale finché la configurazione non è stabile?** Fornendo `LOCATION_INITIALIZED` come `Promise.all` dei `LOCATION_INITIALIZED_MULTI`; l'initializer `enabledBlocking` di Angular attende quella promise prima di `router.initialNavigation()`, e `ConfigInitializerModule` vi contribuisce con `getStable()` — core-libs/core/src/routing/routing.module.ts:L66-L71, angular:packages/router/src/provide_router.ts:L441-L475, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64.

8. **Con l'URL** `/camera/p/123`**, quali params produce il matcher PDP?** Il default matcher fallisce, il suffix matcher trova l'ultimo `p` e produce `productCode = '123'` e `param0 = 'camera'`; `match()` li converte in stringhe — core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L31-L47, angular:packages/router/src/utils/config_matching.ts:L105-L112.
9. **Cosa succede nello store se** `CmsPageGuard` **ritorna** `false` **dopo aver aggiunto route** `childRoutes`**?** Angular cancella la navigazione, NgRx dispatcha `ROUTER_CANCEL` e il reducer azzera `nextState`; la nuova `navigateByUrl` ripopola `nextState` con la propria `ROUTER_NAVIGATION` — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L91, core-libs/core/src/routing/store/reducers/router.reducer.ts:L60-L66, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L217-L256.
10. **Perché** `OnNavigateService` **non usa** `pairwise` **sugli eventi** `Scroll`**?** Perché con hydration il primo `Scroll` della navigazione iniziale viene soppresso, e `pairwise` scarterebbe la prima navigazione client — core-libs/storefront/router/on-navigate.service.ts:L36-L43, angular:packages/router/src/router_scroller.ts:L132-L132.

TAVOLA 05.1 **4.3 Una navigazione verso una pagina CMS**



06

HttpClient e interceptor

La catena di interceptor tra l'adapter e il `fetch()`, su cui Spartacus appoggia auth, site context ed errori.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Angular @ `v21.2.23` (prefisso `angular:`), `angular-oauth2-oidc` 20.0.2 (prefisso `deps:`).

---

1. A cosa serve e perché esiste

Ogni byte che Spartacus scambia con il backend SAP Commerce (OCC), e anche ogni file di traduzioni caricato via HTTP, passa per l'`HttpClient` di Angular. Tra la chiamata `this.http.get(...)` di un adapter e il `fetch()` vero e proprio c'è una **catena di interceptor**: oggetti che vedono la richiesta all'andata e la risposta (o l'errore) al ritorno. Su questa catena Spartacus appoggia funzioni centrali:

- aggiunge `lang` e `curr` a ogni URL OCC (`SiteContextInterceptor`) — `core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L30-L68`;
- aggiunge l'header `Authorization` con il token utente e, su 401, rinnova il token e riprova (`AuthInterceptor`) — `core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L35-L83`;
- impone un timeout alle richieste in SSR (`HttpTimeoutInterceptor`, default 20 s lato server) — `core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15`;
- trasforma gli errori HTTP in global message (`HttpErrorInterceptor`) — `core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L26-L65` — e, in SSR, li inoltra all'`ErrorHandler` di Angular (`HttpErrorHandlerInterceptor`) — `core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L27-L41`.

Il capitolo esiste per tre motivi.

1. **Spartacus usa solo interceptor "di classe".** Tutte le 22 registrazioni di libreria sono provider multi su `HTTP_INTERCEPTORS`; nessuna usa `withInterceptors` o `HttpInterceptorFn` (vedi appendice A06). L'app deve quindi chiamare `provideHttpClient(withInterceptorsFromDi())`, come fa l'app demo — `projects/storefrontapp/src/app/app.config.ts:L26-L35` — e come scrive lo schematic `add-spartacus` — `core-libs/schematics/src/add-spartacus/update-app-config.ts:L83-L100`. Angular, però, avverte che il supporto agli interceptor forniti via DI "may be phased out in a later release" — `angular:packages/common/http/src/provider.ts:L173-L174`.
2. **L'ordine conta, e non è quello che sembra.** Più interceptor commentano esplicitamente la propria posizione: "MUST be provided as the first one" — `core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L37-L38`; "ClientTokenInterceptor must be imported after the AuthInterceptor" — `core-libs/core/src/auth/auth.module.ts:L13-L16`. L'ordine reale deriva dalle regole con cui Angular appiattisce i provider degli NgModule (`walkProviderTree`), che differiscono dall'ordine testuale degli `imports`. In §3.5–§3.6 la catena dell'app demo è ricostruita interceptor per interceptor.
3. `HttpContext` **come canale laterale.** Spartacus passa informazioni agli interceptor tramite `HttpContextToken` (`OCC_HTTP_TOKEN`, `HTTP_TIMEOUT_CONFIG`) — `core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20`, `core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L18` — e tramite header "marcatore" JSON (`InterceptorUtil`) — `core-libs/core/src/occ/utils/interceptor-util.ts:L9-L46`.

Prerequisiti: multi-provider e gerarchia degli injector (01 — Dependency Injection); bootstrap e `importProvidersFrom` (02 — NgModule, standalone e bootstrap). Il livello che *usa* `HttpClient` (adapter, `OccEndpointsService`) è in 21 — Livello OCC; la logica di token/refresh in 22 — Autenticazione; la gestione degli errori e dei global message in 41 — Errori HTTP e global message. L'elenco completo e rigenerabile di classi e registrazioni è nell'appendice A06.

2. Mappa dei file

Angular (`/tmp/angular`, prefisso `angular:`)

PATH	RUOLO
angular:packages/common/http/src/client.ts	`HttpClient`: `request()` crea l'`HttpRequest` e invoca l'`HttpHandler` dentro `concatMap`
angular:packages/common/http/src/backend.ts	`HttpBackend`, `HttpHandler`, `HttpInterceptorHandler` (costruisce la catena funzionale)
angular:packages/common/http/src/interceptor.ts	`HttpInterceptor`, `HttpInterceptorFn`, `HTTP_INTERCEPTORS`, `HTTP_INTERCEPTOR_FNS`, `HTTP_ROOT_INTERCEPTOR_FNS`, `legacyInterceptorFnFactory`, adattatori di catena
angular:packages/common/http/src/provider.ts	`provideHttpClient`, `withInterceptors`, `withInterceptorsFromDi`, `withFetch`, `withRequestsMadeViaParent`
angular:packages/common/http/src/module.ts	`HttpClientModule` (deprecato)
angular:packages/common/http/src/context.ts	`HttpContextToken`, `HttpContext`
angular:packages/common/http/src/request.ts	`HttpRequest` (`context`, `clone`, `urlWithParams`, `timeout`)
angular:packages/common/http/src/params.ts	`HttpParams` immutabili, codec di default
angular:packages/common/http/src/xsrf.ts	`xsrfInterceptorFn`, `HttpXsrfCookieExtractor`
angular:packages/common/http/src/fetch.ts	`FetchBackend`
angular:packages/common/http/src/xhr.ts	`HttpXhrBackend`
angular:packages/common/http/src/transfer_cache.ts	`withHttpTransferCache` (root interceptor fn)
angular:packages/platform-server/src/http.ts	`SERVER_HTTP_PROVIDERS`, `relativeUrlsTransformerInterceptorFn`
angular:packages/platform-browser/src/hydration.ts	`provideClientHydration`, `withNoHttpTransferCache`
angular:packages/core/src/di/provider_collection.ts	`importProvidersFrom`, `walkProviderTree`: ordine di appiattimento dei provider
angular:packages/core/src/di/r3_injector.ts	`processProvider`: array dei multi-provider

App demo

PATH	RUOLO
projects/storefrontapp/src/app/app.config.ts	`provideHttpClient(withFetch(), withInterceptorsFromDi())` e `importProvidersFrom(AppModule)`

PATH	RUOLO
projects/storefrontapp/src/app/app.config.server.ts	provideServerRendering() + merge con appConfig
projects/storefrontapp/src/app/app.module.ts	Radice dell'albero NgModule
projects/storefrontapp/src/app/spartacus/spartacus.module.ts	BaseStorefrontModule → SpartacusFeaturesModule → SpartacusConfigurationModule
projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts	Ordine dei moduli auth/feature che portano interceptor
projects/storefrontapp/src/environments/environment.ts	Flag dei moduli condizionali (B2B, CDS, OPF, ...)

Spartacus core (@spartacus/core)

PATH	RUOLO
core-libs/core/src/base-core.module.ts	Ordine degli import core (ErrorHandlingModule per primo)
core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts	HttpErrorHandlerInterceptor
core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts	HttpErrorInterceptor (global message)
core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts	SiteContextInterceptor (core)
core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts	WithCredentialsInterceptor
core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts	HttpTimeoutInterceptor
core-libs/core/src/http/http-timeout/http-timeout.config.ts	HTTP_TIMEOUT_CONFIG, augmentation di BackendConfig
core-libs/core/src/auth/auth.module.ts	Ordine UserAuthModule → ClientAuthModule
core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts	AuthInterceptor
core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts	Header Authorization, retry su token scaduto
core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts	TokenRevocationInterceptor
core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts	ClientTokenInterceptor
core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts	AnonymousConsentsInterceptor
core-libs/core/src/occ/utils/interceptor-util.ts	InterceptorUtil e costanti degli header marcatore
core-libs/core/src/occ/utils/occ-http-token.ts	OCC_HTTP_TOKEN / OccHttpContext
core-libs/core/src/occ/services/occ-endpoints.service.ts	buildUrl con HttpParams
core-libs/core/src/util/http-params-uri.encoder.ts	HttpParamsURIEncoder

PATH	RUOLO
core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts	I18NEXT_HTTP_BACKEND_CLIENT (traduzioni via HttpClient)

Feature e integration libs (interceptor registrati)

PATH	RUOLO
feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts	CheckoutCartInterceptor
feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts	OccPersonalizationIdInterceptor (e gemello ... TimeInterceptor)
feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts	UserIdHttpHeaderInterceptor
feature-libs/asm/customer-360/root/interceptors/site-context.interceptor.ts	SiteContextInterceptor (ASM customer 360)
feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts	CmsTicketInterceptor
feature-libs/organization/account-summary/root/http-interceptors/blob-error.interceptor.ts	BlobErrorInterceptor (B2B)
integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts	ConsentReferenceInterceptor (CDS)
integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts	DebugInterceptor (CDS)
integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts	OccOppsCouponCodesInterceptor
integration-libs/opf/gift-card/root/http-interceptors/opf-gift-card-payment-api.interceptor.ts	OpfGiftCardPaymentApiInterceptor
integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts	PunchoutCartInterceptor
integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts	OccSegmentRefsInterceptor

3. Come funziona passo per passo

3.1 `HttpClient.request()`: dalla chiamata all'`HttpRequest`

1. `HttpClient` è `providedIn: 'root'` e ha una sola dipendenza: un `HttpHandler` — angular:packages/common/http/src/client.ts:L132-L134.
2. Tutti i metodi shorthand (`get`, `post`, ...) finiscono in `request()`. Se il primo argomento non è già un `HttpRequest`, ne costruisce uno con headers, `context`, params, `responseType` (default `'json'`), `withCredentials`, `timeout` e gli altri campi delle opzioni — angular:packages/common/http/src/client.ts:L807-L856.
3. Un `params` passato come oggetto viene convertito in `new HttpParams({fromObject: ...})` — angular:packages/common/http/src/client.ts:L826-L834.

4. L'handler, che contiene tutti gli interceptor, viene eseguito dentro `of(req).pipe(concatMap(...))`. Il commento nel sorgente spiega che in questo modo gli interceptor girano di nuovo **a ogni subscription** e a ogni retry — angular:packages/common/http/src/client.ts:L857-L863. La richiesta parte solo quando qualcuno si sottoscrive; un `retry()` applicato dal chiamante ripercorre l'intera catena.
5. Con `observe: 'events'`, o se si è passato un `HttpRequest`, si ottiene lo stream grezzo di `HttpEvent`; altrimenti vengono filtrati solo gli `HttpResponse` — angular:packages/common/http/src/client.ts:L868-L877.
6. Con `observe: 'body'` (default) si restituisce il body. Per `arraybuffer`, `blob` e `text` il tipo viene validato, contro "a misbehaving interceptor" che abbia cambiato formato; `json` non è validato — angular:packages/common/http/src/client.ts:L880-L931.

3.2 `HttpHandler`, `HttpInterceptorHandler`, `HttpBackend`

- `HttpHandler` è una classe astratta con default `useExisting: HttpInterceptorHandler` — angular:packages/common/http/src/backend.ts:L150-L153.
- `HttpBackend` è astratta, con default `HttpXhrBackend`. Il JSDoc avverte che iniettarla direttamente **salta la catena di interceptor** — angular:packages/common/http/src/backend.ts:L35-L48.
- `HttpInterceptorHandler.handle()` costruisce la catena in modo *lazy*, alla prima richiesta. Legge `HTTP_INTERCEPTOR_FNS` e `HTTP_ROOT_INTERCEPTOR_FNS`, accoda i secondi ai primi e deduplica con un `Set` — angular:packages/common/http/src/backend.ts:L101-L112.
- La catena è composta con `reduceRight` a partire da `interceptorChainEndFn`, così che l'esecuzione segua l'ordine dell'array: il primo elemento è il più esterno, e l'ultimo anello chiama `this.backend.handle` — angular:packages/common/http/src/backend.ts:L114-L134.
- Con `REQUESTS_CONTRIBUTE_TO_STABILITY` (default `true`), ogni richiesta aggiunge un task a `PendingTasks` e lo rimuove in `finalize` — angular:packages/common/http/src/backend.ts:L125-L129, angular:packages/common/http/src/interceptor.ts:L217-L221. È questo meccanismo a far attendere al rendering SSR la fine delle chiamate HTTP (vedi 07 — SSR, hydration e TransferState).
- In dev mode, su server, se il backend non è `FetchBackend` viene stampato una volta un warning che raccomanda `withFetch()` — angular:packages/common/http/src/backend.ts:L70-L97.

3.3 Due tipi di interceptor e l'adattatore "legacy"

- **Di classe**: `HttpInterceptor` ha il solo metodo `intercept(req, next: HttpHandler)` — angular:packages/common/http/src/interceptor.ts:L46-L63. Si registrano sul multi-token `HTTP_INTERCEPTORS` — angular:packages/common/http/src/interceptor.ts:L187-L197.
- **Funzionali**: `HttpInterceptorFn = (req, next: HttpHandlerFn) => Observable`, eseguiti in un injection context — angular:packages/common/http/src/interceptor.ts:L84-L84, angular:packages/common/http/src/interceptor.ts:L97-L98, angular:packages/common/http/src/interceptor.ts:L127-L130. Si registrano su `HTTP_INTERCEPTOR_FNS`, il cui default è `[xsrfInterceptorFn]` — angular:packages/common/http/src/interceptor.ts:L199-L212.
- `chainedInterceptorFn` esegue ogni interceptor funzionale in `runInInjectionContext(injector, ...)` — angular:packages/common/http/src/interceptor.ts:L174-L185.
- Il ponte tra i due mondi è `legacyInterceptorFnFactory`: produce **un unico** `HttpInterceptorFn` che, alla prima richiesta, fa `inject(HTTP_INTERCEPTORS, {optional: true})` e compone una sotto-catena delle classi, sempre con `reduceRight`, quindi nell'ordine dell'array — angular:packages/common/http/src/interceptor.ts:L227-L241.
- Ogni classe viene adattata da `adaptLegacyInterceptorToChain`, che le passa come `next` un oggetto `{handle}` il quale invoca *il resto* della catena — angular:packages/common/http/src/interceptor.ts:L160-

L168. Ne segue che il `next` ricevuto da un interceptor rappresenta solo gli interceptor **a valle** più il backend. Su questo si reggono i retry dopo un 401 (§3.12).

- Anche la sotto-catena legacy registra un proprio `PendingTasks` per richiesta, in aggiunta a quello di `HttpInterceptorHandler` — angular:packages/common/http/src/interceptor.ts:L243-L250.

3.4 `provideHttpClient` e la configurazione dell'app demo

1. `provideHttpClient()` registra `HttpClient`, `HttpInterceptorHandler`, `HttpHandler → HttpInterceptorHandler`, un `HttpBackend` via factory (`FETCH_BACKEND` se presente, altrimenti `HttpXhrBackend`) e `xsrfInterceptorFn` come primo `HTTP_INTERCEPTOR_FNS`. Poi accoda i provider delle feature nell'ordine in cui sono passate — angular:packages/common/http/src/provider.ts:L118-L139.
2. Le feature (`HttpFeatureKind`) sono `Interceptors`, `LegacyInterceptors`, `CustomXsrfConfiguration`, `NoXsrfProtection`, `JsonpSupport`, `RequestsMadeViaParent`, `Fetch` — angular:packages/common/http/src/provider.ts:L35-L43. In dev mode alcune combinazioni sono vietate (`withXsrfConfiguration` + `withNoXsrfProtection`, `withRequestsMadeViaParent` + `withFetch`) — angular:packages/common/http/src/provider.ts:L97-L116.
3. `withInterceptors(fns)` aggiunge ogni funzione a `HTTP_INTERCEPTOR_FNS`, nell'ordine dato — angular:packages/common/http/src/provider.ts:L150-L163.
4. `withInterceptorsFromDi()` registra `LEGACY_INTERCEPTOR_FN` (factory `legacyInterceptorFnFactory`) e lo aggiunge a `HTTP_INTERCEPTOR_FNS` via `useExisting`. Se la feature è inclusa più volte, la funzione resta la stessa istanza e il `Set` di §3.2 la deduplica — angular:packages/common/http/src/provider.ts:L169-L197.
5. `withFetch()` registra `FetchBackend`, `FETCH_BACKEND` e `HttpBackend → FetchBackend` — angular:packages/common/http/src/provider.ts:L302-L308.
6. `withRequestsMadeViaParent()` sostituisce l'`HttpBackend` con l'`HttpHandler` dell'injector padre — angular:packages/common/http/src/provider.ts:L276-L291.
7. `HttpClientModule` è `@deprecated` ed equivale a `providers: [provideHttpClient(withInterceptorsFromDi())]` — angular:packages/common/http/src/module.ts:L94-L103.

App demo. `appConfig` contiene `provideHttpClient(withFetch(), withInterceptorsFromDi())`, poi `provideClientHydration(withEventReplay(), withNoHttpTransferCache())` e infine `importProvidersFrom(AppModule)` — projects/storefrontapp/src/app/app.config.ts:L26-L35. Quindi:

- `HTTP_INTERCEPTOR_FNS = [xsrfInterceptorFn, LEGACY_INTERCEPTOR_FN]`, il backend è `FetchBackend`, e non c'è `transferCacheInterceptorFn` perché la transfer cache HTTP è disattivata — angular:packages/common/http/src/provider.ts:L128-L136, angular:packages/common/http/src/provider.ts:L186-L196, angular:packages/platform-browser/src/hydration.ts:L255-L266;
- sul server la config è `mergeApplicationConfig(appConfig, serverConfig)` con `provideServerRendering()` — projects/storefrontapp/src/app/app.config.server.ts:L17-L30. `provideServerRendering` porta `SERVER_HTTP_PROVIDERS` — angular:packages/platform-server/src/provide_server.ts:L28-L33, angular:packages/platform-server/src/server.ts:L75-L78 — che registra `relativeUrlsTransformerInterceptorFn` su `HTTP_ROOT_INTERCEPTOR_FNS`: in SSR gli URL relativi vengono risolti contro protocollo, host e porta di `PlatformLocation` — angular:packages/platform-server/src/http.ts:L72-L110.

Nessun file in `projects/storefrontapp/src` registra `HTTP_INTERCEPTORS`, e nessuna libreria Spartacus usa `HttpClientModule` o `withInterceptors`: tutti gli interceptor applicativi arrivano dalle librerie, come classi — projects/storefrontapp/src/app/app.config.ts:L7-L11, A06.

3.5 Come Angular ordina i multi-provider (le regole che decidono la catena)

L'ordine dell'array `HTTP_INTERCEPTORS` è l'ordine in cui l'`R3Injector` processa i provider `multi: true`: ognuno viene accodato a `multiRecord.multi`, e la factory restituisce `injectArgs(multiRecord.multi)` — angular:packages/core/src/di/r3_injector.ts:L224-L226, angular:packages/core/src/di/r3_injector.ts:L497-L512. Per conoscere la catena bisogna dunque sapere in che ordine arrivano i provider.

- `bootstrapApplication` passa all'injector root `[provideZonelessChangeDetectionInternal(), errorHandlerEnvironmentInitializer, ..., ...appProviders]`, quindi i provider dell'`ApplicationConfig` nell'ordine dichiarato — angular:packages/core/src/application/create_application.ts:L66-L74. `forEachSingleProvider` appiattisce array ed `EnvironmentProviders` in profondità, senza cambiare l'ordine — angular:packages/core/src/di/r3_injector.ts:L725-L737. `mergeApplicationConfig` concatena i `providers` nell'ordine degli argomenti — angular:packages/core/src/application/application_config.ts:L31-L37.
- `importProvidersFrom(AppModule)` usa `walkProviderTree` — angular:packages/core/src/di/provider_collection.ts:L144-L188. Valgono tre regole:
  1. **Prima gli import, poi i provider statici.** Per un NgModule vengono visitati ricorsivamente tutti gli `imports` (depth-first, nell'ordine), e solo dopo vengono emessi i `providers` di `@NgModule` del modulo stesso — angular:packages/core/src/di/provider_collection.ts:L292-L306, angular:packages/core/src/di/provider_collection.ts:L346-L353.
  2. **I `forRoot()` sono differiti.** I `providers` dei `ModuleWithProviders` importati da un modulo vengono emessi **dopo** che tutti gli import di quel modulo sono stati visitati, nell'ordine di import ("following View Engine semantics ... FW-1349") — angular:packages/core/src/di/provider_collection.ts:L304-L321, angular:packages/core/src/di/provider_collection.ts:L195-L209.
  3. **Un `forRoot()` senza `providers` non è differito.** `walkProviderTree` restituisce `true` (provider differiti) solo se il container è un `ModuleWithProviders` con `providers !== undefined` — angular:packages/core/src/di/provider_collection.ts:L360-L362. `BaseCoreModule.forRoot()`, `AuthModule.forRoot()` e `HttpModule.forRoot()` restituiscono solo `{ngModule}`, quindi sono trattati come import semplici, visitati sul posto — core-libs/core/src/base-core.module.ts:L47-L52, core-libs/core/src/auth/auth.module.ts:L18-L23, core-libs/core/src/http/http.module.ts:L10-L18.
- Deduplicazione: un NgModule già visitato non riemette i suoi provider statici — angular:packages/core/src/di/provider_collection.ts:L275-L276, angular:packages/core/src/di/provider_collection.ts:L346-L362. Nessun root module con interceptor viene importato staticamente prima della posizione indicata in §3.6. L'unica eccezione nota è `OpfGiftCardModule` (lazy), che re-importa `OpfGiftCardRootModule` — integration-libs/opf/gift-card/opf-gift-card.module.ts:L18-L18.

3.6 La catena dell'app demo, interceptor per interceptor

Applicando le regole di §3.5 all'albero dei moduli dell'app demo (browser, flag d'ambiente di default, cioè B2C senza integrazioni — projects/storefrontapp/src/environments/environment.ts:L22-L44) si ottiene questo ordine. La catena *funzionale* è `xsrfInterceptorFn` → `LEGACY_INTERCEPTOR_FN` (→ sotto-catena qui sotto) → [`relativeUrlsTransformerInterceptorFn`, solo SSR] → `FetchBackend`.

Percorso nell'albero: `AppModule` importa `BrowserModule`, `StoreModule.forRoot({})`, `AppRoutingModule`, `EffectsModule.forRoot([])`, `SpartacusModule` — projects/storefrontapp/src/app/app.module.ts:L15-L25; `SpartacusModule` importa `BaseStorefrontModule`, `SpartacusFeaturesModule`, `SpartacusConfigurationModule` — projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L19; `BaseStorefrontModule` importa per primo `BaseCoreModule.forRoot()` — core-libs/storefront/base-storefront.module.ts:L26-L28.

#	INTERCEPTOR	DA DOVE ARRIVA	PERCHÉ IN QUESTA POSIZIONE
1	`HttpErrorHandlerInterceptor`	`ErrorHandlingModule` (1° import di `BaseCoreModule`) → `HttpErrorHandlerModule.forRoot()` — core-libs/core/src/error-handling/error-handling.module.ts:L13-L31, core-libs/core/src/error-handling/http-error-handler/http-error-handler.module.ts:L11-L24	`forRoot` differito, ma a fine import di `ErrorHandlingModule`, che è il primo visitato — core-libs/core/src/base-core.module.ts:L26-L45
2	`SiteContextInterceptor` (core)	`BaseOccModule` → `SiteContextOccModule`, provider **statici** — core-libs/core/src/occ/base-occ.module.ts:L17-L19, core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts:L18-L30	Statico: emesso subito durante la visita
3	`HttpTimeoutInterceptor`	`HttpModule` → `HttpTimeoutModule`, statico — core-libs/core/src/http/http-timeout/http-timeout.module.ts:L13-L22	`HttpModule.forRoot()` senza providers → visitato sul posto
4	`HttpErrorInterceptor`	`GlobalMessageModule.forRoot()` — core-libs/core/src/global-message/global-message.module.ts:L26-L31, core-libs/core/src/global-message/http-interceptors/index.ts:L65-L71	`forRoot` differito a fine import di `BaseCoreModule`
5	`WithCredentialsInterceptor`	`BaseOccModule.forRoot()` — core-libs/core/src/occ/base-occ.module.ts:L20-L29	Idem, dopo `GlobalMessageModule` nell'ordine di import
6	`DefaultOAuthInterceptor` (angular-oauth2-oidc)	`UserAuthModule` importa `OAuthModule.forRoot()` → `provideOAuthClient` — core-libs/core/src/auth/user-auth/user-auth.module.ts:L98-L99, deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L3118-L3146	Differito a fine import di `UserAuthModule`, primo modulo visitato sotto `AuthModule`
7	`AuthInterceptor`	`UserAuthModule.forRoot()` — core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L12-L23	`forRoot` differito a fine import di `AuthModule`
8	`TokenRevocationInterceptor`	idem	Secondo nell'array `interceptors`
9	`ClientTokenInterceptor`	`ClientAuthModule.forRoot()` — core-libs/core/src/auth/client-auth/client-auth.module.ts:L22-L28	Importato dopo `UserAuthModule.forRoot()` — core-libs/core/src/auth/auth.module.ts:L13-L16
10	`CheckoutCartInterceptor`	`CheckoutFeatureModule` → `CheckoutRootModule`, statico — feature-libs/checkout/base/root/checkout-root.module.ts:L51-L54	Statico, in posizione — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L261-L261
11	`OccPersonalizationIdInterceptor`	`PersonalizationRootModule`, statico — feature-libs/tracking/personalization/root/http-interceptors/index.ts:L13-L24	projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L263-L263
12	`OccPersonalizationTimeInterceptor`	idem	Secondo nell'array
13	`UserIdHttpHeaderInterceptor`	`AsmRootModule`, statico — feature-libs/asm/root/asm-root.module.ts:L73-L95	projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L265-L265
14	`SiteContextInterceptor` (ASM 360)	`AsmCustomer360RootModule`, statico — feature-libs/asm/customer-360/root/asm-customer-360-root.module.ts:L18-L35	projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L266-L266

#	INTERCEPTOR	DA DOVE ARRIVA	PERCHÉ IN QUESTA POSIZIONE
15	`CmsTicketInterceptor`	`SmartEditRootModule`, statico — feature-libs/smartedit/root/smart-edit-root.module.ts:L22-L24	projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L272-L272
16	`AnonymousConsentsInterceptor`	`AnonymousConsentsModule.forRoot()` — core-libs/core/src/anonymous-consents/anonymous-consents.module.ts:L18-L22	Importato a L214 ma `forRoot` **differito** a fine import di `SpartacusFeaturesModule`, quindi ultimo — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L214-L214

Note sulla tabella:

- `SpartacusFeaturesModule` importa per primo `AuthModule.forRoot()`, più avanti `AnonymousConsentsModule.forRoot()`, poi i feature module e in coda `...featureModules` — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L176-L285. Con i flag di default `featureModules` contiene solo `PickupInStoreFeatureModule`, che non registra interceptor — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L115-L175.
- **Varianti con flag d'ambiente.** I moduli condizionali entrano in `...featureModules`, cioè tra #15 e `AnonymousConsentsInterceptor`, con provider statici in quest'ordine: B2B `BlobErrorInterceptor`; CDS `ConsentReferenceInterceptor` e `DebugInterceptor`; OPPS `OccOppsCouponCodesInterceptor`; OPF `OpfGiftCardPaymentApiInterceptor`; Punchout `PunchoutCartInterceptor`; SegmentRefs `OccSegmentRefsInterceptor` — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L117-L163, feature-libs/organization/account-summary/root/account-summary-root.module.ts:L49-L57, integration-libs/cds/src/profiletag/profile-tag.module.ts:L16-L24, integration-libs/opps/root/coupon-codes/opps-coupon-codes.module.ts:L19-L21, integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L57-L61, integration-libs/punchout/root/punchout.root.module.ts:L36-L48, integration-libs/segment-refs/root/segment-refs-root.module.ts:L12-L17.
- DA VERIFICARE: l'ordine è stato derivato **staticamente** dalle regole di `walkProviderTree`/`R3Injector`. Non è stato osservato a runtime, perché `node_modules` non è installato e non è stato possibile stampare `inject(HTTP_INTERCEPTORS)`. Dipendenze npm non estratte (es. `@ngrx/store-devtools` in projects/storefrontapp/src/app/private/private.providers.ts:L52-L56) potrebbero registrare altri interceptor; il controllo è stato fatto solo per `angular-oauth2-oidc` (sì), `@ngrx/store`, `@ngrx/effects`, `@ngrx/router-store`, `@angular/ssr` (no).

Questa ricostruzione risolve i NOTA lasciati aperti sull'ordine in 21 — Livello OCC, 29 — ASM e 41 — Errori HTTP.

3.7 `HttpContext` e `HttpContextToken`

- `HttpContextToken<T>` contiene soltanto una funzione `defaultValue: () => T` — angular:packages/common/http/src/context.ts:L16-L18. L'identità del token è l'istanza stessa: la chiave della `Map` interna è l'oggetto token — angular:packages/common/http/src/context.ts:L55-L56.
- `HttpContext` è "mutable and is shared between cloned requests unless explicitly specified" — angular:packages/common/http/src/context.ts:L20-L26. `HttpRequest.clone()` passa la stessa istanza di `context` se `update.context` non è fornito — angular:packages/common/http/src/request.ts:L763-L781. Se non si passa un context, il costruttore ne crea uno nuovo — angular:packages/common/http/src/request.ts:L541-L542.
- `HttpContext.get()` su un token assente **scrive** il default nella mappa e poi lo restituisce — angular:packages/common/http/src/context.ts:L78-L83.

Uso in Spartacus:

- `OCC_HTTP_TOKEN = new HttpContextToken<OccHttpContext>(() => ({}))`, con l'unico campo `sendUserIdAsHeader?: string | boolean` — core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20. Gli adapter lo impostano con `new HttpContext().set(OCC_HTTP_TOKEN, {...})` e lo passano come `{ context }`; la ricerca prodotti, per esempio, usa `sendUserIdAsHeader: true` — core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L51-L56. Altri usi: cost center utente, configuratore variant, payment type B2B — core-libs/core/src/occ/adapters/user/occ-user-cost-centers.adapter.ts:L32-L32, feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.adapter.ts:L403-L403, feature-libs/checkout/b2b/occ/adapters/occ-checkout-payment-type.adapter.ts:L40-L40. Il consumatore è `UserIdHttpHeaderInterceptor` di ASM (§3.11).
- `HTTP_TIMEOUT_CONFIG = new HttpContextToken<HttpTimeoutConfig | undefined>(() => undefined)`. Permette di impostare il timeout di una singola richiesta, separato per browser e server; `BackendConfig` è esteso via module augmentation con `timeout` — core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L41.

3.8 `HttpParams` e il codec di Spartacus

- `HttpParams` è immutabile: ogni mutazione restituisce una nuova istanza — angular:packages/common/http/src/params.ts:L155-L165. `set()`/`delete()` delegano a un `clone()` privato che accumula `updates` e un riferimento `cloneFrom`; la valutazione è rimandata a `init()` — angular:packages/common/http/src/params.ts:L275-L288, angular:packages/common/http/src/params.ts:L314-L319. `fromString` e `fromObject` si escludono a vicenda — angular:packages/common/http/src/params.ts:L173-L178.
- Il codec di default applica `encodeURIComponent` e poi **ri-decodifica** `@ : $ , ; = ? /` — angular:packages/common/http/src/params.ts:L102-L119.
- `urlWithParams` concatena i params all'URL con `?`, `&` o niente, a seconda che l'URL contenga già una query — angular:packages/common/http/src/request.ts:L548-L566.
- `OccEndpointsService.buildUrl` costruisce la query con `new HttpParams({encoder: new HttpParamsURIEncoder()})`, conservando l'eventuale query già presente nell'endpoint (`fromString`). Un valore `null` cancella il param, `undefined` viene ignorato — core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L147, core-libs/core/src/occ/services/occ-endpoints.service.ts:L152-L167.
- `HttpParamsURIEncoder` usa *solo* `encodeURIComponent`, così i caratteri riservati RFC 3986 restano codificati — core-libs/core/src/util/http-params-uri.encoder.ts:L7-L23.

3.9 Header "marcatore": `InterceptorUtil`

Prima che esistesse `HttpContext`, Spartacus passava parametri agli interceptor come **header JSON**, e in parte lo fa ancora:

- `InterceptorUtil.createHeader` serializza il valore con `JSON.stringify` (in `append` o `set`), `getInterceptorParam` lo rilegge con `JSON.parse`, `removeHeader` clona la richiesta senza quell'header — core-libs/core/src/occ/utils/interceptor-util.ts:L13-L46.
- Costanti: `cx-use-client-token`, `cx-use-csagent-token`, `sap-commerce-cloud-captcha-token` — core-libs/core/src/occ/utils/interceptor-util.ts:L9-L11.
- Gli adapter che richiedono il client token (ordini guest, indirizzi, registrazione) chiamano `InterceptorUtil.createHeader(USE_CLIENT_TOKEN, true, headers)` — feature-libs/order/occ/adapters/occ-order.adapter.ts:L48-L48, core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L104-L104. `ClientTokenInterceptor` rimuove il marcatore prima che la richiesta lasci il browser (§3.11).

3.10 Backend: fetch, XHR, SSR, XSRF, transfer cache

- `FetchBackend` è `@Injectable()` (non root). Usa `FetchFactory.fetch` se fornito, altrimenti `globalThis.fetch` tramite una arrow function — angular:packages/common/http/src/fetch.ts:L57-L63. Esegue `fetch` fuori dalla zone di Angular ed emette `HttpEventType.Sent` **prima** di attendere la risposta — angular:packages/common/http/src/fetch.ts:L105-L118. Implementa `request.timeout` con `setTimeout` e `AbortController` — angular:packages/common/http/src/fetch.ts:L69-L84. Con `withCredentials: true` imposta `credentials = 'include'`, sovrascrivendo `req.credentials` — angular:packages/common/http/src/fetch.ts:L321-L332.
- `HttpXhrBackend` usa `XhrFactory`, applica `withCredentials` e `timeout` all'XHR ed emette anch'esso `Sent` — angular:packages/common/http/src/xhr.ts:L108-L113, angular:packages/common/http/src/xhr.ts:L158-L182, angular:packages/common/http/src/xhr.ts:L418-L418.
- `xsrfInterceptorFn` lascia passare GET/HEAD, le richieste cross-origin rispetto a `PlatformLocation.href` e quelle che hanno già l'header; alle altre aggiunge il token (header di default `X-XSRF-TOKEN`) — angular:packages/common/http/src/xsrf.ts:L93-L125, angular:packages/common/http/src/xsrf.ts:L40-L40. Su server il cookie extractor restituisce sempre `null` — angular:packages/common/http/src/xsrf.ts:L64-L75.
- `withHttpTransferCache` registra `transferCacheInterceptorFn` come root interceptor fn e lo disattiva dopo `appRef.whenStable()` — angular:packages/common/http/src/transfer_cache.ts:L439-L468. `provideClientHydration` la abilita salvo `withNoHttpTransferCache()` — angular:packages/platform-browser/src/hydration.ts:L255-L266. L'app demo la disattiva (§3.4): Spartacus trasferisce lo stato con i propri meccanismi, trattati in 07 e 39 — SSR ottimizzato.
- **Traduzioni.** `I18NEXT_HTTP_BACKEND_CLIENT` usa `HttpClient` (`responseType: 'text'`), quindi anche i JSON di traduzione attraversano tutta la catena di interceptor — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L16-L42. In SSR `I18nextHttpBackendInitializer.getLoadPath` rende assoluto il `loadPath` con `windowRef.location.origin` — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L73-L92 (vedi 37 — i18n).

3.11 Che cosa fa ciascun interceptor

Andata = modifica della richiesta; ritorno = gestione di risposta o errore.

Core

- **#1** `HttpErrorHandlerInterceptor`**.** All'andata non tocca nulla. Al ritorno inoltra ogni errore all'`ErrorHandler` di Angular, compresi errori RxJS come `TimeoutError` provenienti dagli interceptor successivi — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L27-L41. Lo fa solo fuori dal browser (`shouldHandleError = !windowRef.isBrowser()`), avvolgendo l'errore in `OutboundHttpError` oppure in `CmsPageNotFoundOutboundHttpError` per i 404 sull'endpoint `pages` — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L52-L98. Dettagli in 41.
- **#2** `SiteContextInterceptor` **(core).** Parte da lingua e valuta di default della config e le aggiorna con una subscribe a `LanguageService`/`CurrencyService` fatta nel costruttore. Aggiunge `lang` e `curr` a ogni richiesta il cui URL contiene `occEndpoints.getBaseUrl()` — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L30-L68. `getBaseUrl()` vale `baseUrl + prefix + baseSite` attivo — core-libs/core/src/occ/services/occ-endpoints.service.ts:L81-L101. Vedi 13 — Site context.
- **#3** `HttpTimeoutInterceptor`**.** Prende il timeout da `HTTP_TIMEOUT_CONFIG` o, in mancanza, da `config.backend.timeout`, scegliendo `browser` o `server` in base alla piattaforma; se il valore è `undefined` non applica alcun timeout — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L44-L47, core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L77-L84. Il conto parte solo all'evento `Sent`, e un `TimeoutError` diventa un `HttpErrorResponse` con messaggio contenente URL e millise-

condi, loggato con `warn` — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L65, core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L101-L123.

- **#4** `HttpErrorInterceptor`. Su `HttpErrorResponse` sceglie con `resolveApplicable` tra gli `HttpErrorHandler` multi ottenuti da `UnifiedInjector.getMulti` (root + moduli lazy) e chiama `handleError` — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L26-L65.
- **#5** `WithCredentialsInterceptor`. Imposta `withCredentials: true` se `backend.occ.useWithCredentials` è definito e l'URL contiene `backend.occ.prefix` — core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L28-L52.
- **#6** `DefaultOAuthInterceptor`. Passa la richiesta invariata se `OAuthModuleConfig` è assente o `null` — deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L3075-L3081. Spartacus chiama `OAuthModule.forRoot()` senza argomenti, quindi il config è `null` — core-libs/core/src/auth/user-auth/user-auth.module.ts:L98-L99, deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L3118-L3146. In pratica è un pass-through.
- **#7** `AuthInterceptor`. Calcola `shouldCatchError` e `shouldAddAuthorizationHeader`, prende il token stabile (`take(1)`) solo se serve, e modifica la richiesta con `AuthHttpHeaderService.alterRequest` — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L35-L49. `alterRequest` aggiunge `Authorization` solo a URL OCC che non hanno già l'header e che non sono la richiesta `baseSites` — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L188-L242. Al ritorno: con un 401 da token scaduto su URL OCC → `handleExpiredAccessToken`; con un 401 `invalid_token` dal token endpoint → `handleExpiredRefreshToken()` ed `EMPTY` — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L54-L83. Con un 400 `invalid_grant` su richiesta `grant_type=refresh_token` → `handleExpiredRefreshToken()`, e l'errore viene comunque rilanciato — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L74-L84.
- **#8** `TokenRevocationInterceptor`. È attivo solo se `sendAuthHeaderOnRevoke()` è vero; aggiunge `Authorization` soltanto alla chiamata verso il revoke endpoint — core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L33-L57, core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L112-L113.
- **#9** `ClientTokenInterceptor`. Rimuove **sempre** il marcatore `cx-use-client-token`, anche quando è disattivato. Solo se `authentication.useClientTokens` è vero (default `false` se assente) aggiunge `Authorization` con il client token alle richieste OCC marcate — core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L33-L68. Su un 401 da token scaduto delega a `ClientErrorHandlingService.handleExpiredClientToken(request, next)` — core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L70-L85, core-libs/core/src/auth/client-auth/services/client-error-handling.service.ts:L30-L37.
- **#16** `AnonymousConsentsInterceptor`. Sugli URL OCC aggiunge l'header `X-Anonymous-Consents` con i consensi serializzati. Sulla risposta dell'endpoint `anonymousConsentTemplates` aggiorna i consensi, se l'utente è anonimo — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L40-L107, core-libs/core/src/model/consent.model.ts:L32-L32.

Nell'app demo `ClientTokenInterceptor` **e** `TokenRevocationInterceptor` **non fanno nulla.** `useClientTokens: true` e `sendAuthHeaderOnRevoke: true` compaiono nel default solo quando il toggle `authorizationCodeFlowByDefault` è falso — core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L48-L73, core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L116-L128. L'app demo invece attiva `authorizationCodeFlowByDefault`, `asyncAuthConfigInitializer` e `oauthCallbackPage` — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L316-L316, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L377-L377, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L395-L395.

Feature e integration

- **#10** `CheckoutCartInterceptor`. Legge lo stato del router (`take(1)`). Su un 400 `CartError/notFound` ricevuto mentre la semantic route inizia con `checkout`, naviga a `cart` e ricarica il carrello — feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L37-L54, feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L84-L89.
- **#11/#12 personalization.** Si attivano solo in browser, con `localStorage` disponibile e `personalization.enabled` (default `false`). Propagano l'header configurato e salvano in `localStorage` il valore ricevuto — feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L33-L97, feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L10-L11.
- **#13** `UserIdHttpHeaderInterceptor`. Agisce solo se `asm.userIdHttpHeader.enable` (default `true`). Legge `OCC_HTTP_TOKEN` e aggiunge `sap-commerce-cloud-user-id` con l'id passato come stringa, oppure con quello di `UserIdService.takeUserId()` se non è una costante (`current`, `anonymous`, …) — feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L31-L83, feature-libs/asm/root/config/default-asm-config.ts:L77-L79. Vedi 29 — ASM.
- **#14** `SiteContextInterceptor` **(ASM 360).** Aggiunge `lang`/`curr` solo alle richieste verso `baseUrl` (senza prefix e baseSite) che contengono `/assistedservicewebservices/` — feature-libs/asm/customer-360/root/interceptors/site-context.interceptor.ts:L58-L70.
- **#15** `CmsTicketInterceptor`. Agisce solo se esiste un `cmsTicketId`: aggiunge il param `cmsTicketId` alle URL `/cms/` e `/products/` e, per `/productList`, anche `categoryCode` preso dal page context — feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L29-L72.
- **Condizionali.** `BlobErrorInterceptor` (B2B) converte, in browser, un errore con body `Blob` JSON in un nuovo `HttpErrorResponse` con l'errore parsato — feature-libs/organization/account-summary/root/http-interceptors/blob-error.interceptor.ts:L31-L54. `ConsentReferenceInterceptor` e `DebugInterceptor` (CDS) aggiungono `X-Consent-Reference` e `X-Profile-Tag-Debug` — integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts:L28-L40, integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts:L28-L36. `OccOppsCouponCodesInterceptor` legge il coupon nel costruttore e lo invia solo in browser — integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L28-L64. `OpfGiftCardPaymentApiInterceptor` agisce solo su `/orders/paymentAuthorizedOrderPlacement` e ricarica il carrello in caso di errore — integration-libs/opf/gift-card/root/http-interceptors/opf-gift-card-payment-api.interceptor.ts:L32-L50. `PunchoutCartInterceptor` aggiunge (append) `punchoutsid` alle richieste `/carts/{punchoutCartId}` — integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts:L24-L46. `OccSegmentRefsInterceptor` legge `segmentrefs` da query string o `localStorage` nel costruttore — integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L39-L82. Queste integrazioni sono trattate in 34, 35 e 36.

3.12 Retry dopo un 401: chi rivede la richiesta

1. `AuthInterceptor` (#7) riceve l'errore da valle — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L53-L57.
2. Chiama `handleExpiredAccessToken(request, next, token)`. Qui `next` è l'handler costruito da `adaptLegacyInterceptorToChain`, cioè il resto della catena a partire da #8 — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L60-L64, angular:packages/common/http/src/interceptor.ts:L164-L167.
3. `getValidToken` produce il nuovo token, poi `next.handle(createNewRequestWithNewToken(...))`; se il token manca restituisce `EMPTY` — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L247-L259. `createNewRequestWithNewToken` usa `setHeaders: { Authorization }`, quindi sovrascrive l'header precedente — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L382-L392.

4. La richiesta ritentata attraversa solo #8–#16 e il backend. #1–#6 non la rivedono, ma l'oggetto richiesta conserva i params e gli header già aggiunti all'andata (`lang`/`curr`, `withCredentials`).

`ClientTokenInterceptor` (#9) segue lo stesso schema, e il suo retry attraversa solo #10–#16 — `core-libs/core/src/auth/client-auth/services/client-error-handling.service.ts:L30-L37`. Secondo il commento in `AuthModule`, questo ordine fa sì che `ClientTokenInterceptor` sia "the first to handle 401 errors" — `core-libs/core/src/auth/auth.module.ts:L13-L15`: essendo più interno, vede l'errore prima di `AuthInterceptor`.

---

4. Diagrammi

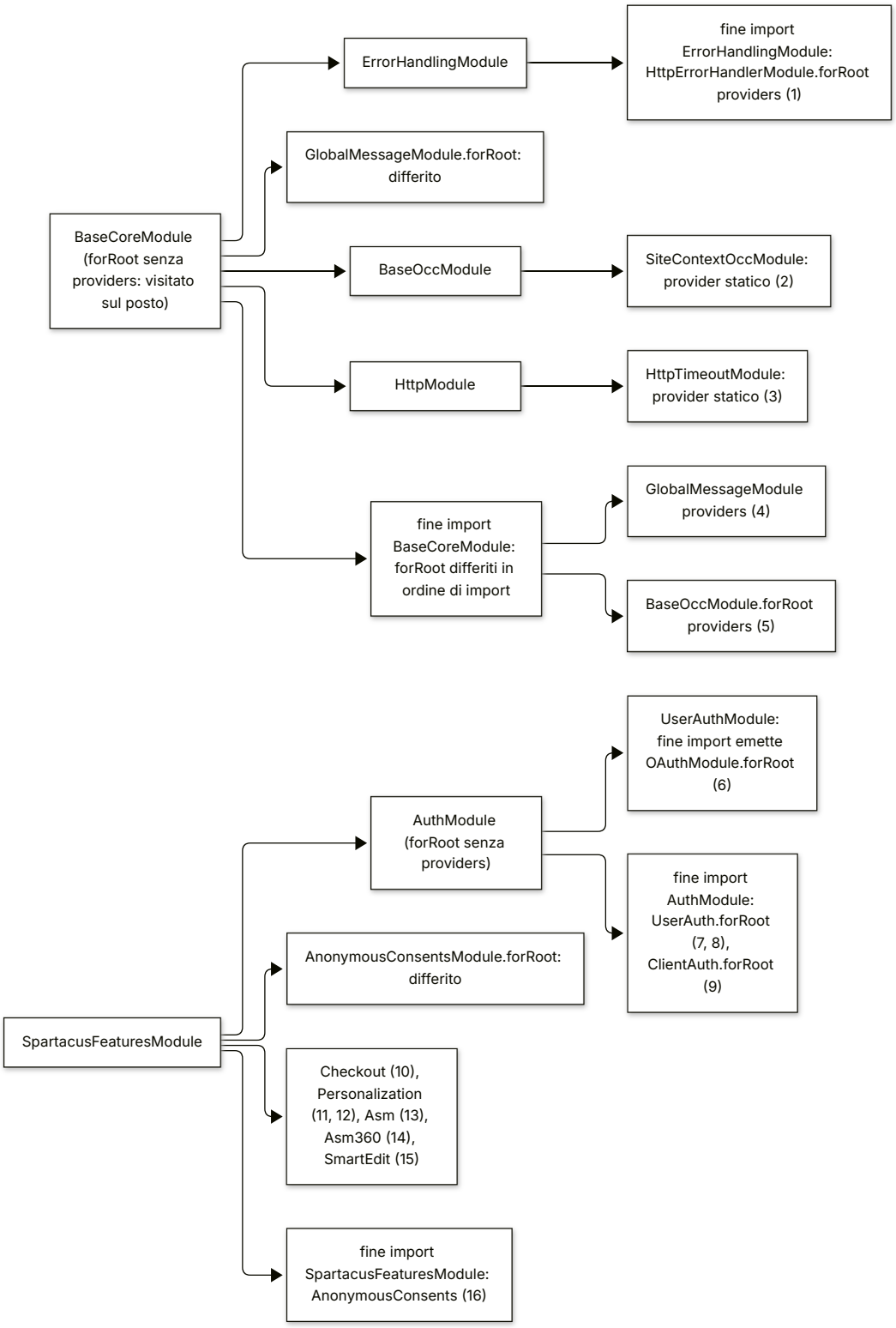
4.1 Costruzione lazy della catena (prima richiesta)

Ogni nodo corrisponde a §3.2–§3.4.

■ **TAVOLA 06.1** diagramma a tutta pagina, in fondo al capitolo

4.2 Come `walkProviderTree` produce l'ordine (estratto app demo)

I numeri tra parentesi sono le posizioni della tabella di §3.6.

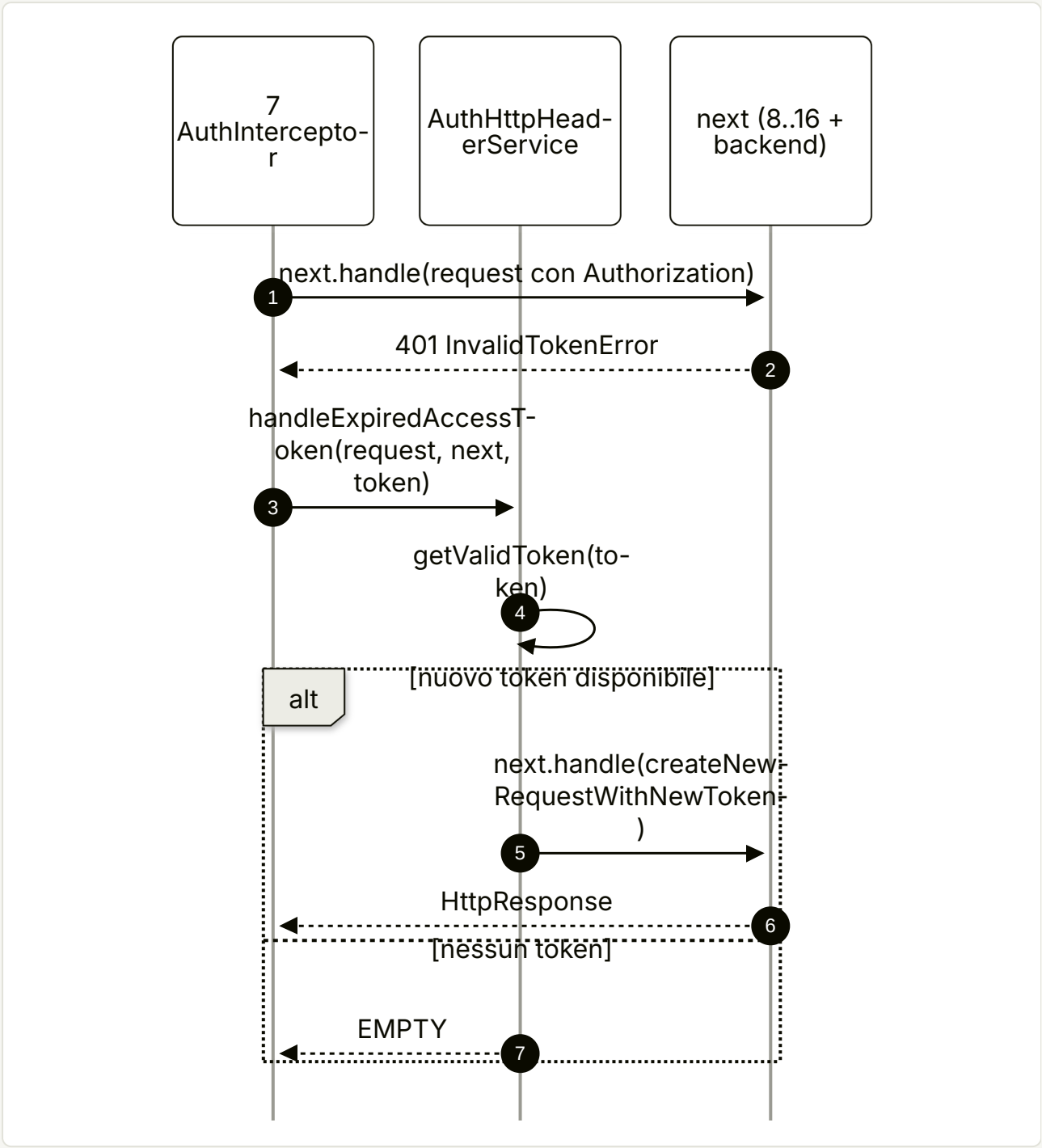


4.3 Percorso di una richiesta OCC (browser, app demo)

Corrisponde a §3.11; il ritorno percorre la catena al contrario.

■ **TAVOLA 06.2** diagramma a tutta pagina, in fondo al capitolo

4.4 Retry dopo 401 con token utente scaduto



5. Estratti di codice reale

5.1 La catena funzionale di Angular

angular:packages/common/http/src/backend.ts:L101-L135

```ts
  handle(initialRequest: HttpRequest<any>): Observable<HttpEvent<any>> {
    if (this.chain === null) {
      const parentHandler = this.injector.get(HttpHandler, null, {skipSelf: true});
      const isDelegating = parentHandler !== null && this.backend === parentHandler;
      const rootInterceptorFns = this.injector.get(
        HTTP_ROOT_INTERCEPTOR_FNS,
        [],
        isDelegating ? {self: true} : undefined,
      );
      const dedupedInterceptorFns = Array.from(
        new Set([...this.injector.get(HTTP_INTERCEPTOR_FNS), ...rootInterceptorFns]),
      );

      // Note: interceptors are wrapped right-to-left so that final execution order is
      // left-to-right. That is, if `dedupedInterceptorFns` is the array `[a, b, c]`, we want to
      // produce a chain that is conceptually `c(b(a(end)))`, which we build from the inside
      // out.
      this.chain = dedupedInterceptorFns.reduceRight(
        (nextSequencedFn, interceptorFn) =>
          chainedInterceptorFn(nextSequencedFn, interceptorFn, this.injector),
        interceptorChainEndFn as ChainedInterceptorFn<unknown>,
      );
    }

    if (this.contributeToStability) {
      const removeTask = this.pendingTasks.add();
      return this.chain(initialRequest, (downstreamRequest) =>
        this.backend.handle(downstreamRequest),
      ).pipe(finalize(removeTask));
    } else {
      // ...
    }
  }
```

5.2 L'adattatore delle classi e la sotto-catena legacy

angular:packages/common/http/src/interceptor.ts:L160-L168

```ts
export function adaptLegacyInterceptorToChain(
  chainTailFn: ChainedInterceptorFn<any>,
  interceptor: HttpInterceptor,
): ChainedInterceptorFn<any> {
  return (initialRequest, finalHandlerFn) =>
    interceptor.intercept(initialRequest, {
      handle: (downstreamRequest) => chainTailFn(downstreamRequest, finalHandlerFn),
    });
}
```

angular:packages/common/http/src/interceptor.ts:L227-L250

```ts
export function legacyInterceptorFnFactory(): HttpInterceptorFn {
  let chain: ChainedInterceptorFn<any> | null = null;

  return (req, handler) => {
    if (chain === null) {
      const interceptors = inject(HTTP_INTERCEPTORS, {optional: true}) ?? [];
      // Note: interceptors are wrapped right-to-left so that final execution order is
      // left-to-right. ...
      chain = interceptors.reduceRight(
        adaptLegacyInterceptorToChain,
        interceptorChainEndFn as ChainedInterceptorFn<any>,
      );
    }

    const pendingTasks = inject(PendingTasks);
    const contributeToStability = inject(REQUESTS_CONTRIBUTE_TO_STABILITY);
    if (contributeToStability) {
      const removeTask = pendingTasks.add();
      return chain(req, handler).pipe(finalize(removeTask));
    // ...
```

5.3 `withInterceptorsFromDi` e la deduplicazione

angular:packages/common/http/src/provider.ts:L186-L197

```ts
  return makeHttpFeature(HttpFeatureKind.LegacyInterceptors, [
    {
      provide: LEGACY_INTERCEPTOR_FN,
      useFactory: legacyInterceptorFnFactory,
    },
    {
      provide: HTTP_INTERCEPTOR_FNS,
      useExisting: LEGACY_INTERCEPTOR_FN,
      multi: true,
    },
  ]);
}
```

5.4 La configurazione HTTP dell'app demo

projects/storefrontapp/src/app/app.config.ts:L26-L35

```ts
export const appConfig: ApplicationConfig = {
  providers: [
    provideHttpClient(withFetch(), withInterceptorsFromDi()),
    provideClientHydration(withEventReplay(), withNoHttpTransferCache()),
```

```
    provideZoneChangeDetection({ eventCoalescing: true }),
    provideBrowserGlobalErrorListeners(),

    importProvidersFrom(AppModule),
  ],
};
```

5.5 Il differimento dei `ModuleWithProviders`

angular:packages/core/src/di/provider_collection.ts:L303-L321

```ts
        deepForEach(injDef.imports, (imported) => {
          if (walkProviderTree(imported, visitor, parents, dedup)) {
            importTypesWithProviders ||= [];
            // If the processed import is an injector type with providers, we store it in the
            // list of import types with providers, so that we can process those afterwards.
            importTypesWithProviders.push(imported);
          }
        });
      } finally {
        // Remove it from the parents set when finished.
        ngDevMode && parents.pop();
      }

      // Imports which are declared with providers (TypeWithProviders) need to be processed
      // after all imported modules are processed. This is similar to how View Engine
      // processes/merges module imports in the metadata resolver. See: FW-1349.
      if (importTypesWithProviders !== undefined) {
        processInjectorTypesWithProviders(importTypesWithProviders, visitor);
      }
```

5.6 L'ordine dichiarato dai moduli core

core-libs/core/src/base-core.module.ts:L26-L46

```ts
@NgModule({
  imports: [
    ErrorHandlingModule.forRoot(), // Import this module before any other interceptor to handle HTTP
errors efficiently
    StateModule.forRoot(),
    ConfigModule.forRoot(),
    // ...
    GlobalMessageModule.forRoot(),
    ProcessModule.forRoot(),
    FeaturesConfigModule.forRoot(),
    SiteContextModule.forRoot(), // should be imported after RouterModule.forRoot, because it overwrites
UrlSerializer
```

```
    MetaTagConfigModule.forRoot(),
    BaseOccModule.forRoot(),
    LazyLoadingModule.forRoot(),
    HttpModule.forRoot(),
    SiteThemeModule.forRoot(),
    FederatedLoginModule,
  ],
})
```

core-libs/core/src/auth/auth.module.ts:L12-L17

```ts
@NgModule({
  // ClientAuthModule should always be imported after UserAuthModule because the ClientTokenInterceptor must be imported after the AuthInterceptor.
  // This way, the ClientTokenInterceptor is the first to handle 401 errors and attempt to refresh the client token.
  // If the request is not for the client token, the AuthInterceptor handles the refresh.
  imports: [CommonModule, UserAuthModule.forRoot(), ClientAuthModule.forRoot()],
})
```

5.7 Un interceptor di sola andata: `SiteContextInterceptor`

core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68

```ts
  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    if (request.url.includes(this.occEndpoints.getBaseUrl())) {
      request = request.clone({
        setParams: {
          lang: this.activeLang ?? '',
          curr: this.activeCurr ?? '',
        },
      });
    }

    return next.handle(request);
  }
```

5.8 Un interceptor di solo ritorno: `HttpErrorHandlerInterceptor`

core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L48-L74

```ts
  intercept(
    request: HttpRequest<unknown>,
    next: HttpHandler
```

```
  ): Observable<HttpEvent<unknown>> {
    return next.handle(request).pipe(
      tap({
        error: (error: unknown) => {
          if (this.shouldHandleError(error)) {
            this.handleError(error);
          }
        },
      })
    );
  }

  /**
   * Determine if the error should be handled by the `ErrorHandler`.
   *
   * Be default, we avoid sending unpredictable errors to the browser's console, to prevent
   * possibly exposing there potentially confidential user's data.
   * This isn't an issue in SSR, where pages are rendered anonymously.
   * Moreover, in SSR we want to capture all app's errors, so we can potentially send
   * a HTTP error response (e.g. 500 error page) from SSR to a client.
   */
  protected shouldHandleError(_error: unknown): boolean {
    return !this.windowRef.isBrowser();
  }
```

5.9 Timeout che parte dal `Sent`

core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L40-L66

TS

```
  intercept(
    request: HttpRequest<unknown>,
    next: HttpHandler
  ): Observable<HttpEvent<unknown>> {
    const timeoutValue = this.getTimeoutValue(request);
    if (typeof timeoutValue === 'undefined') {
      return next.handle(request);
    }

    return next.handle(request).pipe(
      switchMap((event) => {
        // When event `HttpEventType.Sent` happens, let's start counting time for timeout.
        // But when event `HttpEventType.Response` is received, `switchMap` will unsubscribe from the
  following timeout observable.
        if (event.type === HttpEventType.Sent) {
          return NEVER.pipe(startWith(event), timeout(timeoutValue));
        }
        return of(event);
      }),
```

```
      catchError((error) => {
        throw this.convertTimeoutToHttpErrorResponse({
          error,
          request,
          timeoutValue,
        });
      })
    );
  }
```

5.10 `HttpContextToken` in Spartacus: definizione e uso

core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20

```ts
export interface OccHttpContext {
  /**
   * Indicates to the interceptor that the emulated customer ID should be sent to the OCC
   * to avoid ambiguity.
   * 'string' if the request has the user ID at hand.
   * 'boolean' (or, 'true') if the request does not have the user ID but still needs to send the
emulated ID.
   * @see feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts
   */
  sendUserIdAsHeader?: string | boolean;
}

export const OCC_HTTP_TOKEN = new HttpContextToken<OccHttpContext>(() => ({}));
```

core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L51-L56

```ts
    const context = new HttpContext().set(OCC_HTTP_TOKEN, {
      sendUserIdAsHeader: true,
    });

    return this.http
      .get(this.getSearchEndpoint(query, searchConfig, scope), { context })
```

5.11 Header marcatore e client token

core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L33-L52

```ts
  protected enableClientToken =
    inject(AuthConfig).authentication?.useClientTokens ?? false;

  constructor(
    protected clientTokenService: ClientTokenService,
    protected clientErrorHandlingService: ClientErrorHandlingService,
```

```
    protected occEndpoints: OccEndpointsService
  ) {}

  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    const isClientTokenRequest = this.isClientTokenRequest(request);
    if (isClientTokenRequest) {
      request = InterceptorUtil.removeHeader(USE_CLIENT_TOKEN, request);
    }
    if (!this.enableClientToken) {
      return next.handle(request);
    }
```

5.12 `WithCredentialsInterceptor` e il controllo `!== undefined`

core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L43-L48

```ts
  protected requiresWithCredentials(request: HttpRequest<any>): boolean {
    return (
      this.occConfig?.useWithCredentials !== undefined &&
      request.url.indexOf(this.occConfig?.prefix ?? '') > -1
    );
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`backend.timeout.server` / `.browser`	`number` (ms)	server `20_000`, browser assente	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L23-L37; default core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15	core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L79-L83
`HTTP_TIMEOUT_CONFIG` (HttpContextToken, per richiesta)	`Http Timeout Config | undefined`	undefined	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L18	core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L78-L78
`backend.occ.useWithCredentials`	`boolean` (conta solo se definito)	assente	core-libs/core/src/occ/config/occ-config.ts:L17-L21	core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L45-L46

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.prefix	string	'/occ/v2/' (libreria); la demo lo imposta da environment.occApi Prefix — core-libs/core/src/occ/config/default-occ-config.ts:L9-L12, projects/storefrontapp/src/app/private/private.providers.ts:L22-L25	core-libs/core/src/occ/config/occ-config.ts:L16-L16	core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L46-L46
authentication.use ClientTokens	boolean	true solo con toggle authorization CodeFlowByDefault falso, altrimenti assente (→ false)	core-libs/core/src/auth/user-auth/config/auth-config.ts:L76-L76; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L124-L124	core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L33-L34
authentication.send AuthHeaderOnRevoke	boolean	true solo nel ramo legacy	core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L123-L123	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L112-L113
asm.userIdHttpHeader. enable	boolean	true	feature-libs/asm/root/config/asm-config.ts:L39-L45; feature-libs/asm/root/config/default-asm-config.ts:L77-L79	feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L49-L49
OCC_HTTP_TOKEN.send UserIdAsHeader (HttpContextToken)	string | boolean	{}	core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20	feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L53-L59
personalization. enabled / .httpHeader Name.{id,timestamp}	boolean / string	enabled: false; header **DA VERIFICARE** (default non letto)	feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L10-L11	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L35-L46
opps.couponcodes.http HeaderName	string	**DA VERIFICARE** (default non letto)	—	integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L35-L36
segmentRefs.http HeaderName	string	**DA VERIFICARE** (default non letto)	—	integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L57-L58
Angular XSRF_ENABLED / XSRF_COOKIE_NAME / XSRF_HEADER_NAME	token	true / XSRF-TOKEN / X-XSRF-TOKEN	angular:packages/common/http/src/xsrf.ts:L24-L45	angular:packages/common/http/src/xsrf.ts:L99-L119
Angular REQUESTS_ CONTRIBUTE_TO_ STABILITY	boolean	true	angular:packages/common/http/src/interceptor.ts:L217-L221	angular:packages/common/http/src/backend.ts:L61-L61, angular:packages/common/http/src/interceptor.ts:L244-L244
provideHttpClient(...) features (app)	Http Feature[]	demo: withFetch(), with InterceptorsFromDi()	angular:packages/common/http/src/provider.ts:L35-L43	projects/storefrontapp/src/app/app.config.ts:L28-L28
provideClient Hydration(withNoHttp TransferCache())	feature	demo: transfer cache HTTP disattivata	angular:packages/platform-browser/src/hydration.ts:L255-L266	projects/storefrontapp/src/app/app.config.ts:L29-L29

7. Commenti e note notevoli dal sorgente

- `HttpErrorHandlerInterceptor`: "CAUTION: It MUST be provided as the first one in the application to be able to catch errors from all subsequent interceptors." — `core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L37-L38`. Nella catena funzionale lo precede comunque `xsrfInterceptorFn` (§9, T15).
- `BaseCoreModule`: "Import this module before any other interceptor to handle HTTP errors efficiently" — `core-libs/core/src/base-core.module.ts:L28-L28`.
- `AuthModule`: "ClientAuthModule should always be imported after UserAuthModule because the ClientTokenInterceptor must be imported after the AuthInterceptor. This way, the ClientTokenInterceptor is the first to handle 401 errors..." — `core-libs/core/src/auth/auth.module.ts:L13-L15`.
- `ClientAuthModule`: "This pattern should not be used in the frontend apps, but until OCC changes this requirement we provide this module... After OCC improvements regarding client authentication this module can be safely removed." — `core-libs/core/src/auth/client-auth/client-auth.module.ts:L12-L18`.
- `TokenRevocationInterceptor`: "This interceptor is dedicated for Hybris OAuth server which requires `Authorization` header for revoke token calls." — `core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L19-L21`.
- `HttpTimeoutInterceptor`: "It starts counting time for timeout only after the request is sent." — `core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L35-L39`.
- `HttpErrorHandlerInterceptor.shouldHandleError`: "we avoid sending unpredictable errors to the browser's console, to prevent possibly exposing there potentially confidential user's data." — `core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L63-L71`.
- `AsmRootModule`: "The providers below preserve backward compatibility for the CXSPA-12514 fix and should be removed once the enableExpiredRefreshTokenHandlers feature toggle is fully rolled out." — `feature-libs/asm/root/asm-root.module.ts:L77-L79`.
- `OccPersonalizationTimeInterceptor`: commento finale `// CHECK SONAR` — `feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L101-L101`.
- `HttpParamsURIEncoder`: "Angular uses the native encodeURIComponent function, but then un-encoded some characters that are allowed to be part of the query according to IETF RFC 3986." — `core-libs/core/src/util/http-params-uri.encoder.ts:L7-L23`.
- i18next: "It uses Angular HttpClient under the hood, so it works in SSR." — `core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L26-L26`.
- Angular `withInterceptorsFromDi`: "Prefer `withInterceptors` and functional interceptors instead, as support for DI-provided interceptors may be phased out in a later release." — `angular:packages/common/http/src/provider.ts:L173-L174`.
- Angular `HttpClientModule`: `@deprecated use provideHttpClient(withInterceptorsFromDi()) as providers instead` — `angular:packages/common/http/src/module.ts:L94-L94`.
- Angular `HttpInterceptor` JSDoc: importare `HttpClientModule` più volte (anche nei lazy module) "creates a new copy ... which overwrites the interceptors provided in the root module" — `angular:packages/common/http/src/interceptor.ts:L46-L63`.
- Angular `HttpClient.request`: "This way, the handler runs inside an Observable chain, which causes interceptors to be re-run on every subscription" — `angular:packages/common/http/src/client.ts:L857-L863`.
- Angular `REQUESTS_CONTRIBUTE_TO_STABILITY`: "TODO(atscott): We need a larger discussion about stability and what should contribute to stability." — `angular:packages/common/http/src/interceptor.ts:L214-L216`.
- Angular `FetchBackend`: "TODO: Replace with AbortSignal.any([aborter.signal, AbortSignal.timeout(request.timeout)])" — `angular:packages/common/http/src/fetch.ts:L77-L77`.

- Angular `walkProviderTree`: "This allows us to process providers of injector types after all imports of an injector definition are processed. (following View Engine semantics: see FW-1349)" — `angular:packages/core/src/di/provider_collection.ts:L224-L231`.

8. Extension point

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`HTTP_INTERCEPTORS` (Angular)	Nuovo interceptor di classe: `{ provide: HTTP_INTERCEPTORS, useExisting/useClass, multi: true }` in un modulo importato **staticamente** dall'albero root (§3.5, T9). La posizione nella catena dipende da dove si importa il modulo	`angular:packages/common/http/src/interceptor.ts:L187-L197`	50 — Ricette: config, DI, interceptor, eventi
`withInterceptors` / `HttpInterceptorFn` (Angular)	Interceptor funzionale in `provideHttpClient(...)`: gira **fuori** dalla sotto-catena legacy, prima o dopo `LEGACY_INTERCEPTOR_FN` a seconda dell'ordine delle feature	`angular:packages/common/http/src/provider.ts:L150-L163`, `angular:packages/common/http/src/provider.ts:L118-L139`	50
`HttpContextToken` (Angular)	Canale laterale tipizzato verso gli interceptor	`angular:packages/common/http/src/context.ts:L16-L18`	50
`HTTP_TIMEOUT_CONFIG`	Timeout per singola richiesta (`new HttpContext().set(HTTP_TIMEOUT_CONFIG, {server, browser})`)	`core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L18`	52 — Ricette: OCC, adapter, converter
`backend.timeout`	Timeout globale via `provideConfig`	`core-libs/core/src/http/http-timeout/http-timeout.config.ts:L23-L41`	50
`OCC_HTTP_TOKEN` / `OccHttpContext`	Chiedere a ASM di inviare `sap-commerce-cloud-user-id` da un adapter custom	`core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20`	52
`InterceptorUtil`, `USE_CLIENT_TOKEN`, `USE_CUSTOMER_SUPPORT_AGENT_TOKEN`, `USE_CAPTCHA_TOKEN`	Marcare una richiesta di un adapter custom (es. client token)	`core-libs/core/src/occ/utils/interceptor-util.ts:L9-L47`	52
`HttpErrorHandlerInterceptor` (`shouldHandleError`, `handleError` protected)	Estendere e rimpiazzare via DI (registrato con `useExisting`)	`core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L41-L81`	50, 41
`HttpTimeoutInterceptor` (`getTimeoutValue`, `convertTimeoutToHttpErrorResponse`, `buildError` protected)	Estendere e rimpiazzare via DI	`core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L77-L124`	50

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`HttpErrorHandler` (multi)	Gestire uno status HTTP con un global message custom	`core-libs/core/src/global-message/http-interceptors/index.ts:L22-L63`	41, 50
`AuthHttpHeaderService`	Sostituibile via DI (ASM lo fa con `AsmAuthHttpHeaderService`)	`core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L175-L259`; override `feature-libs/asm/root/asm-root.module.ts:L73-L76`	22, 50
`WithCredentialsInterceptor.requiresWithCredentials`	Metodo protected da ridefinire	`core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L43-L48`	50
Interceptor core `providedIn: 'root'` registrati con `useExisting` (`AuthInterceptor`, `TokenRevocationInterceptor`, `ClientTokenInterceptor`, `AnonymousConsentsInterceptor`, `SiteContextInterceptor`)	`{ provide: AuthInterceptor, useClass: MyAuthInterceptor }`: la registrazione `useExisting` risolve la sostituta **mantenendo la posizione** nella catena	A06, tab. 2	50, 54 — Upgrade-safe
`I18NEXT_HTTP_BACKEND_CLIENT`	Sostituire il client HTTP delle traduzioni	`core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L16-L44`	53 — Ricette: feature, stili, traduzioni
`HttpParamsURIEncoder`	Codec per `HttpParams` custom coerente con `buildUrl`	`core-libs/core/src/util/http-params-uri.encoder.ts:L16-L32`	52

Superficie pubblica: `HTTP_TIMEOUT_CONFIG`, `OCC_HTTP_TOKEN`, `InterceptorUtil` e `HttpErrorHandler` sono esportati da `@spartacus/core` (`core-libs/core/src/occ/utils/index.ts:L13-L16`, `core-libs/core/src/global-message/index.ts:L10-L11`). Sono esportati anche `AuthInterceptor`, `TokenRevocationInterceptor`, `ClientTokenInterceptor`, `AnonymousConsentsInterceptor`, `HttpTimeoutInterceptor`, `WithCredentialsInterceptor`, `SiteContextInterceptor` e `HttpErrorHandlerInterceptor` (`core-libs/core/src/auth/user-auth/index.ts:L13-L14`, `core-libs/core/src/auth/client-auth/index.ts:L8-L8`, `core-libs/core/src/anonymous-consents/index.ts:L12-L12`, `core-libs/core/src/http/http-timeout/index.ts:L9-L9`, `core-libs/core/src/occ/interceptors/index.ts:L7-L7`, `core-libs/core/src/occ/adapters/site-context/index.ts:L10-L10`, `core-libs/core/src/error-handling/http-error-handler/index.ts:L7-L7`). `HttpErrorInterceptor` (global message) invece non è esportato: il barrel lo importa soltanto per usarlo in `httpErrorInterceptors` (`core-libs/core/src/global-message/http-interceptors/index.ts:L20-L20`); per personalizzare il global message si usano gli `HttpErrorHandler`. Lo sono anche `HttpParamsURIEncoder` (`core-libs/core/src/util/index.ts:L14-L14`), `I18NEXT_HTTP_BACKEND_CLIENT` (`core-libs/core/src/i18n/i18next/i18next-backend/index.ts:L10-L10`) e `AuthHttpHeaderService` (`core-libs/core/src/auth/user-auth/services/index.ts:L10-L10`), che arrivano a `core-libs/core/public_api.ts:L11-L35` tramite i barrel. Le ricette della Parte III ricontrollano ciascun simbolo nei `public_api`.

Il punto delicato per gli interceptor custom è la posizione. Un modulo con `providers` statici che registra `HTTP_INTERCEPTORS` e viene importato in `SpartacusFeaturesModule` finisce nel punto in cui è importato. Se invece lo si espone come `forRoot()` con `providers`, i suoi provider sono differiti alla fine degli import del modulo padre (§3.5). Per mettere un interceptor di classe *prima* di tutti quelli Spartacus basta registrarlo in `appConfig.providers` prima di `importProvidersFrom(AppModule)`, oppure in un modulo con provider statici im-

portato da `AppModule` prima di `SpartacusModule`: i multi-provider vengono accodati nell'ordine di processamento — angular:packages/core/src/application/create_application.ts:L66-L71, angular:packages/core/src/di/r3_injector.ts:L224-L226, angular:packages/core/src/di/r3_injector.ts:L508-L512, projects/storefrontapp/src/app/app.config.ts:L27-L33. Per farlo girare prima dell'intera sotto-catena legacy serve invece `withInterceptors([...])` passato prima di `withInterceptorsFromDi()` (§3.4, punto 1). Come si comporti questo caso in combinazione con gli `HttpErrorHandler` è DA VERIFICARE: non c'è codice di riferimento nel repo.

---

9. Trappole, casi limite, comportamenti controintuitivi

- **T1 — L'ordine testuale degli import non è l'ordine della catena.** I provider dei `forRoot()` sono differiti a fine import del modulo padre, quelli statici di `@NgModule` sono emessi subito. Per questo `SiteContextInterceptor` (#2, statico) precede `HttpErrorInterceptor` (#4, `GlobalMessageModule.forRoot()`) pur essendo importato dopo, e `AnonymousConsentsInterceptor` (importato a L214) finisce dopo `CmsTicketInterceptor` (L272) — angular:packages/core/src/di/provider_collection.ts:L304-L321, core-libs/core/src/base-core.module.ts:L35-L40, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L214-L272.
- **T2 — Perché il commento di `AuthModule` è vero.** Entrambi gli interceptor sono provider di `forRoot()` differiti nella stessa lista, processati nell'ordine di import — core-libs/core/src/auth/auth.module.ts:L13-L16, angular:packages/core/src/di/provider_collection.ts:L195-L209. Se i provider di `ClientAuthModule` diventassero statici (e quelli di `UserAuthModule` restassero in `forRoot()`), `ClientTokenInterceptor` verrebbe emesso durante la visita, prima di `AuthInterceptor`, e l'ordine si invertirebbe; il caso opposto (solo `UserAuthModule` statico) lascerebbe l'ordine invariato — angular:packages/core/src/di/provider_collection.ts:L316-L321, angular:packages/core/src/di/provider_collection.ts:L346-L353, core-libs/core/src/auth/auth.module.ts:L16-L16.
- **T3 — I timeout non producono global message.** `HttpErrorInterceptor` (#4) sta *dentro* `HttpTimeoutInterceptor` (#3): il `TimeoutError` convertito in #3 risale solo verso #2 e #1. Lo vede `HttpErrorHandlerInterceptor` (#1), non il global message — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L65, core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L34-L42. È una deduzione dall'ordine di §3.6, che a sua volta è DA VERIFICARE a runtime.
- **T4 — I retry vedono solo la parte a valle.** `AuthInterceptor` e `ClientTokenInterceptor` ritentano con il proprio `next`: #1–#6 non rivedono la richiesta, che però conserva params e header già applicati — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L252-L258, core-libs/core/src/auth/client-auth/services/client-error-handling.service.ts:L37-L37. Il retry resta annidato nello stesso observable di #3 e usa lo stesso `timeoutValue`. Il timer avviato dal primo `Sent` corre anche durante il refresh del token, e il `Sent` della richiesta ritentata lo fa ripartire: il tempo totale può superare `timeoutValue`, e un refresh lento può far scattare il timeout prima del retry — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L56, angular:packages/common/http/src/fetch.ts:L117-L118.
- **T5 — Il client token vince sull'utente.** `ClientTokenInterceptor` (#9) viene dopo `AuthInterceptor` (#7). Se è attivo, per le richieste marcate `cx-use-client-token` sovrascrive con `setHeaders` (cioè `headers.set`) l'eventuale `Authorization` utente — core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L57-L67, angular:packages/common/http/src/request.ts:L767-L772.
- **T6 — Il marcatore sopravvive al retry utente.** La richiesta passata a `handleExpiredAccessToken` è quella alterata da #7, quindi precedente a #9. `ClientTokenInterceptor` rimuove il marcatore solo su un proprio clone, perciò la richiesta ritentata lo conserva e ripassa correttamente da #9 — core-libs/core/src/

auth/user-auth/http-interceptors/auth.interceptor.ts:L44-L64, core-libs/core/src/occ/utils/interceptor-util.ts:L29-L35.

- **T7 — Un interceptor OAuth "dormiente".** `DefaultOAuthInterceptor` di angular-oauth2-oidc è nella catena (#6) ma inerte, perché `OAuthModuleConfig` è `null`. Fornire un `OAuthModuleConfig` con `resourceServer.sendAccessToken` aggiungerebbe un secondo meccanismo di `Authorization` — deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L3075-L3095.
- **T8 — Costruttori eseguiti alla prima richiesta.** Gli interceptor di classe sono istanziati tutti insieme alla prima richiesta HTTP (lazy `inject(HTTP_INTERCEPTORS)`). La logica nei costruttori gira in quel momento: subscribe di `SiteContextInterceptor`, lettura di coupon e URL in `OccOppsCouponCodesInterceptor`/`OccSegmentRefsInterceptor`, `toSignal` in `HttpErrorHandlerInterceptor` — angular:packages/common/http/src/interceptor.ts:L230-L241, integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L28-L38, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L46-L46. `DA VERIFICARE`: quale sia la prima richiesta HTTP rispetto ai location initializer (per esempio il caricamento dei base site in un `CONFIG_INITIALIZER`) non è stato verificato.
- **T9 — Gli interceptor nei moduli lazy vengono ignorati.** I moduli lazy di Spartacus sono istanziati con `moduleFactory.create(parentInjector)` — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L97. `LEGACY_INTERCEPTOR_FN` fa `inject(HTTP_INTERCEPTORS)` nel contesto dell'`EnvironmentInjector` di `HttpInterceptorHandler`, cioè quello root in cui è chiamato `provideHttpClient` — angular:packages/common/http/src/backend.ts:L63-L66, angular:packages/common/http/src/backend.ts:L118-L121, angular:packages/common/http/src/interceptor.ts:L174-L185. Per questo tutti gli interceptor di feature stanno nei moduli `*-root` statici.
- **T10 — Due** `SiteContextInterceptor`**.** Esistono due classi omonime (core e asm/customer-360), entrambe nella catena (#2 e #14) con filtri URL disgiunti — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L58-L58, feature-libs/asm/customer-360/root/interceptors/site-context.interceptor.ts:L58-L63. Attenzione agli import quando si fa override via DI.
- **T11 — Leggere il context lo modifica.** `HttpContext.get()` scrive il default nella mappa, e il context è condiviso tra i clone: leggere `OCC_HTTP_TOKEN` o `HTTP_TIMEOUT_CONFIG` in un interceptor muta il context della richiesta originale — angular:packages/common/http/src/context.ts:L78-L83, angular:packages/common/http/src/request.ts:L763-L764.
- **T12 — Due codifiche nella stessa URL.** `SiteContextInterceptor` aggiunge `lang`/`curr` via `setParams`, con il codec Angular di default; la query costruita da `buildUrl` usa invece `HttpParamsURIEncoder` — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L59-L64, core-libs/core/src/occ/services/occ-endpoints.service.ts:L126-L145, angular:packages/common/http/src/params.ts:L114-L119.
- **T13 —** `useWithCredentials: false` **attiva le credenziali.** Il controllo è `!== undefined`, quindi anche `false` porta a `withCredentials: true` — core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L45-L46.
- **T14 —** `withCredentials` **sovrascrive** `credentials`**.** Con `FetchBackend`, `withCredentials: true` diventa `credentials: 'include'` e sovrascrive un `credentials` esplicito — angular:packages/common/http/src/fetch.ts:L325-L332.
- **T15 — "Primo" non significa primo in assoluto.** `HttpErrorHandlerInterceptor` è il primo di `HTTP_INTERCEPTORS`, ma nella catena funzionale lo precede `xsrfInterceptorFn` (e in SSR `relativeUrlsTransformerInterceptorFn` sta in fondo). Errori sincroni in `xsrfInterceptorFn` non passerebbero da lui — angular:packages/common/http/src/provider.ts:L128-L136, angular:packages/common/http/src/backend.ts:L110-L112.
- **T16 — Un workaround non più necessario.** Il commento i18next "Angular Universal doesn't support relative URLs in HttpClient" è superato in Angular 21 da `relativeUrlsTransformerInterceptorFn`, ma Sparta-

cus continua a rendere assoluto il path — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L73-L92, angular:packages/platform-server/src/http.ts:L72-L101.

- **T17 — Config letta una sola volta.** `ClientTokenInterceptor` legge `useClientTokens` come field initializer, alla costruzione: cambi di config successivi non hanno effetto — core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L33-L34.
- **T18 — `useClass` crea un'istanza in più.** `OpfGiftCardPaymentApiInterceptor` usa `useClass` e non è `providedIn: 'root'`. Il modulo lazy `OpfGiftCardModule` re-importa `OpfGiftCardRootModule`, quindi nell'injector figlio nasce un'altra istanza, che la catena root comunque ignora (T9) — integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L57-L61, integration-libs/opf/gift-card/opf-gift-card.module.ts:L18-L18.
- **T19 — Senza `Sent` niente timeout.** `HttpTimeoutInterceptor` inizia a contare solo all'evento `Sent`. `FetchBackend` e `HttpXhrBackend` lo emettono, ma un backend custom o di test che non lo emette disattiverebbe il timeout — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L53-L56, angular:packages/common/http/src/fetch.ts:L118-L118, angular:packages/common/http/src/xhr.ts:L418-L418.
- **T20 — Due pending task per richiesta.** Ogni richiesta registra due `PendingTasks` (handler e sotto-catena legacy), e la stabilità SSR li attende entrambi — angular:packages/common/http/src/backend.ts:L125-L129, angular:packages/common/http/src/interceptor.ts:L243-L247.
- **T21 — Iniettare `HttpBackend` salta tutto.** Un servizio che inietta `HttpBackend` e costruisce un `new HttpClient(backend)` bypassa l'intera catena: niente `lang`/`curr`, niente `Authorization`, niente timeout — angular:packages/common/http/src/backend.ts:L35-L48, angular:packages/common/http/src/client.ts:L132-L134.

---

10. Come fare debug

1. **Stampare la catena reale.** In un componente o in un initializer dell'app, `inject(HTTP_INTERCEPTORS).map(i => i.constructor.name)` restituisce l'array nell'ordine di esecuzione (§3.3) e permette di confermare la tabella di §3.6. Attenzione: questa chiamata istanzia subito tutti gli interceptor (T8). In produzione i nomi delle classi possono essere minificati; conviene farlo con una build di sviluppo.
2. **Verificare le feature di `HttpClient`.** `inject(HttpBackend) instanceof FetchBackend` conferma `withFetch()` (§3.4). In dev mode, su server, l'assenza di `withFetch()` produce il warning `NOT_USING_FETCH_BACKEND_IN_SSR` — angular:packages/common/http/src/backend.ts:L70-L97.
3. **Header e params in DevTools.** Nella scheda Network, una richiesta OCC dell'app demo dovrebbe mostrare `lang`/`curr` nella query (#2), `Authorization` se l'utente è loggato (#7), `X-Anonymous-Consents` (#16) e, se è attivo ASM e l'adapter ha impostato `OCC_HTTP_TOKEN`, `sap-commerce-cloud-user-id` (#13). L'assenza di `cx-use-client-token` è normale: #9 lo rimuove sempre.
4. **Timeout in SSR.** Un warning `Request to URL '...' exceeded expected time of ...ms and was aborted.` nei log del server viene da `HttpTimeoutInterceptor.buildError` — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L113-L124. Per una sola richiesta si può alzare il limite con `HTTP_TIMEOUT_CONFIG` (§3.7).
5. **Errori in SSR.** Gli errori HTTP lato server arrivano all'`ErrorHandler` come `OutboundHttpError`/`CmsPageNotFoundOutboundHttpError` (#1) — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L76-L80. Nel browser #1 non fa nulla: per vedere l'errore servono il global message (#4) o un breakpoint.
6. **Breakpoint utili.** `legacyInterceptorFnFactory` alla riga `inject(HTTP_INTERCEPTORS ...)` — angular:packages/common/http/src/interceptor.ts:L232-L232 — si ferma una sola volta, con l'array

completo. `adaptLegacyInterceptorToChain` — angular:packages/common/http/src/interceptor.ts:L164-L167 — si ferma a ogni passaggio di classe.

7. **Interceptor custom che "non parte".** Controllare che il modulo che lo registra sia importato staticamente dall'albero root e non da un modulo lazy (T9), e che l'app usi `withInterceptorsFromDi()` (§3.4).
8. **Elenco completo.** Per sapere quali classi esistono e dove sono registrate, l'appendice A06 è rigenerabile allo SHA corrente con `python3 tools/extract2/a06_interceptors.py`.

---

11. Domande di autoverifica

1. **Perché gli interceptor di Spartacus non funzionano con un semplice `provideHttpClient()`?** Perché sono tutti registrati su `HTTP_INTERCEPTORS`. Senza `withInterceptorsFromDi()` non esiste `LEGACY_INTERCEPTOR_FN` in `HTTP_INTERCEPTOR_FNS`, e nessuno inietta quel token: `provideHttpClient()` registra soltanto `xsrfInterceptorFn` — angular:packages/common/http/src/provider.ts:L118-L139, angular:packages/common/http/src/provider.ts:L186-L196, A06.
2. **Data la lista `HTTP_INTERCEPTORS = [a, b, c]`, chi vede per primo la richiesta e chi per primo l'errore?** La richiesta la vede per primo `a`, l'errore `c`. La catena è costruita con `reduceRight` ("conceptually `c(b(a(end)))`" in costruzione) per ottenere un'esecuzione da sinistra a destra — angular:packages/common/http/src/interceptor.ts:L233-L240, angular:packages/common/http/src/interceptor.ts:L160-L168.
3. **Perché `AnonymousConsentsInterceptor` è l'ultimo della catena dell'app demo, anche se `AnonymousConsentsModule.forRoot()` è importato prima dei feature module?** Perché i `providers` dei `ModuleWithProviders` sono emessi solo dopo che tutti gli import del modulo padre (`SpartacusFeaturesModule`) sono stati visitati, mentre i root module delle feature hanno provider statici emessi subito — angular:packages/core/src/di/provider_collection.ts:L304-L321, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L214-L214, feature-libs/smartedit/root/smart-edit-root.module.ts:L22-L24.
4. **`BaseCoreModule.forRoot()` è un `forRoot`: perché i suoi interceptor non vengono differiti?** Perché restituisce solo `{ngModule: BaseCoreModule}` senza `providers`, e `walkProviderTree` differisce solo i container con `providers !== undefined` — core-libs/core/src/base-core.module.ts:L47-L52, angular:packages/core/src/di/provider_collection.ts:L360-L362.
5. **Un `TimeoutError` in SSR genera un global message?** Secondo l'ordine ricostruito, no: `HttpTimeoutInterceptor` (#3) converte l'errore e lo propaga verso #2 e #1, mentre `HttpErrorInterceptor` (#4) sta più all'interno. Lo vede `HttpErrorHandlerInterceptor`, che in SSR lo passa all'`ErrorHandler` — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L65, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L52-L71.
6. **Dopo un 401 con token scaduto, la richiesta ritentata ripassa da `SiteContextInterceptor`?** No. `handleExpiredAccessToken` usa il `next` ricevuto da `AuthInterceptor`, che contiene solo gli interceptor a valle (#8 in poi) e il backend; i param `lang`/`curr` restano perché l'oggetto richiesta li contiene già — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L247-L259, angular:packages/common/http/src/interceptor.ts:L164-L167.
7. **Come fa un adapter a chiedere a ASM di inviare l'id del cliente emulato?** Passando `{ context: new HttpContext().set(OCC_HTTP_TOKEN, { sendUserIdAsHeader: true }) }`. `UserIdHttpHeaderInterceptor` legge il token e aggiunge `sap-commerce-cloud-user-id` se `asm.userIdHttpHeader.enable` è vero — core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L51-L56, feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L31-L83.

8. **Perché un interceptor registrato in un modulo caricato con lazy loading non intercetta nulla?** Perché la sotto-catena legacy fa `inject(HTTP_INTERCEPTORS)` nell'`EnvironmentInjector` di `HttpInterceptorHandler`, che è l'injector root. Il modulo lazy vive in un injector figlio creato da `moduleFactory.create(parentInjector)` — angular:packages/common/http/src/backend.ts:L63-L66, angular:packages/common/http/src/interceptor.ts:L174-L185, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L97.
9. **Con `backend.occ.useWithCredentials: false` le richieste OCC partono senza credenziali?** No. Il controllo è `useWithCredentials !== undefined`, quindi `false` attiva `withCredentials: true`; con `FetchBackend` questo diventa `credentials: 'include'` — core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L45-L46, angular:packages/common/http/src/fetch.ts:L325-L332.
10. **Leggere un `HttpContextToken` in un interceptor può avere effetti collaterali?** Sì. `HttpContext.get()` scrive il valore di default nella mappa se il token manca, e il context è condiviso tra la richiesta e i suoi clone — angular:packages/common/http/src/context.ts:L78-L83, angular:packages/common/http/src/request.ts:L763-L764.

TAVOLA 06.1 **4.1 Costruzione lazy della catena (prima richiesta)**

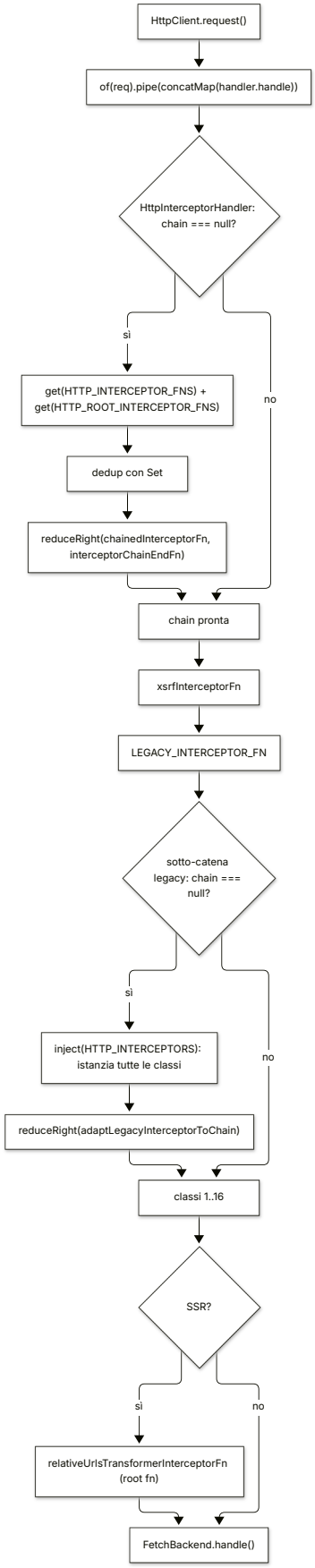
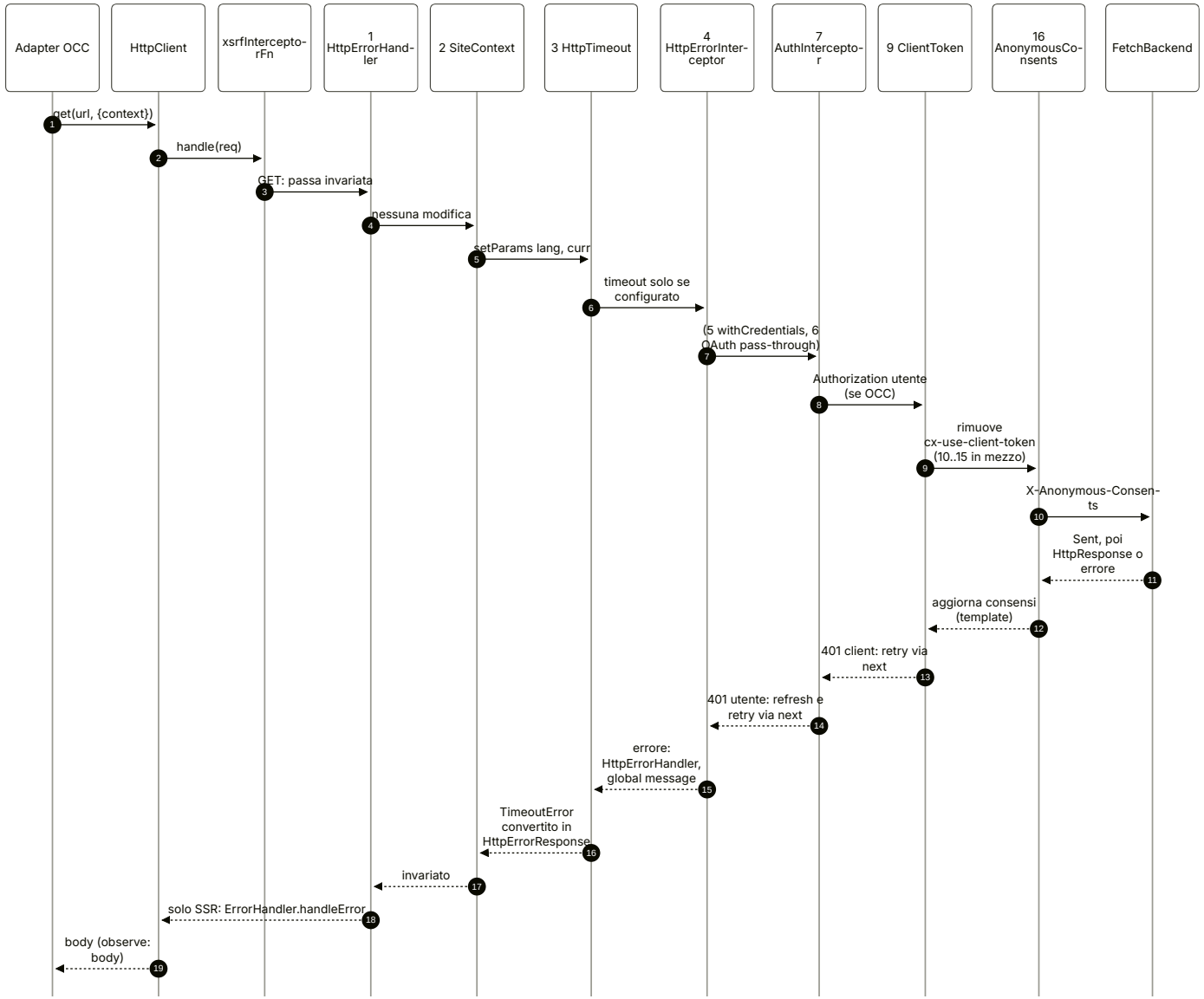


TAVOLA 06.2 **4.3 Percorso di una richiesta OCC (browser, app demo)**



07

SSR, hydration e TransferState

Render su Node, hydration nel browser, TransferState. Le scelte di Spartacus: `CxCommonEngine` e lo state NgRx trasferito.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Angular @ `v21.2.23` (prefisso `angular:`); `@angular/ssr` 21.2.23 e NgRx 21.0.1 dai pacchetti npm pubblicati (prefisso `deps:`, path relativi a `/tmp/deps`).

---

1. A cosa serve e perché esiste

Uno storefront e-commerce deve restituire HTML già popolato alla prima richiesta: per i crawler (SEO), per il tempo al primo contenuto visibile e per i client lenti. Angular risolve il problema con tre meccanismi che questo capitolo smonta pezzo per pezzo:

1. **Server-Side Rendering (SSR)**: l'applicazione viene avviata in Node su una *platform server* con un DOM emulato (Domino), si attende che diventi *stabile* e poi il DOM viene serializzato in stringa (`angular:packages/platform-server/src/utils.ts:L342-L380`, `angular:packages/platform-server/src/platform_state.ts:L39-L56`).
2. **Hydration**: nel browser Angular non ricostruisce da zero il DOM ricevuto dal server, lo "reclama" nodo per nodo grazie alle annotazioni (`ngh`, `__nghData__`) scritte durante il render server (`angular:packages/core/src/hydration/annotate.ts:L243-L301`, `angular:packages/core/src/hydration/api.ts:L211-L228`).
3. **TransferState**: un dizionario chiave→valore JSON che il server serializza in `<script id="ng-state" type="application/json">` e che il browser rilegge al bootstrap, per non ripetere lavoro già fatto (`angular:packages/platform-server/src/transfer_state.ts:L70-L111`, `angular:packages/core/src/transfer_state.ts:L66-L79`).

Spartacus si appoggia a tutti e tre, ma con scelte proprie che è fondamentale conoscere prima di toccare qualunque cosa in area SSR:

- usa ancora il motore `CommonEngine` di `@angular/ssr/node`, avvolto in `CxCommonEngine` e in un `ngExpressEngine` copiato dal vecchio `@nguniversal/express-engine` (`core-libs/setup/ssr/engine/cx-common-engine.ts:L7-L23`, `core-libs/setup/ssr/engine/ng-express-engine.ts:L68-L81`);
- trasferisce lo **state NgRx** (non le risposte HTTP) tramite un meta-reducer che scrive in `TransferState` sotto la chiave `cx-state` (`core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16`), e per questo **disattiva la HTTP transfer cache** di Angular con `withNoHttpTransferCache()` (`projects/storefrontapp/src/app/app.config.ts:L29-L29`, `docs/migration/221121_7/migration.md:L207-L209`);
- espone l'URL e l'origin della richiesta tramite i token `SERVER_REQUEST_URL` / `SERVER_REQUEST_ORIGIN` (`core-libs/core/src/util/ssr.tokens.ts:L9-L21`), consumati da `WindowRef` (`core-libs/core/src/window/window-ref.ts:L19-L27`);
- trasferisce piccoli stati di direttive come attributi DOM `data-cx-state-transfer-*` (`core-libs/storefront/utils/directive-state-transfer.service.ts:L10-L42`).

Questo capitolo copre il **meccanismo** (Angular + integrazione Spartacus). L'ottimizzazione del rendering (timeout, cache, concorrenza, fallback CSR di `OptimizedSsrEngine`), il logger e la gestione errori Express sono trattati in 39 — SSR ottimizzato. I concetti di injector platform/environment usati qui sono in 01 — Dependency Injection, `bootstrapApplication` in 02 — NgModule, standalone e bootstrap, `PendingTasks` e Zone in 04 — Change detection, Zone, zoneless e signals.

2. Mappa dei file

PATH	RUOLO
angular:packages/platform-server/src/utils.ts	renderApplication, renderModule, renderInternal, prepareForHydration, validazione allowedHosts, distruzione asincrona della platform
angular:packages/platform-server/src/server.ts	provider platform/app del server, platformServer, _document (Domino)
angular:packages/platform-server/src/provide_server.ts	provideServerRendering()
angular:packages/platform-server/src/tokens.ts	PlatformConfig, INITIAL_CONFIG, BEFORE_APP_SERIALIZED
angular:packages/platform-server/src/transfer_state.ts	serializzazione di TransferState in <script id="{appId}-state">
angular:packages/platform-server/src/http.ts	ServerXhr (deprecated), interceptor che rende assoluti gli URL relativi
angular:packages/core/src/transfer_state.ts	makeStateKey, TransferState, retrieveTransferredState
angular:packages/core/src/application/application_ref.ts	isStable / whenStable() basati su PendingTasksInternal
angular:packages/core/src/pending_tasks.ts	PendingTasks (add, run)
angular:packages/core/src/hydration/api.ts	withDomHydration, withIncrementalHydration, cleanup viste disidratate
angular:packages/core/src/hydration/annotate.ts	annotazione server (ngh, NGH_DATA_KEY)
angular:packages/core/src/hydration/event_replay.ts	withEventReplay (core)
angular:packages/core/src/hydration/skip_hydration.ts	attributo ngSkipHydration
angular:packages/core/src/hydration/utils.ts	chiavi __nghData__, marker nghm, verifySsrContentsIntegrity
angular:packages/platform-browser/src/hydration.ts	provideClientHydration e feature (withEventReplay, withNoHttpTransferCache, …)
angular:packages/common/http/src/transfer_cache.ts	HTTP transfer cache
deps:angular-ssr-21.2.23/package/fesm2022/node.mjs	CommonEngine (e la classe AngularNodeAppEngine, non usata da Spartacus)
core-libs/setup/ssr/engine/cx-common-engine.ts	CxCommonEngine: propaga al server il primo errore dell'app
core-libs/setup/ssr/engine/ng-express-engine.ts	ngExpressEngine: view engine Express
core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts	NgExpressEngineDecorator: aggiunge provider Spartacus e OptimizedSsrEngine
core-libs/setup/ssr/providers/ssr-providers.ts	provideServer(), getServerRequestProviders()
core-libs/setup/ssr/providers/server-request-origin.ts, server-request-url.ts, model.ts	factory di SERVER_REQUEST_ORIGIN / SERVER_REQUEST_URL, ServerOptions
core-libs/setup/ssr/tokens/express.tokens.ts	token REQUEST / RESPONSE (Express)
core-libs/setup/ssr/express-utils/express-request-origin.ts, express-request-url.ts	origin/URL dalla request Express

PATH	RUOLO
core-libs/core/src/util/ssr.tokens.ts	SERVER_REQUEST_URL, SERVER_REQUEST_ORIGIN
core-libs/core/src/window/window-ref.ts	WindowRef: location emulata in SSR
core-libs/core/src/state/reducers/transfer-state.reducer.ts	CX_KEY, meta-reducer server/browser
core-libs/core/src/state/reducers/index.ts	TRANSFER_STATE_META_REDUCER, registrazione in META_REDUCERS
core-libs/core/src/state/config/state-config.ts	StateConfig.state.ssrTransfer.keys, StateTransferType
core-libs/core/src/state/utils/get-state-slice.ts	estrazione dello slice da trasferire
core-libs/storefront/utils/directive-state-transfer.service.ts	DirectiveStateTransferService
core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts	unico consumatore del servizio sopra
core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts	PendingTasks intorno all'import() dei componenti CMS lazy
core-libs/storefront/shared/components/carousel/carousel.component.ts	esempio di ngSkipHydration
projects/storefrontapp/src/server.ts	server Express della demo
projects/storefrontapp/src/main.server.ts, main.ts	bootstrap server (con BootstrapContext) e client
projects/storefrontapp/src/app/app.config.ts, app.config.server.ts, app.module.server.ts	hydration, config server, provideServer

3. Come funziona passo per passo

3.1 La platform server e il render di Angular

Creazione della platform. renderApplication(bootstrap, options) valida l'URL contro allowedHosts, crea la platform server, chiama la funzione di bootstrap passando {platformRef}, attende applicationRef.whenStable() e solo allora serializza; nel finally distrugge la platform (angular:packages/platform-server/src/utils.ts:L342-L380). renderModule è lo stesso schema con platformRef.bootstrapModule(moduleType) (angular:packages/platform-server/src/utils.ts:L283-L309).

createServerPlatform passa come provider di **livello platform** INITIAL_CONFIG = {document, url} più gli eventuali platformProviders (angular:packages/platform-server/src/utils.ts:L55-L74). PlatformConfig ha solo i campi document? e url? (angular:packages/platform-server/src/tokens.ts:L16-L34).

I provider di platform interni registrano DOCUMENT (factory _document), PLATFORM_ID = PLATFORM_SERVER_ID, l'inizializzatore dell'adapter Domino, ServerPlatformLocation e PlatformState (angular:packages/platform-server/src/server.ts:L47-L57). _document() legge INITIAL_CONFIG e, se document è una stringa, la parsa con Domino usando config.url risolto contro http://localhost (angular:packages/platform-server/src/server.ts:L95-L118). platformServer() imposta globalThis.ngServerMode = true se non definito e lo azzera alla distruzione (angular:packages/platform-server/src/server.ts:L130-L149).

Provider applicativi. `provideServerRendering()` restituisce `PLATFORM_SERVER_PROVIDERS` come `EnvironmentProviders` (angular:packages/platform-server/src/provide_server.ts:L28-L34); questi includono `TRANSFER_STATE_SERIALIZATION_PROVIDERS`, il plugin eventi server, `SERVER_HTTP_PROVIDERS` e `NullViewportScroller` (angular:packages/platform-server/src/server.ts:L75-L82). Tra i provider HTTP c'è un interceptor root che trasforma gli URL relativi in assoluti usando `PlatformLocation` (angular:packages/platform-server/src/http.ts:L72-L110); il vecchio backend `ServerXhr` è `@deprecated` a favore di `withFetch()` (angular:packages/platform-server/src/http.ts:L22-L29), che è infatti ciò che usa la demo (projects/storefrontapp/src/app/app.config.ts:L28-L28).

Bootstrap con `BootstrapContext`. `bootstrapApplication(root, options, context?)` accetta un `BootstrapContext` con `platformRef`, necessario in SSR dove la platform è creata per ogni richiesta (angular:packages/platform-browser/src/browser.ts:L58-L63, angular:packages/platform-browser/src/browser.ts:L125-L140). La demo esporta proprio questa funzione da `main.server.ts` (projects/storefrontapp/src/main.server.ts:L23-L26).

Stabilità. `ApplicationRef.isStable` è derivato da `PendingTasksInternal.hasPendingTasksObservable` (stabile = nessun task pendente) e `whenStable()` risolve alla prima emissione `true` (angular:packages/core/src/application/application_ref.ts:L326-L350). Chi contribuisce ai task pendenti:

- **Zone.js**: `ZoneStablePendingTask` aggiunge un task su `onUnstable` e lo rimuove dopo `onStable` quando non restano macro/microtask (angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L231-L281). La demo usa `provideZoneChangeDetection({ eventCoalescing: true })` (projects/storefrontapp/src/app/app.config.ts:L30-L30).
- **HttpClient**: ogni richiesta che passa da `HttpInterceptorHandler` aggiunge un `PendingTasks.add()` rimosso con `finalize`, perché `REQUESTS_CONTRIBUTE_TO_STABILITY` vale `true` di default (angular:packages/common/http/src/backend.ts:L125-L134, angular:packages/common/http/src/interceptor.ts:L217-L221); gli interceptor a classi (`withInterceptorsFromDi()`) ne aggiungono uno anche per il proprio tratto di catena (angular:packages/common/http/src/interceptor.ts:L243-L247).
- **Codice applicativo**: `PendingTasks.add()` restituisce una funzione di cleanup idempotente (angular:packages/core/src/pending_tasks.ts:L42-L61); la JSDoc cita esplicitamente SSR (angular:packages/core/src/pending_tasks.ts:L24-L26). Spartacus lo usa in `LazyComponentHandler` (§3.6).

Serializzazione. `renderInternal` esegue nell'ordine: `prepareForHydration`, `appendServerContextInfo`, i callback `BEFORE_APP_SERIALIZED` (sincroni e asincroni, con `Promise.allSettled`), infine `platformState.renderToString()` (angular:packages/platform-server/src/utils.ts:L197-L234). Gli errori dei callback vanno a `INTERNAL_APPLICATION_ERROR_HANDLER` (angular:packages/platform-server/src/utils.ts:L207-L221). `appendServerContextInfo` scrive `ng-server-context` sugli host bootstrappati con il valore di `SERVER_CONTEXT` (default `'other'`) (angular:packages/platform-server/src/utils.ts:L145-L155, angular:packages/platform-server/src/utils.ts:L249-L269).

Distruzione. La platform è distrutta in un `setTimeout(…, 0)`, per lasciare che le promise pendenti si risolvano (angular:packages/platform-server/src/utils.ts:L236-L247).

3.2 `CommonEngine` di `@angular/ssr`

`@angular/ssr` non è nel repo Angular: `packages/ssr/docs/BUILD.bazel` dichiara che si trova nel repo Angular/CLI (angular:packages/ssr/docs/BUILD.bazel:L16-L16). Il codice qui descritto proviene dal pacchetto npm pubblicato 21.2.23.

1. **Costruttore.** `allowedHosts` = variabile d'ambiente letta con `getArrayFromEnv('NG_ALLOWED_HOSTS')` se presente, altrimenti `options.allowedHosts`, altrimenti insieme vuoto (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L112, deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L10-L12). Se `Zone`

non è definito, aggancia una volta per processo handler globali `unhandledRejection`/`uncaughtException` (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L31-L41`).

2. **Validazione host.** Se `url` è assoluto, `render` ne valida l'hostname (`'*'`, match esatto o wildcard `*.dominio`, `deps:angular-ssr-21.2.23/package/fesm2022/_validation-chunk.mjs:L72-L86`). Se la validazione fallisce **e non c'è alcun host configurato**, stampa un errore e **restituisce il template HTML non renderizzato** (fallback CSR); se invece ci sono host configurati rilancia l'errore (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L113-L136`).
3. **Pagina SSG.** Se `publicPath`, `documentFilePath` e `url` sono definiti, cerca `publicPath/<pathname>/index.html` e lo serve solo se contiene il marker `ng-server-context="...ssg..."` (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L101-L101`, `deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L155-L184`).
4. **Render.** Altrimenti chiama il proprio `renderApplication` e, salvo `inlineCriticalCss === false`, inlinea il CSS critico (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L137-L150`).
5. **Provider.** Il `renderApplication` interno costruisce `[{SERVER_CONTEXT: 'ssr'}, ...opts.providers, ...this.options.providers]`, legge il documento (con cache per path) e chiama `renderApplication` di platform-server con questi provider come `platformProviders` (se `bootstrap` è una funzione) o `renderModule` con `extraProviders` (se è un NgModule), sempre con `allowedHosts: ['*']` (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L185-L219`, `deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L228-L230`). ɵSERVER_CONTEXT è l'export privato di `SERVER_CONTEXT` (`angular:packages/platform-server/src/private_export.ts:L13-L13`): per questo l'HTML SSR ha `ng-server-context="ssr"`.

Conseguenza: in SSR **tutti** i provider passati all'engine (di setup e di render) vivono nel **platform injector**, non nell'environment injector dell'app. È il motivo per cui i factory di `provideServer` leggono i valori Express con `skipSelf` (§3.4).

`@angular/ssr/node` espone anche la classe AngularNodeAppEngine, che delega ad AngularAppEngine e converte la request Node in una `Request` web (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L290-L307`); Spartacus non lo usa (nessuna occorrenza in `core-libs`/`projects`).

3.3 La catena di engine di Spartacus

`ngExpressEngine` è codice copiato da `@nguniversal/express-engine`, rimosso in Angular 17, mantenuto per retrocompatibilità (`core-libs/setup/ssr/engine/ng-express-engine.ts:L68-L81`). Alla chiamata crea **una sola** istanza di `CxCommonEngine` con `bootstrap`, `providers`, `enablePerformanceProfiler`, `allowedHosts` e restituisce la funzione view-engine Express `(filePath, options, callback)` (`core-libs/setup/ssr/engine/ng-express-engine.ts:L82-L94`). Per ogni render:

- se manca `bootstrap` sia nel setup sia nelle render options lancia `Error('You must pass in a NgModule to be bootstrapped')` (`core-libs/setup/ssr/engine/ng-express-engine.ts:L95-L99`);
- calcola `url` da `req.protocol`, header `host`, `req.baseUrl`, `req.url`; `documentFilePath = filePath`; `publicPath` da render options / setup / `settings.views` (`core-libs/setup/ssr/engine/ng-express-engine.ts:L101-L118`);
- aggiunge ai provider `REQUEST` (sempre) e `RESPONSE` (se presente, fallback `req.res`) (`core-libs/setup/ssr/engine/ng-express-engine.ts:L38-L53`, `core-libs/setup/ssr/engine/ng-express-engine.ts:L102-L112`);
- passa l'esito al callback Express: `callback(null, html)` oppure `callback(err)` (`core-libs/setup/ssr/engine/ng-express-engine.ts:L120-L126`).

`REQUEST` e `RESPONSE` di Spartacus sono `InjectionToken<Request>`/`<Response>` di **Express** (`core-libs/setup/ssr/tokens/express.tokens.ts:L10-L26`): da non confondere con `REQUEST` di `@angular/core` (web `Request`, `providedIn: 'platform'`, factory `null`, `angular:packages/core/src/application/platform_tokens.ts:L31-L37`).

`CxCommonEngine` estende `CommonEngine` (core-libs/setup/ssr/engine/cx-common-engine.ts:L7-L23) e ridefinisce `render` anteponendo un provider `PROPAGATE_ERROR_TO_SERVER` che memorizza solo il **primo** errore (`??=`); a rendering terminato, se un errore è stato catturato, rigetta la promise (core-libs/setup/ssr/engine/cx-common-engine.ts:L41-L66). L'errore viene alimentato da `PropagatingToServerErrorHandler`, registrato come `MULTI_ERROR_HANDLER` da `provideServer` (core-libs/setup/ssr/providers/ssr-providers.ts:L41-L45), che chiama il token o un no-op se assente (core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts:L26-L38). Il token esiste "finché Angular non implementa una propagazione corretta degli errori asincroni" (core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L9-L22).

`NgExpressEngineDecorator.get(engine, optimizationOptions)` delega a `decorateExpressEngine` (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L36-L49), che antepone `getServerRequestProviders()` ai provider di setup e, se `optimizationOptions` è truthy (default `defaultSsrOptimizationOptions`), avvolge l'istanza in `OptimizedSsrEngine` (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L74). I dettagli di `OptimizedSsrEngine` sono nel capitolo 39.

3.4 Token di richiesta: `SERVER_REQUEST_ORIGIN` e `SERVER_REQUEST_URL`

I due token sono `InjectionToken<string>` di `@spartacus/core`, senza factory (core-libs/core/src/util/ssr.tokens.ts:L9-L21). Vengono forniti a **due livelli**:

1. **Platform** (solo SSR con Express): `getServerRequestProviders()` li dichiara come `StaticProvider` con `deps: [REQUEST]` e factory `getRequestOrigin`/`getRequestUrl` (core-libs/setup/ssr/providers/ssr-providers.ts:L48-L66). `getRequestOrigin` usa `X-Forwarded-Host` (primo valore) solo se la `trust proxy fn` di Express si fida del `remoteAddress`, altrimenti l'header `host` (core-libs/setup/ssr/express-utils/express-request-origin.ts:L9-L26); `getRequestUrl` = origin + `req.originalUrl` (core-libs/setup/ssr/express-utils/express-request-url.ts:L10-L12). Per §3.2 questi provider finiscono nel platform injector.
2. **App** (SSR e prerendering): `provideServer(options?)` fornisce i due token con factory basate su `inject`, più `LoggerService` e `PropagatingToServerErrorHandler` (core-libs/setup/ssr/providers/ssr-providers.ts:L24-L47). La demo lo registra in `AppServerModule` con `serverRequestOrigin: process.env['SERVER_REQUEST_ORIGIN']` (projects/storefrontapp/src/app/app.module.server.ts:L10-L17).

`serverRequestOriginFactory` legge il valore platform con `inject(SERVER_REQUEST_ORIGIN, {optional: true, skipSelf: true})` e restituisce, nell'ordine, `options.serverRequestOrigin`, il valore Express, oppure lancia un errore esplicativo (core-libs/setup/ssr/providers/server-request-origin.ts:L14-L55). L'opzione è facoltativa in SSR ma **obbligatoria per il prerendering**, e se impostata ha precedenza su Express (core-libs/setup/ssr/providers/model.ts:L10-L23).

`serverRequestUrlFactory` inietta `INITIAL_CONFIG`, `SERVER_REQUEST_ORIGIN` (senza `skipSelf`) e `SERVER_REQUEST_URL` del parent (`skipSelf`); in SSR restituisce l'URL Express (con sostituzione dell'origin se `options.serverRequestOrigin` è impostato), nel prerendering costruisce `origin + pathname` da `INITIAL_CONFIG.url` (core-libs/setup/ssr/providers/server-request-url.ts:L15-L49). Il commento avverte che in Angular recente `platformConfig.url` può essere un URL completo, da cui l'estrazione del solo `pathname` (core-libs/setup/ssr/providers/server-request-url.ts:L37-L47).

Il consumatore è `WindowRef`, che inietta i due token `@Optional()` (core-libs/core/src/window/window-ref.ts:L19-L27): in SSR `location` restituisce un `Partial<Location>` con **solo** `href` e `origin`, lanciando un errore se i token mancano (core-libs/core/src/window/window-ref.ts:L62-L89); `nativeWindow` è `undefined` (core-libs/core/src/window/window-ref.ts:L37-L44) e `resize$` emette `of(null)` (core-libs/core/src/window/window-ref.ts:L91-L107).

Anche il logger dipende dal contesto: `serverLoggerServiceFactory` sceglie `ExpressLoggerService` se `REQUEST` è iniettabile, altrimenti `PrerenderingLoggerService` (core-libs/setup/ssr/logger/services/server-logger-

service-factory.ts:L12-L17).

3.5 TransferState: da Angular allo state NgRx di Spartacus

Angular. `makeStateKey<T>(key)` è un semplice cast di stringa a un tipo brandizzato (angular:packages/core/src/transfer_state.ts:L28-L49). `TransferState` è `providedIn: 'root'`; nel browser la factory popola lo store con `retrieveTransferredState(DOCUMENT, APP_ID)`, sul server parte vuoto (angular:packages/core/src/transfer_state.ts:L66-L79). L'API è `get` (default se `undefined`), `set`, `remove`, `hasKey`, `isEmpty`, `onSerialize` (angular:packages/core/src/transfer_state.ts:L89-L126). La serializzazione è `JSON.stringify` con `<` e `/` escapati (angular:packages/core/src/transfer_state.ts:L131-L147), quindi solo boolean, number, string, null e oggetti non-classe sopravvivono senza perdite (angular:packages/core/src/transfer_state.ts:L60-L62).

Sul server la serializzazione è un provider `multi` su `BEFORE_APP_SERIALIZED` (angular:packages/platform-server/src/transfer_state.ts:L31-L37): se lo store non è vuoto appende in fondo al `<body>` `<script id="{appId}-state" type="application/json">` (angular:packages/platform-server/src/transfer_state.ts:L70-L111). Con `APP_ID` di default `'ng'` lo script è `ng-state` (angular:packages/core/src/application/application_tokens.ts:L44-L52). Nel browser `retrieveTransferredState` cerca quell'elemento e fa `JSON.parse`, con warning e `{}` in caso di errore (angular:packages/core/src/transfer_state.ts:L150-L169).

Spartacus. La chiave è `CX_KEY = makeStateKey<string>('cx-state')` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16). Il flusso:

1. `BaseCoreModule` importa `StateModule.forRoot()` (core-libs/core/src/base-core.module.ts:L29-L29), che fornisce `stateMetaReducers` (core-libs/core/src/state/state.module.ts:L10-L18): `TRANSFER_STATE_META_REDUCER` con factory `getTransferStateReducer` (deps `PLATFORM_ID`, `TransferState`/`Config`/`AuthStatePersistenceService` opzionali), aggiunto a `META_REDUCERS` di NgRx con `useExisting`, `multi: true` (core-libs/core/src/state/reducers/index.ts:L23-L43).
2. In NgRx i `META_REDUCERS` vengono concatenati ai meta-reducer di `forRoot` e composti attorno al reducer factory (`combineReducers`) per formare il token NgRx REDUCER_FACTORY (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1337-L1355, deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L359-L372); il ReducerManager di NgRx ricompone tutto lo state root con quel factory a ogni aggiunta di feature (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L425-L444). Il meta-reducer di Spartacus vede quindi **l'intero state**.
3. Le feature dichiarano quali slice trasferire con `state.ssrTransfer.keys` (core-libs/core/src/state/config/state-config.ts:L20-L39), valore `StateTransferType.TRANSFER_STATE = 'SSR'` (core-libs/core/src/state/config/state-config.ts:L16-L18). OOTB lo fanno quattro store: site-context, site-theme, cms, product (core-libs/core/src/site-context/store/site-context-store.module.ts:L20-L30, core-libs/core/src/site-theme/store/site-theme-store.module.ts:L20-L29, core-libs/core/src/cms/store/cms-store.module.ts:L21-L31, core-libs/core/src/product/store/product-store.module.ts:L20-L30).
4. `getTransferStateReducer` restituisce il meta-reducer browser o server in base a `PLATFORM_ID`, solo se esistono `TransferState` e le chiavi; altrimenti un meta-reducer identità (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L18-L40).
5. **Server**: dopo **ogni** azione, estrae lo slice delle chiavi `SSR` dal nuovo state con `getStateSlice` e lo scrive in `TransferState` sotto `cx-state`, sovrascrivendo (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L42-L62). `filterKeysByType` seleziona le chiavi per tipo (core-libs/core/src/state/utils/get-state-slice.ts:L104-L112); `getStateSlice` supporta la notazione a punti e ricompone lo slice con `deepMerge` (core-libs/core/src/state/utils/get-state-slice.ts:L10-L55), scartando i valori vuoti (core-libs/core/src/state/utils/get-state-slice.ts:L23-L36).
6. **Server, fine render**: `BEFORE_APP_SERIALIZED` scrive l'intero store in `#ng-state` (passo §3.1).

7. **Browser**: il meta-reducer interviene solo sull'azione `INIT` di NgRx: calcola lo state iniziale se assente e, **se l'utente non risulta loggato** e `cx-state` esiste, fa `deepMerge({}, state, slice)`; per tutte le altre azioni delega al reducer (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L64-L96). "Loggato" significa presenza di `token.access_token` nello storage (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L119-L121).

In NgRx `INIT` è il valore iniziale del `BehaviorSubject` `ActionsSubject`, mentre l'aggiunta di feature emette `UPDATE` (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L196-L200, deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L391-L391, deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L438-L444). `combineReducers` passa a ogni reducer registrato `state[key]` e restituisce lo **stesso oggetto** (con eventuali chiavi extra) se nessun reducer cambia riferimento, altrimenti un nuovo oggetto con le sole chiavi registrate (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L319-L343). Ne consegue che una feature registrata dopo `INIT` riceve lo slice trasferito come stato precedente **solo se** nessun reducer registrato ha cambiato riferimento tra `INIT` e il suo `UPDATE`; altrimenti `combineReducers` ha già scartato le chiavi non registrate e la feature parte dal proprio `initialState` (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L329-L341, core-libs/core/src/state/reducers/transfer-state.reducer.ts:L89-L93). `DA VERIFICARE` a runtime quali feature siano registrate all'`INIT` e quali azioni le precedano (dipende dall'ordine di istanziazione DI, non osservato).

`CX_KEY`, `TRANSFER_STATE_META_REDUCER` e `getTransferStateReducer` **non** sono nel public API: `core-libs/core/src/state/index.ts` non riesporta `reducers/`, e `getStateSlice` è esposto solo tramite il namespace `StateUtils` (core-libs/core/src/state/index.ts:L7-L11, core-libs/core/src/state/utils/index.ts:L7-L8, core-libs/core/src/state/utils/utils-group.ts:L10-L10).

3.6 Stabilità in Spartacus: `PendingTasks` e caricamento istantaneo

- `LazyComponentHandler` avvolge l'`import()` dinamico dei componenti CMS lazy in `PendingTasks.add()` + `finalize`, per impedire che SSR serializzi e distrugga l'app con l'import in volo (NG0205) (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L31-L31, core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71).
- `DeferLoaderService` carica sempre istantaneamente sul server (core-libs/storefront/layout/loading/defer-loader.service.ts:L55-L64): nessun lazy rendering "on viewport" in SSR.
- `CmsComponentsService.shouldRender` esclude dal render SSR i componenti la cui mapping ha `disableSSR` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L187-L190, core-libs/core/src/cms/config/cms-config.ts:L75-L75).

Questi punti, insieme a Zone e HttpClient (§3.1), determinano quando `whenStable()` risolve e quindi quando l'HTML viene serializzato.

3.7 Hydration lato server

1. `provideClientHydration(...features)` è nell'`appConfig` condiviso fra client e server (projects/storefrontapp/src/app/app.config.ts:L29-L29, projects/storefrontapp/src/app/app.config.server.ts:L17-L30). Restituisce `withDomHydration()` + (HTTP transfer cache solo se non disattivata né configurata) + i provider delle feature (angular:packages/platform-browser/src/hydration.ts:L225-L266).
2. Sul server `IS_HYDRATION_DOM_REUSE_ENABLED` è sempre `true` (angular:packages/core/src/hydration/api.ts:L211-L228), quindi `prepareForHydration` aggiunge il marker `nghm` come primo figlio di `<body>`, chiama `annotateForHydration` e inserisce lo script di replay eventi se ci sono tipi di evento da riprodurre (angular:packages/platform-server/src/utils.ts:L95-L125, angular:packages/platform-server/src/utils.ts:L127-L139).

3. `annotateForHydration` annota le viste di `appRef._views` e scrive **sempre** `NGH_DATA_KEY` (`__nghData__`) in `TransferState`, anche con lista vuota, come segnale al client (angular:packages/core/src/hydration/annotate.ts:L243-L301, angular:packages/core/src/hydration/utils.ts:L46-L69). Ogni host annotato riceve `ngh="<indice>"`; componenti i18n non supportati o `ShadowDom` ricevono `ngSkipHydration` (angular:packages/core/src/hydration/annotate.ts:L820-L844).
4. I tipi di evento vengono raccolti solo se `IS_EVENT_REPLAY_ENABLED` (angular:packages/core/src/hydration/annotate.ts:L250-L254); lo script di replay chiama `window.__jsaction_bootstrap(...)` ed è inserito dopo lo script `ng-event-dispatch-contract`, solo se questo esiste — cioè se la build CLI l'ha inlineato (angular:packages/platform-server/src/utils.ts:L36-L44, angular:packages/platform-server/src/utils.ts:L157-L186).

Risultato: hydration (`__nghData__`), eventuale HTTP transfer cache e Spartacus (`cx-state`) condividono **lo stesso** store `TransferState` e **lo stesso** `<script id="ng-state">`.

3.8 Hydration lato browser

1. Il client avvia `bootstrapApplication(AppComponent, appConfig)` (projects/storefrontapp/src/main.ts:L17-L23).
2. `IS_HYDRATION_DOM_REUSE_ENABLED` vale `true` solo se `TransferState` contiene `__nghData__` (angular:packages/core/src/hydration/api.ts:L211-L228).
3. L'initializer verifica il marker di integrità e abilita il runtime di hydration; se le annotazioni mancano, in dev mode emette "MISSING_HYDRATION_ANNOTATIONS" (angular:packages/core/src/hydration/api.ts:L229-L266). `verifySsrContentsIntegrity` lancia `MISSING_SSR_CONTENT_INTEGRITY_MARKER` se non trova `nghm` (angular:packages/core/src/hydration/utils.ts:L766-L810).
4. `PRESERVE_HOST_CONTENT = IS_HYDRATION_DOM_REUSE_ENABLED`: il DOM SSR dell'host non viene svuotato al bootstrap (angular:packages/core/src/hydration/api.ts:L269-L280).
5. I sottoalberi con `ngSkipHydration` (case-insensitive, ereditato dagli antenati, angular:packages/core/src/hydration/skip_hydration.ts:L16-L72) vengono ricreati da zero. In Spartacus: `CarouselComponent` (core-libs/storefront/shared/components/carousel/carousel.component.ts:L57-L65) e 12 componenti di form in feature-libs, es. `LoginFormComponent` (feature-libs/user/account/components/login-form/login-form.component.ts:L36-L36).
6. **Event replay**: `IS_EVENT_REPLAY_ENABLED` nel browser è `true` solo se `window._ejsas[appId]` esiste (angular:packages/core/src/hydration/event_replay.ts:L76-L95); dopo `whenStable()` gli eventi catturati vengono riprodotti e il contratto ripulito (angular:packages/core/src/hydration/event_replay.ts:L132-L197).
7. **Cleanup**: un `APP_BOOTSTRAP_LISTENER` attende `whenStableWithTimeout(appRef)` ed esegue `cleanupDehydratedViews` per le viste SSR non reclamate (angular:packages/core/src/hydration/api.ts:L281-L320); in dev mode avverte se l'app non diventa stabile entro 10 s (angular:packages/core/src/hydration/api.ts:L87-L91).
8. `PageTemplateDirective` recupera la classe di template SSR dall'attributo trasferito (§3.9).

`withIncrementalHydration()` (che include `withEventReplay()` e processa i blocchi `@defer`, angular:packages/core/src/hydration/api.ts:L352-L398) **non** è usato da Spartacus né dalla demo.

3.9 HTTP transfer cache e perché Spartacus la spegne

`withHttpTransferCache` registra `CACHE_OPTIONS`, l'interceptor root `transferCacheInterceptorFn` e un listener che disattiva la cache quando l'app diventa stabile (angular:packages/common/http/src/transfer_cache.ts:L439-L468). Sul server l'interceptor salva in `TransferState` le risposte riuscite cacheabili (di default

GET/HEAD, le POST solo con `includePostRequests` o con opzione per richiesta; escluse le richieste con credenziali o header auth/cookie, con `Cache-Control` `no-store`/`private`/`no-cache` o con `cache` `no-cache`/`no-store`, o scartate dal `filter`; escluse le risposte con quelle direttive o con `Set-Cookie`), nel browser restituisce `of(cachedResponse)` (angular:packages/common/http/src/transfer_cache.ts:L126-L152, angular:packages/common/http/src/transfer_cache.ts:L132-L148, angular:packages/common/http/src/transfer_cache.ts:L333-L333, angular:packages/common/http/src/transfer_cache.ts:L294-L296, angular:packages/common/http/src/transfer_cache.ts:L252-L316).

`withNoHttpTransferCache()` non ha provider: è un flag che disattiva la cache; la sua JSDoc dice che le richieste vengono eseguite due volte, sul server e nel browser (angular:packages/platform-browser/src/hydration.ts:L64-L76). Spartacus lo usa nella demo e la schematic `add-ssr` lo inserisce sempre, con il commento "Http Transfer Cache is temporarily disabled" (ticket CXSPA-10430) (core-libs/schematics/src/add-ssr/update-app-config-in-ssr.ts:L47-L50, core-libs/schematics/src/add-ssr/update-app-config-in-ssr.ts:L138-L149). La guida di migrazione lo motiva con il meccanismo di state transfer proprio di Spartacus (docs/migration/221121_7/migration.md:L207-L209).

3.10 `DirectiveStateTransferService`

Trasferisce **stringhe** come attributi DOM `data-cx-state-transfer-{key}`: `set` e `clear` agiscono solo sul server (via un `Renderer2` creato con `createRenderer(null, null)`), `get` legge l'attributo ovunque (core-libs/storefront/utils/directive-state-transfer.service.ts:L17-L42). L'unico consumatore è `PageTemplateDirective` con chiave `'page-template'` (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L46-L51): inizializza `currentTemplate` dall'attributo SSR, così dopo l'hydration può rimuovere la classe del template precedente (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L83-L87), e in `addStyleClass` persiste il template corrente (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L114-L129).

3.11 La demo: dal build al server Express

- Build: `@angular-builders/custom-esbuild:application` con `server: main.server.ts`, `ssr.entry: server.ts`, `prerender: false`; configurazioni `noSsr` e `withPrerender` (projects/storefrontapp/project.json:L10-L10, projects/storefrontapp/project.json:L142-L146, projects/storefrontapp/project.json:L177-L186). `withPrerender` usa `prerender.txt`, che elenca 11 route di esempio (projects/storefrontapp/prerender.txt:L1-L11).
- Config server: `mergeApplicationConfig(appConfig, serverConfig)` con `provideServerRendering()`, `AppServerModule` e `TestConfigServerModule.forRoot()` marcato "DO NOT USE IN CUSTOMERS APPS" (projects/storefrontapp/src/app/app.config.server.ts:L17-L30).
- `server.ts`: `SsrOptimizationOptions` da env (`SSR_TIMEOUT`, `SSR_CACHE`), engine decorato con `NgExpressEngineDecorator.get(engine, ssrOptions)` (projects/storefrontapp/src/server.ts:L22-L29); `trust proxy = 'loopback'`, middleware di validazione origin da `SSR_ALLOWED_ORIGINS`, engine `html` con `bootstrap`, statici con `maxAge: '1y'`, tutte le altre rotte con `res.render(indexHtml, { req, providers: [APP_BASE_HREF] })`, error handler di default (projects/storefrontapp/src/server.ts:L32-L80). Il template è `index.server.html` nella cartella server del dist (projects/storefrontapp/src/server.ts:L34-L37).
- La demo **non** passa `allowedHosts` a `ngExpressEngine` (projects/storefrontapp/src/server.ts:L51-L56): si affida alla variabile `NG_ALLOWED_HOSTS=localhost` impostata dagli script npm `serve:ssr*` (package.json:L152-L155) e dagli script di installazione, che esportano `NG_ALLOWED_HOSTS=${SSR_ALLOWED_HOSTS:-localhost}` (`SSR_ALLOWED_HOSTS` vuota di default = `localhost`) (scripts/install/functions.sh:L583-L583, scripts/install/functions.sh:L599-L599, scripts/install/config.default.sh:L74-L77). Vedi §9.

4. Diagrammi

4.1 Richiesta SSR end-to-end (demo)

■ TAVOLA 07.1 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: passi 2-3 projects/storefrontapp/src/server.ts:L39-L75; 4 core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L58-L73; 5 core-libs/setup/ssr/engine/ng-express-engine.ts:L95-L118; 6 core-libs/setup/ssr/engine/ng-express-engine.ts:L120-L121; 7 core-libs/setup/ssr/engine/cx-common-engine.ts:L41-L59; 8-9 deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L113-L219; 10-11 angular:packages/platform-server/src/utils.ts:L342-L362; 12 core-libs/setup/ssr/providers/server-request-origin.ts:L16-L29; 14 angular:packages/platform-server/src/utils.ts:L367-L371; 15 angular:packages/platform-server/src/utils.ts:L197-L234; 19 core-libs/setup/ssr/engine/cx-common-engine.ts:L60-L65; 20 core-libs/setup/ssr/engine/ng-express-engine.ts:L120-L123.

4.2 Decisioni di `CommonEngine.render`

■ TAVOLA 07.2 diagramma a tutta pagina, in fondo al capitolo

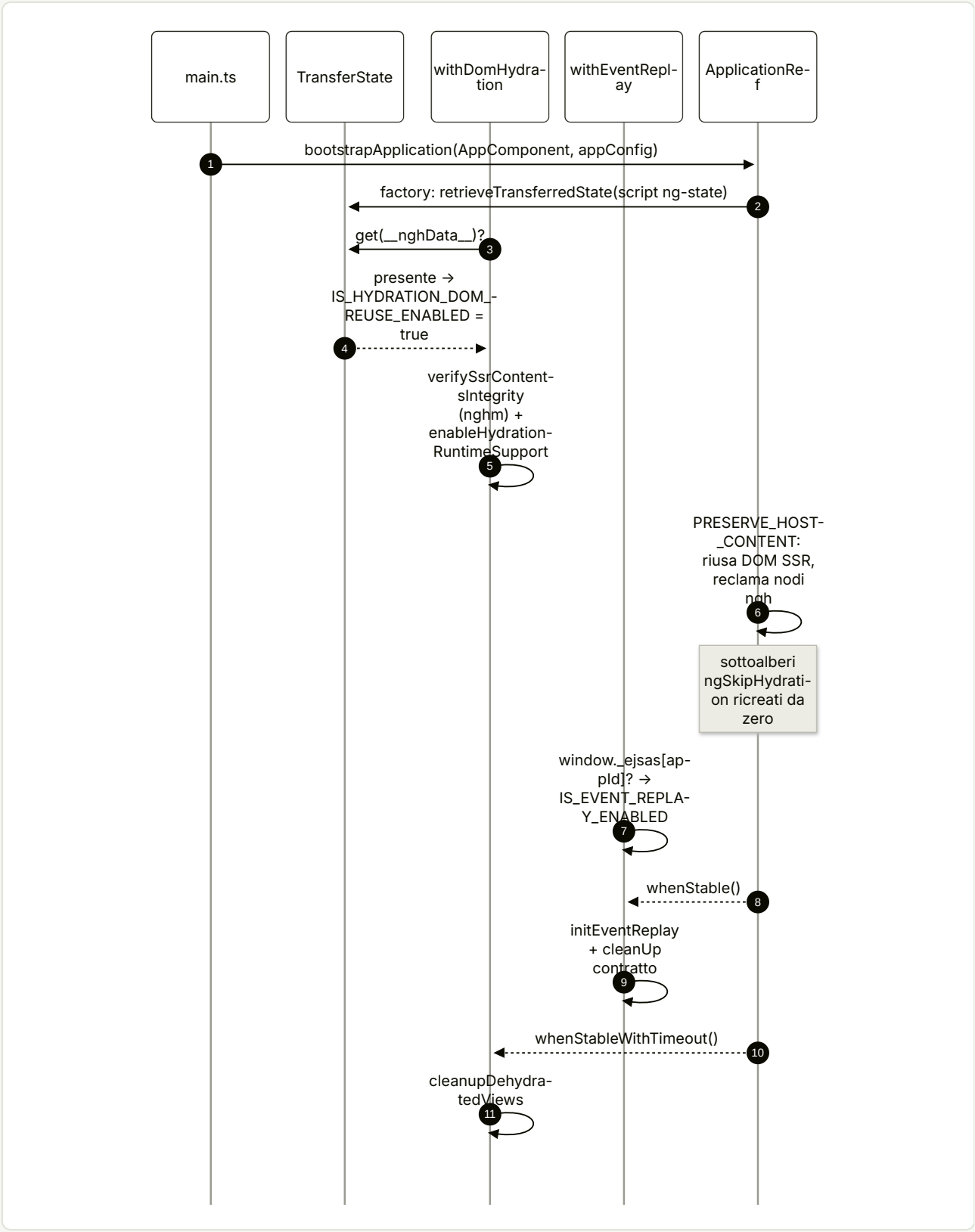
Riferimenti: deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L150, deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L155-L219.

4.3 Transfer dello state NgRx

■ TAVOLA 07.3 diagramma a tutta pagina, in fondo al capitolo

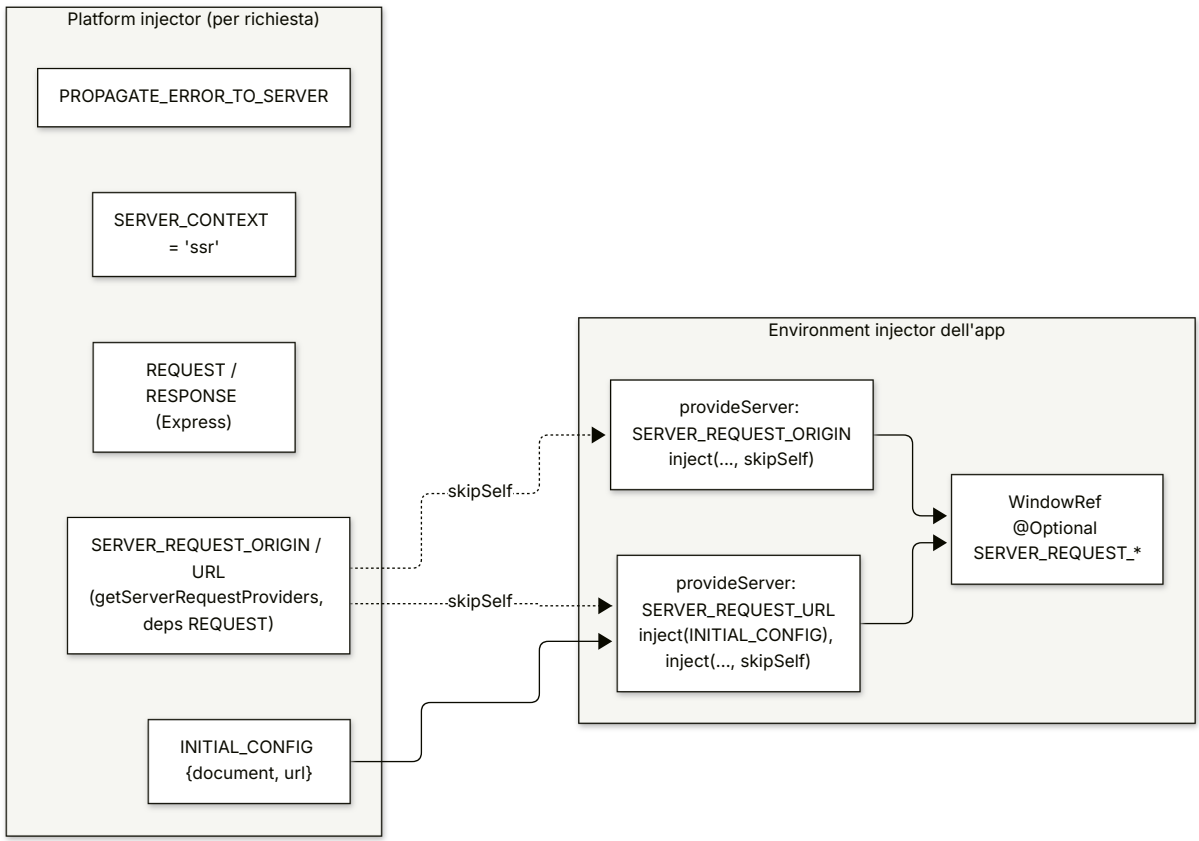
Riferimenti: core-libs/core/src/state/reducers/transfer-state.reducer.ts:L42-L96; angular:packages/platform-server/src/transfer_state.ts:L70-L111; angular:packages/core/src/transfer_state.ts:L66-L79, angular:packages/core/src/transfer_state.ts:L150-L169; core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L119-L121.

4.4 Hydration nel browser



Riferimenti: `projects/storefrontapp/src/main.ts:L17-L23`; `angular:packages/core/src/transfer_state.ts:L66-L79`; `angular:packages/core/src/hydration/api.ts:L211-L320`; `angular:packages/core/src/hydration/event_replay.ts:L76-L95`, `angular:packages/core/src/hydration/event_replay.ts:L132-L197`.

4.5 Livelli DI dei token di richiesta in SSR



Riferimenti: deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L185-L209; angular:packages/platform-server/src/utils.ts:L55-L74; core-libs/setup/ssr/providers/ssr-providers.ts:L24-L66; core-libs/setup/ssr/providers/server-request-origin.ts:L14-L29; core-libs/setup/ssr/providers/server-request-url.ts:L15-L35; core-libs/core/src/window/window-ref.ts:L19-L27.

---

5. Estratti di codice reale

`CommonEngine`: **provider di platform e** `allowedHosts: ['*']` — `deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L185-L209`

```JS
async renderApplication(opts) {
  const moduleOrFactory = this.options?.bootstrap ?? opts.bootstrap;
  if (!moduleOrFactory) {
    throw new Error('A module or bootstrap option must be provided.');
  }
  const extraProviders = [{
    provide: _SERVER_CONTEXT,
    useValue: 'ssr'
  }, ...(opts.providers ?? []), ...(this.options?.providers ?? [])];
  let document = opts.document;
```

```
    if (!document && opts.documentFilePath) {
      document = await this.getDocument(opts.documentFilePath);
    }
    const commonRenderingOptions = {
      url: opts.url,
      document,
      allowedHosts: ['*']
    };
    return isBootstrapFn(moduleOrFactory) ? renderApplication(moduleOrFactory, {
      platformProviders: extraProviders,
      ...commonRenderingOptions
    }) : renderModule(moduleOrFactory, {
      extraProviders,
      ...commonRenderingOptions
    });
  }
```

renderApplication **di platform-server** — angular:packages/platform-server/src/utils.ts:L354-L380

```ts
  const {url, allowedHosts} = options;

  validateAllowedHosts(url, allowedHosts);

  startMeasuring(renderAppLabel);
  const platformRef = createServerPlatform(options);
  try {
    startMeasuring(bootstrapLabel);
    const applicationRef = await bootstrap({platformRef});
    stopMeasuring(bootstrapLabel);
    // ...
    // Block until application is stable.
    await applicationRef.whenStable();
    stopMeasuring(measuringLabel);

    const rendered = await renderInternal(platformRef, applicationRef);
    stopMeasuring(_renderLabel);
    return rendered;
  } finally {
    await asyncDestroyPlatform(platformRef);
    stopMeasuring(renderAppLabel);
  }
}
```

CxCommonEngine.render — core-libs/setup/ssr/engine/cx-common-engine.ts:L41-L66

```ts
  override async render(options: CommonEngineRenderOptions): Promise<string> {
    let error: undefined | unknown;
```

```
  return super
    .render({
      ...options,
      providers: [
        {
          provide: PROPAGATE_ERROR_TO_SERVER,
          useFactory: () => {
            return (propagatedError: unknown) => {
              // We're interested only the first propagated error, so we use `??=` instead of `=`:
              error ??= propagatedError;
            };
          },
        },
        ...(options.providers ?? []),
      ],
    })
    .then((html: string) => {
      if (error) {
        throw error;
      }
      return html;
    });
}
```

ngExpressEngine: **render options per richiesta** — core-libs/setup/ssr/engine/ng-express-engine.ts:L101-L123

```ts
    const { req } = renderOptions;
    const res = renderOptions.res ?? req.res;

    renderOptions.url =
      renderOptions.url ??
      `${req.protocol}://${req.get('host') || ''}${req.baseUrl}${req.url}`;
    renderOptions.documentFilePath =
      renderOptions.documentFilePath ?? filePath;
    renderOptions.providers = [
      ...(renderOptions.providers ?? []),
      getReqResProviders(req, res),
    ];
    renderOptions.publicPath =
      renderOptions.publicPath ??
      setupOptions.publicPath ??
      (options as any).settings?.views;
    renderOptions.inlineCriticalCss =
      renderOptions.inlineCriticalCss ?? setupOptions.inlineCriticalCss;

    engine
```

```
        .render(renderOptions)
        .then((html) => callback(null, html))
        .catch(callback);
```

Provider SSR di Spartacus — core-libs/setup/ssr/providers/ssr-providers.ts:L27-L66

```ts
export function provideServer(options?: ServerOptions): Provider[] {
  return [
    {
      provide: SERVER_REQUEST_ORIGIN,
      useFactory: serverRequestOriginFactory(options),
    },
    {
      provide: SERVER_REQUEST_URL,
      useFactory: serverRequestUrlFactory(options),
    },
    {
      provide: LoggerService,
      useFactory: serverLoggerServiceFactory,
    },
    {
      provide: MULTI_ERROR_HANDLER,
      useExisting: PropagatingToServerErrorHandler,
      multi: true,
    },
  ];
}
// ...
export function getServerRequestProviders(): StaticProvider[] {
  return [
    {
      provide: SERVER_REQUEST_ORIGIN,
      useFactory: getRequestOrigin,
      deps: [REQUEST],
    },
    {
      provide: SERVER_REQUEST_URL,
      useFactory: getRequestUrl,
      deps: [REQUEST],
    },
  ];
}
```

serverRequestUrlFactory — core-libs/setup/ssr/providers/server-request-url.ts:L15-L35

```ts
export function serverRequestUrlFactory(options?: ServerOptions): Function {
  return (): string => {
    const platformConfig = inject(INITIAL_CONFIG);
```

```
    const serverRequestOrigin = inject(SERVER_REQUEST_ORIGIN);
    const serverRequestUrl = inject(SERVER_REQUEST_URL, {
      optional: true,
      skipSelf: true,
    });

    // SSR mode
    if (serverRequestUrl) {
      // should override the automatically recognized origin
      if (options?.serverRequestOrigin) {
        return serverRequestUrl.replace(
          serverRequestOrigin,
          options.serverRequestOrigin
        );
      }

      return serverRequestUrl;
    }
```

Meta-reducer di transfer — core-libs/core/src/state/reducers/transfer-state.reducer.ts:L51-L95

TS

```
  return function (reducer: any) {
    return function (state: any, action: any) {
      const newState = reducer(state, action);
      if (newState) {
        const stateSlice = getStateSlice(transferStateKeys, [], newState);
        transferState.set(CX_KEY, stateSlice);
      }

      return newState;
    };
  };
}
// ...
  return function (reducer: any) {
    return function (state: any, action: any) {
      if (action.type === INIT) {
        if (!state) {
          state = reducer(state, action);
        }

        if (!isLoggedIn && transferState.hasKey(CX_KEY)) {
          const cxKey = transferState.get<Object>(CX_KEY, {});
          const transferredStateSlice = getStateSlice(
            transferStateKeys,
            [],
            cxKey
          );
```

```
          state = deepMerge({}, state, transferredStateSlice);
        }
        return state;
      }
      return reducer(state, action);
    };
  };
```

Registrazione in `META_REDUCERS` — core-libs/core/src/state/reducers/index.ts:L27-L43

```ts
export const stateMetaReducers: Provider[] = [
  {
    provide: TRANSFER_STATE_META_REDUCER,
    useFactory: getTransferStateReducer,
    deps: [
      PLATFORM_ID,
      [new Optional(), TransferState],
      [new Optional(), Config],
      [new Optional(), AuthStatePersistenceService],
    ],
  },
  {
    provide: META_REDUCERS,
    useExisting: TRANSFER_STATE_META_REDUCER,
    multi: true,
  },
];
```

Dichiarazione delle chiavi da trasferire (CMS) — core-libs/core/src/cms/store/cms-store.module.ts:L21-L31

```ts
export function cmsStoreConfigFactory(): StateConfig {
  // if we want to reuse CMS_FEATURE const in config, we have to use factory instead of plain object
  const config: StateConfig = {
    state: {
      ssrTransfer: {
        keys: { [CMS_FEATURE]: StateTransferType.TRANSFER_STATE },
      },
    },
  };
  return config;
}
```

Serializzazione Angular in `#ng-state` — angular:packages/platform-server/src/transfer_state.ts:L81-L110

```ts
    const content = transferStore.toJson();

    if (transferStore.isEmpty) {
      // The state is empty, nothing to transfer,
      // avoid creating an extra `<script>` tag in this case.
      return;
    }
    // ...
    script.id = appId + '-state';
    script.setAttribute('type', 'application/json');

    // It is intentional that we add the script at the very bottom. Angular CLI script tags for
    // bundles are always `type="module"`. These are deferred by default and cause the
    // transfer data to be queried only after the browser has finished parsing the DOM.
    doc.body.appendChild(script);
```

PendingTasks **intorno all'**`import()` **lazy** — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71

```ts
    return defer(() => {
      // Register the dynamic import as a pending task so Angular's stability
      // counter stays > 0 until the chunk is loaded. This prevents SSR from
      // serializing and destroying the app while the import() is still in flight
      // (which would cause NG0205 — destroyed injector). The task is removed by
      // finalize(), which fires on complete, error, and unsubscribe.

      const removeTaskTracking = this.pendingTasks.add();
      return from(componentMapping.component()).pipe(
        finalize(removeTaskTracking)
      );
    }).pipe(
```

DirectiveStateTransferService — core-libs/storefront/utils/directive-state-transfer.service.ts:L17-L42

```ts
@Injectable({ providedIn: 'root' })
export class DirectiveStateTransferService {
  protected rendererFactory = inject(RendererFactory2);
  protected renderer2 = this.rendererFactory.createRenderer(null, null);
  protected windowRef = inject(WindowRef);

  set(el: HTMLElement, key: string, value: string): void {
    if (!this.windowRef.isBrowser()) {
      this.renderer2.setAttribute(el, this.attributeFrom(key), value);
    }
  }

  get(el: HTMLElement, key: string): string | undefined {
```

```
    return el?.getAttribute(this.attributeFrom(key)) ?? undefined;
  }
  // ...
  protected attributeFrom(key: string): string {
    return `data-cx-state-transfer-${key}`;
  }
}
```

Config client della demo — projects/storefrontapp/src/app/app.config.ts:L26-L35

```ts
export const appConfig: ApplicationConfig = {
  providers: [
    provideHttpClient(withFetch(), withInterceptorsFromDi()),
    provideClientHydration(withEventReplay(), withNoHttpTransferCache()),
    provideZoneChangeDetection({ eventCoalescing: true }),
    provideBrowserGlobalErrorListeners(),

    importProvidersFrom(AppModule),
  ],
};
```

Bootstrap server della demo — projects/storefrontapp/src/main.server.ts:L23-L26

```ts
const bootstrap = (context: BootstrapContext) =>
  bootstrapApplication(AppComponent, config, context);

export default bootstrap;
```

Server Express della demo — projects/storefrontapp/src/server.ts:L22-L29, :L51-L75

```ts
const ssrOptions: SsrOptimizationOptions = {
  timeout: Number(
    process.env['SSR_TIMEOUT'] ?? defaultSsrOptimizationOptions.timeout
  ),
  cache: process.env['SSR_CACHE'] === 'true',
};

const ngExpressEngine = NgExpressEngineDecorator.get(engine, ssrOptions);
// ...
  server.engine(
    'html',
    ngExpressEngine({
      bootstrap,
    })
  );
  // ...
  server.get(/.*/, (req, res) => {
```

```
    res.render(indexHtml, {
      req,
      providers: [{ provide: APP_BASE_HREF, useValue: req.baseUrl }],
    });
  });
```

`withDomHydration`: **abilitazione nel browser** — angular:packages/core/src/hydration/api.ts:L211-L228

```ts
export function withDomHydration(): EnvironmentProviders {
  const providers: Provider[] = [
    {
      provide: IS_HYDRATION_DOM_REUSE_ENABLED,
      useFactory: () => {
        let isEnabled = true;
        if (typeof ngServerMode === 'undefined' || !ngServerMode) {
          // On the client, verify that the server response contains
          // hydration annotations. Otherwise, keep hydration disabled.
          const transferState = inject(TransferState, {optional: true});
          isEnabled = !!transferState?.get(NGH_DATA_KEY, null);
        }
        if (isEnabled) {
          performanceMarkFeature('NgHydration');
        }
        return isEnabled;
      },
    },
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`state.ssrTransfer.keys`	`{[key: string]: StateTransferType | undefined}`	nessuno nel tipo; OOTB `siteContext`, `siteTheme`, `cms`, `product` → `'SSR'`	core-libs/core/src/state/config/state-config.ts:L20-L39; default nei `*-store.module.ts` (es. core-libs/core/src/cms/store/cms-store.module.ts:L21-L31)	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L24-L35
`StateTransferType.TRANSFER_STATE`	enum	`'SSR'`	core-libs/core/src/state/config/state-config.ts:L16-L18	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L46-L49, core-libs/core/src/state/reducers/transfer-state.reducer.ts:L69-L72
`ServerOptions.serverRequestOrigin`	`string?`	`undefined` (demo: `process.env.SERVER_REQUEST_ORIGIN`)	core-libs/setup/ssr/providers/model.ts:L10-L23	core-libs/setup/ssr/providers/server-request-origin.ts:L22-L24, core-libs/setup/ssr/providers/server-request-url.ts:L27-L32

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
NgSetupOptions (bootstrap, providers, publicPath, inline CriticalCss, enable PerformanceProfiler, allowedHosts)	Pick<CommonEngineRenderOptions,…> & CommonEngineOptions	—	core-libs/setup/ssr/engine/ng-express-engine.ts:L24-L28	core-libs/setup/ssr/engine/ng-express-engine.ts:L83-L88, core-libs/setup/ssr/engine/ng-express-engine.ts:L113-L118
RenderOptions.req / res / url / documentFilePath	Express / string	url da req; document FilePath = filePath	core-libs/setup/ssr/engine/ng-express-engine.ts:L63-L66	core-libs/setup/ssr/engine/ng-express-engine.ts:L101-L112
allowedHosts di CommonEngine	string[]	env NG_ALLOWED_HOSTS=… > options.allowedHosts > vuoto (= fallback CSR per URL assoluti)	core-libs/setup/ssr/engine/ng-express-engine.ts:L87-L87	deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L136
env NG_ALLOWED_HOSTS=localhost	string (lista separata da virgole)	impostata negli script serve:ssr*; negli script di installazione ${SSR_ALLOWED_HOSTS:-localhost}	package.json:L152-L155; scripts/install/config.default.sh:L74-L77; scripts/install/functions.sh:L583-L583, scripts/install/functions.sh:L599-L599	deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L10-L28
inlineCriticalCss	boolean?	attivo salvo false	core-libs/setup/ssr/engine/ng-express-engine.ts:L117-L118	deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L141-L144
enablePerformanceProfiler	boolean?	off	core-libs/setup/ssr/engine/ng-express-engine.ts:L86-L86	deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L136-L149
optimizationOptions di NgExpressEngineDecorator.get	SsrOptimizationOptions | null	defaultSsrOptimizationOptions; falsy = nessun OptimizedSsrEngine	core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L57	core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L69-L72
env demo SSR_TIMEOUT, SSR_CACHE, SSR_ALLOWED_ORIGINS, SERVER_REQUEST_ORIGIN, PORT	string	timeout di default; cache false; allowlist vuota = no-op; porta 4000	projects/storefrontapp/src/server.ts:L22-L27, projects/storefrontapp/src/server.ts:L45-L49, projects/storefrontapp/src/server.ts:L83-L83; projects/storefrontapp/src/app/app.module.server.ts:L13-L13	idem
INITIAL_CONFIG (document, url)	token Angular	—	angular:packages/platform-server/src/tokens.ts:L16-L34	angular:packages/platform-server/src/server.ts:L95-L118; core-libs/setup/ssr/providers/server-request-url.ts:L17-L17
provideClientHydration(withEventReplay(), withNoHttpTransferCache())	provider	—	projects/storefrontapp/src/app/app.config.ts:L29-L29	angular:packages/platform-browser/src/hydration.ts:L225-L266
initialNavigation: 'enabledBlocking'	ExtraOptions del router	—	core-libs/storefront/router/app-routing.module.ts:L13-L19	angular:packages/router/src/router_module.ts:L237-L244
CmsComponentMapping.disableSSR	boolean?	undefined	core-libs/core/src/cms/config/cms-config.ts:L75-L75	core-libs/storefront/cms-structure/services/cms-components.service.ts:L187-L190
APP_ID	string	'ng' → <script id="ng-state">	angular:packages/core/src/application/application_tokens.ts:L44-L52	angular:packages/core/src/transfer_state.ts:L74-L74

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
build `server`, `ssr.entry`, `prerender`, `noSsr`/`withPrerender`	`project.json`	`prerender: false`	projects/storefrontapp/project.json:L142-L146, projects/storefrontapp/project.json:L177-L186	builder `@angular-builders/custom-esbuild:application` (projects/storefrontapp/project.json:L10-L10)

7. Commenti e note notevoli dal sorgente

- core-libs/setup/ssr/engine/cx-common-engine.ts:L14-L19 — la JSDoc rimanda al sorgente Angular CLI di `CommonEngine` a un commit specifico: Spartacus dipende dalla forma interna di quel motore.
- core-libs/setup/ssr/engine/cx-common-engine.ts:L25-L40 — "will be thrown and the result promise rejected - but only AFTER the rendering is complete ... only the first one will be used".
- core-libs/setup/ssr/engine/ng-express-engine.ts:L68-L81 — `ngExpressEngine` copiato da `angular/universal` (licenza MIT Google) "to avoid larger refactors".
- core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L9-L22 — `PROPAGATE_ERROR_TO_SERVER` fornito OOTB solo in SSR (non prerendering), "until Angular implements a proper propagation of async errors" (angular#33642).
- core-libs/setup/ssr/testing/test-config-server.module.ts:L26-L58 — "CAUTION: DON'T USE IT IN PRODUCTION! IT HASN'T BEEN REVIEWED FOR SECURITY ISSUES."; la demo lo marca "DO NOT USE IN CUSTOMERS APPS" (projects/storefrontapp/src/app/app.config.server.ts:L23-L26).
- core-libs/schematics/src/add-ssr/update-app-config-in-ssr.ts:L47-L50 — "Http Transfer Cache is temporarily disabled; https://jira.tools.sap/browse/CXSPA-10430".
- core-libs/storefront/utils/directive-state-transfer.service.ts:L10-L16 — "useful for when a singleton state key through TransferStateService isn's sufficient" (refuso "isn's"; una classe `TransferStateService` non esiste in Spartacus: il commento si riferisce verosimilmente al `TransferState` di Angular. `DA VERIFICARE`: intenzione dell'autore non documentata).
- core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L61-L65 — motivazione esplicita del pending task (NG0205).
- core-libs/storefront/shared/components/carousel/carousel.component.ts:L57-L59 — hydration disattivata per "inconsistencies between client-side rendering (CSR) and server-side rendering (SSR)".
- core-libs/storefront/router/on-navigate.service.ts:L36-L43 — "with Angular hydration the initial navigation's `Scroll` event is suppressed".
- core-libs/core/src/window/window-ref.ts:L37-L44 — `// TODO(#11133): Consider throwing in SSR.`
- core-libs/setup/ssr/providers/server-request-url.ts:L37-L47 — "platformConfig.url may be a full URL (e.g. 'http://ng-localhost/') in newer Angular versions".
- angular:packages/core/src/hydration/annotate.ts:L283-L287 — `NGH_DATA_KEY` scritto "always ... even if the list of serialized views is empty".
- angular:packages/core/src/hydration/annotate.ts:L444-L456 — fix #67928 (componenti in `LContainer` con `ngSkipHydration`), presente in v21.2.23.
- angular:packages/platform-server/src/http.ts:L22-L29 — `ServerXhr` `@deprecated`: "xhr2 library does not safely handle redirects".
- angular:packages/platform-server/src/transfer_state.ts:L39-L39 — `TODO: Move this to a utils folder and convert to use SafeValues.`
- angular:packages/core/src/transfer_state.ts:L56-L58 — la JSDoc cita ancora `ServerTransferStateModule`, mentre oggi la serializzazione è registrata da

TRANSFER_STATE_SERIALIZATION_PROVIDERS in PLATFORM_SERVER_PROVIDERS (angular:packages/platform-server/src/server.ts:L75-L76).

- deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L123-L123 — "Falling back to client side rendering. This will become a 400 Bad Request in a future major version.": il fallback CSR per host non consentito è dichiarato transitorio.

---

8. Extension point

EXTENSION POINT	COME	DEFINITO	EXPORT
Trasferire lo state di una feature custom	provideDefaultConfigFactory / provideConfig con state.ssrTransfer.keys: { [MY_FEATURE]: StateTransferType.TRANSFER_STATE }	core-libs/core/src/state/config/state-config.ts:L20-L39	@spartacus/core (core-libs/core/src/state/index.ts:L7-L7)
Leggere URL/origin della richiesta	inject(SERVER_REQUEST_URL) / inject(SERVER_REQUEST_ORIGIN) o WindowRef.location	core-libs/core/src/util/ssr.tokens.ts:L9-L21; core-libs/core/src/window/window-ref.ts:L62-L89	@spartacus/core
Origin fisso (prerendering, proxy)	provideServer({ serverRequestOrigin })	core-libs/setup/ssr/providers/ssr-providers.ts:L27-L47; core-libs/setup/ssr/providers/model.ts:L10-L23	@spartacus/setup/ssr
Accedere a request/response Express	inject(REQUEST, { optional: true }) / RESPONSE da @spartacus/setup/ssr	core-libs/setup/ssr/tokens/express.tokens.ts:L10-L26	core-libs/setup/ssr/public_api.ts:L16-L16
Provider extra per ogni render	ngExpressEngine({ bootstrap, providers }) o res.render(..., { providers })	core-libs/setup/ssr/engine/ng-express-engine.ts:L82-L118	core-libs/setup/ssr/public_api.ts:L10-L10
allowedHosts / CSS critico / profiler	opzioni di ngExpressEngine (allowedHosts, inlineCriticalCss, enablePerformanceProfiler) o env NG_ALLOWED_HOSTS=…	core-libs/setup/ssr/engine/ng-express-engine.ts:L24-L28	idem
Engine custom	estendere CxCommonEngine ridefinendo render (chiamando super.render)	core-libs/setup/ssr/engine/cx-common-engine.ts:L20-L67	core-libs/setup/ssr/public_api.ts:L9-L9
Escludere un componente CMS dal render SSR	cmsComponents: { X: { component, disableSSR: true } }	core-libs/core/src/cms/config/cms-config.ts:L75-L75	@spartacus/core
Escludere un componente dall'hydration	host: { ngSkipHydration: 'true' } (come il carousel)	angular:packages/core/src/hydration/skip_hydration.ts:L16-L72	Angular
Trasferire una stringa per una direttiva	DirectiveStateTransferService.set/get (o sottoclasse ridefinendo attributeFrom)	core-libs/storefront/utils/directive-state-transfer.service.ts:L17-L42	core-libs/storefront/public_api.ts:L30-L30

EXTENSION POINT	COME	DEFINITO	EXPORT
Bloccare la serializzazione finché un lavoro asincrono non termina	`inject(PendingTasks).add()` + cleanup (come `LazyComponentHandler`)	angular:packages/core/src/pending_tasks.ts:L42-L61	Angular
Dati custom in `TransferState`	`makeStateKey` + `TransferState.set/get` o `onSerialize`	angular:packages/core/src/transfer_state.ts:L28-L49, angular:packages/core/src/transfer_state.ts:L89-L126	Angular

Non sono estendibili via API pubblica `CX_KEY`, `TRANSFER_STATE_META_REDUCER` e `getTransferStateReducer` (core-libs/core/src/state/index.ts:L7-L11).

Ricette pertinenti in Parte III:

- override di config e provider via DI (es. aggiungere chiavi `ssrTransfer`, sostituire un servizio solo lato server): 50 — Ricette: configurazione, DI, interceptor, eventi;
- componenti CMS con `disableSSR`, direttive che usano `DirectiveStateTransferService`: 51 — Ricette: componenti CMS, outlet, route e layout;
- feature lib custom con store da trasferire: 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni;
- simboli interni da non importare (`CX_KEY`, factory non riesportate): 54 — API pubblica vs interna, upgrade, anti-pattern, checklist.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Senza `NG_ALLOWED_HOSTS=…` la demo non fa SSR.** `ngExpressEngine` costruisce sempre un URL assoluto (core-libs/setup/ssr/engine/ng-express-engine.ts:L104-L106) e la demo non passa `allowedHosts` (projects/storefrontapp/src/server.ts:L51-L56); con insieme vuoto `CommonEngine` logga l'errore e restituisce il template non renderizzato (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L113-L136). Gli script `serve:ssr*` impostano `NG_ALLOWED_HOSTS=localhost` (package.json:L152-L155): in produzione va impostata con i propri host, altrimenti si ottiene CSR silenzioso (solo un `console.error`). Il `validateAllowedHosts` di platform-server non interviene perché `CommonEngine` passa `['*']` (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L198-L202).
2. **Due `REQUEST` omonimi.** Quello di `@spartacus/setup/ssr` è l'oggetto Express (core-libs/setup/ssr/tokens/express.tokens.ts:L15-L17), quello di `@angular/core` è una `Request` web (angular:packages/core/src/application/platform_tokens.ts:L31-L37) e, con `CommonEngine`, nessuno lo fornisce (factory `null`). Spartacus non usa `RESPONSE_INIT` né `REQUEST_CONTEXT`.
3. **Provider dell'engine = platform.** Tutto ciò che passate a `ngExpressEngine`/`res.render` è un provider di platform (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L190-L205): un provider app-level con lo stesso token lo oscura, a meno di usare `skipSelf` come fa `provideServer` (core-libs/setup/ssr/providers/server-request-origin.ts:L16-L19).
4. **Possibile `replace` no-op in `serverRequestUrlFactory`.** `inject(SERVER_REQUEST_ORIGIN)` è senza `skipSelf` (core-libs/setup/ssr/providers/server-request-url.ts:L18-L18), quindi risolve il provider app-level di `provideServer` che, con `options.serverRequestOrigin` impostato, restituisce proprio quel valore (core-libs/setup/ssr/providers/server-request-origin.ts:L22-L24); il `replace(serverRequestOrigin, options.serverRequestOrigin)` (core-libs/setup/ssr/providers/server-request-url.ts:L28-L31) sostituisce la stringa con sé stessa e l'URL resta con l'origin Express. Lo spec mocka `inject` (core-libs/

setup/ssr/providers/server-request-url.spec.ts:L109-L128) e non copre la risoluzione DI reale. DA VERIFICARE a runtime (analisi statica coerente, non eseguito: niente node_modules).

5. SERVER_REQUEST_ORIGIN **da env vince su Express.** La demo passa process.env['SERVER_REQUEST_ORIGIN'] (projects/storefrontapp/src/app/app.module.server.ts:L13-L13): se la variabile è impostata anche in SSR, l'origin è fisso per tutte le richieste (core-libs/setup/ssr/providers/model.ts:L17-L17).
6. **Header** Host **e proxy.** getRequestOrigin si fida di X-Forwarded-Host solo se trust proxy fn approva il remoteAddress (core-libs/setup/ssr/express-utils/express-request-origin.ts:L12-L14); la demo imposta trust proxy = 'loopback' (projects/storefrontapp/src/server.ts:L39-L39). Senza SSR_ALLOWED_ORIGINS il middleware di validazione è un no-op (core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L9-L30).
7. **Utenti loggati non ricevono lo state trasferito.** Il meta-reducer browser salta il merge se in storage c'è un access_token (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L81-L81, core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L119-L121): i dati vengono ricaricati dal client.
8. **Solo** INIT**.** Il merge avviene solo sull'azione INIT (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L76-L76); una feature registrata dopo riceve lo slice trasferito come stato precedente all'UPDATE solo se nessun reducer registrato ha cambiato riferimento nel frattempo; altrimenti combineReducers ha già scartato quella chiave e la feature parte dal proprio initialState (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L329-L341). Se lo slice arriva, il suo initialState non viene fuso con esso, salvo le sotto-chiavi assenti di una feature basata su mappa di reducer, che ricevono il default del rispettivo sotto-reducer (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L329-L341). DA VERIFICARE a runtime l'ordine effettivo di registrazione e le azioni che precedono l'UPDATE.
9. **Costo per azione sul server.** Il meta-reducer server ricalcola e riscrive cx-state dopo ogni azione (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L51-L61): vince l'ultimo state prima di BEFORE_APP_SERIALIZED.
10. **Solo JSON.** Istanze di classe, Date, Map non sopravvivono a TransferState (angular:packages/core/src/transfer_state.ts:L60-L62): lo state NgRx dichiarato in ssrTransfer deve essere serializzabile.
11. **Un solo script per tutti.** Hydration (__nghData__), Spartacus (cx-state) ed eventuale HTTP cache condividono <script id="ng-state"> (angular:packages/core/src/hydration/utils.ts:L46-L69, core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16): con hydration lo script è sempre presente, perché __nghData__ è scritto anche vuoto (angular:packages/core/src/hydration/annotate.ts:L283-L287).
12. **Richieste doppie.** Con withNoHttpTransferCache() le richieste OCC fatte in SSR vengono rifatte nel browser, salvo quanto già coperto dallo state trasferito (angular:packages/platform-browser/src/hydration.ts:L64-L76). Combinare withNoHttpTransferCache() e withHttpTransferCacheOptions() lancia un errore in dev mode (angular:packages/platform-browser/src/hydration.ts:L243-L253).
13. **Hydration +** enabledBlocking**.** Spartacus usa initialNavigation: 'enabledBlocking' (core-libs/storefront/router/app-routing.module.ts:L17-L17), che fornisce IS_ENABLED_BLOCKING_INITIAL_NAVIGATION (angular:packages/router/src/router_module.ts:L237-L244, angular:packages/router/src/provide_router.ts:L441-L444); in dev mode provideClientHydration emette il warning "found both hydration and enabledBlocking initial navigation ... contradiction" (angular:packages/platform-browser/src/hydration.ts:L152-L174). DA VERIFICARE a runtime: non è stato trovato codice che lo sopprima.
14. WindowRef.location **parziale.** In SSR ha solo href e origin e lancia se i token mancano (core-libs/core/src/window/window-ref.ts:L74-L83): pathname, search ecc. sono undefined sul server.
15. **Attributo di direttiva "congelato".** DirectiveStateTransferService.set/clear sono no-op nel browser (core-libs/storefront/utils/directive-state-transfer.service.ts:L23-L37): dopo navigazioni client

l'attributo `data-cx-state-transfer-*` resta quello SSR; supporta solo stringhe.

16. **Field initializer che usa `host`.** `PageTemplateDirective.currentTemplate` è inizializzato leggendo `this.host` (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L84-L87); funziona perché con `useDefineForClassFields: false` (tsconfig.json:L799-L799) le parameter properties sono assegnate prima degli initializer. DA VERIFICARE: comportamento di emissione TypeScript, non verificato con una build.
17. **`import()` e timer non tracciati.** Solo l'`import()` di `LazyComponentHandler` è protetto da `PendingTasks` (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71): un lavoro asincrono custom che non passa da Zone/HttpClient/`PendingTasks` può essere troncato dalla serializzazione o eseguito su un injector distrutto, dato che la platform viene distrutta subito dopo (angular:packages/platform-server/src/utils.ts:L236-L247).
18. **CDN che rimuovono i commenti** rompono l'hydration (`MISSING_SSR_CONTENT_INTEGRITY_MARKER`, angular:packages/core/src/hydration/utils.ts:L766-L810).
19. **Primo errore vince, HTML scartato.** In `CxCommonEngine` solo il primo errore propagato rigetta la promise e l'HTML generato viene scartato (core-libs/setup/ssr/engine/cx-common-engine.ts:L52-L64).
20. **Event replay dipende dalla build.** Si attiva nel browser solo se lo script inline della build ha creato `window._ejsas[appId]` (angular:packages/core/src/hydration/event_replay.ts:L76-L95, angular:packages/platform-server/src/utils.ts:L157-L186). DA VERIFICARE se `@angular-builders/custom-esbuild:application` inlinea `ng-event-dispatch-contract` come il builder standard (pacchetto non disponibile).
21. **Fallback CSR e hydration.** Un HTML restituito come template (trappola 1, o fallback di `OptimizedSsrEngine`, cap. 39) non contiene `__nghData__`: nel browser l'hydration resta disattivata (angular:packages/core/src/hydration/api.ts:L211-L228) e in dev mode compare il warning sulle annotazioni mancanti (angular:packages/core/src/hydration/api.ts:L229-L266). DA VERIFICARE a runtime (dipende anche da `isClientRenderModeEnabled`).
22. **SSG implicito.** `ngExpressEngine` imposta `publicPath` da `settings.views` (core-libs/setup/ssr/engine/ng-express-engine.ts:L113-L116), che la demo punta alla cartella browser (projects/storefrontapp/src/server.ts:L59-L59; DA VERIFICARE: che Express esponga le impostazioni dell'app in `options.settings` è comportamento di Express, non verificato sul sorgente, `express` non disponibile): `CommonEngine` serve quindi una pagina prerenderizzata se esiste `index.html` con marker SSG sotto quel path (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L155-L184), saltando il render (e `CxCommonEngine` non vede errori).

---

10. Come fare debug

- **È stato davvero fatto SSR?** Guardare il sorgente HTML (non il DOM ispezionato): con SSR riuscito l'host root ha `ng-server-context="ssr"` (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L190-L193, angular:packages/platform-server/src/utils.ts:L145-L155), gli host annotati hanno `ngh` e il body inizia con il commento `nghm` (angular:packages/platform-server/src/utils.ts:L127-L139). Se l'HTML è il template vuoto, cercare nel log del server la riga "is not allowed … Falling back to client side rendering" (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L123-L123) e impostare `NG_ALLOWED_HOSTS=…` o `allowedHosts`.
- **Cosa è stato trasferito?** Nel sorgente cercare `<script id="ng-state" type="application/json">` (angular:packages/platform-server/src/transfer_state.ts:L102-L109); le chiavi `cx-state` (state NgRx) e `__nghData__` (hydration) sono al primo livello del JSON. In console: `JSON.parse(document.getElementById('ng-state').textContent)['cx-state']`.

- **Lo state trasferito non viene applicato?** Verificare `access_token` nello storage (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L119-L121) e che la feature abbia la chiave in `state.ssrTransfer.keys` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L24-L35); un breakpoint nel meta-reducer browser su `action.type === INIT` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L76-L90) mostra lo slice mergeato.
- **Render che non termina / timeout.** Significa `whenStable()` mai risolto: cercare `setInterval`, polling, observable HTTP mai completati o `PendingTasks` non rimossi (angular:packages/core/src/application/application_ref.ts:L326-L350, angular:packages/core/src/change_detection/scheduling/ng_zone_scheduling.ts:L231-L281). Nel browser, in dev mode, il warning dopo 10 s di `whenStableWithTimeout` è un buon indizio dello stesso problema (angular:packages/core/src/hydration/api.ts:L164-L182).
- **Tempi delle fasi.** `ngExpressEngine({ bootstrap, enablePerformanceProfiler: true })` stampa la durata di "Retrieve SSG Page", "Render Page", "Inline Critical CSS" (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L136-L149).
- **Errori SSR.** Gli errori arrivati a `ErrorHandler` sono propagati a Express tramite `PROPAGATE_ERROR_TO_SERVER` (core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts:L26-L38) e gestiti da `defaultExpressErrorHandlers` (projects/storefrontapp/src/server.ts:L77-L77, cap. 39). In prerendering il token non c'è e l'handler è no-op (core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L9-L22).
- **Mismatch di hydration (NG05xx).** Isolare il componente con `ngSkipHydration` come fa Spartacus per carousel e form (core-libs/storefront/shared/components/carousel/carousel.component.ts:L57-L65); verificare che il componente non legga `window`/`document` direttamente ma tramite `WindowRef` (core-libs/core/src/window/window-ref.ts:L37-L44).
- **URL/origin sbagliati in SSR.** Loggare `inject(SERVER_REQUEST_URL)` / `inject(SERVER_REQUEST_ORIGIN)`; controllare `SERVER_REQUEST_ORIGIN` in env (trappola 5) e `trust proxy` (trappola 6).
- **Test e2e con config dinamica.** `TestConfigServerModule` legge la config da un cookie della request (core-libs/setup/ssr/testing/test-config-server.module.ts:L26-L58): utile solo in test, mai in produzione.

---

11. Domande di autoverifica

1. **In quale injector finiscono i `providers` passati a `res.render(..., { providers })` in SSR?** Nel platform injector: `CommonEngine` li passa come `platformProviders` a `renderApplication`, che li registra insieme a `INITIAL_CONFIG` in `platformServer` (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L190-L205, angular:packages/platform-server/src/utils.ts:L55-L74).
2. **Perché `serverRequestOriginFactory` usa `skipSelf: true`?** Perché il provider app-level di `provideServer` fornisce lo stesso token di quello platform di `getServerRequestProviders()`; con `skipSelf` legge il valore Express del parent invece di sé stesso (core-libs/setup/ssr/providers/server-request-origin.ts:L16-L29, core-libs/setup/ssr/providers/ssr-providers.ts:L53-L66).
3. **Cosa succede se avvio il server della demo senza `NG_ALLOWED_HOSTS=…` e senza `allowedHosts`?** `CommonEngine` non riesce a validare l'hostname dell'URL assoluto costruito da `ngExpressEngine`, logga un errore e restituisce il template non renderizzato (fallback CSR) (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L136, core-libs/setup/ssr/engine/ng-express-engine.ts:L104-L106).
4. **Sotto quale chiave e in quale elemento HTML viaggia lo state NgRx di Spartacus?** Chiave `cx-state` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16) dentro `<script id="ng-state" type="application/json">` (con `APP_ID` di default `'ng'`) (angular:packages/platform-server/src/

transfer_state.ts:L102-L109, angular:packages/core/src/application/application_tokens.ts:L44-L52).

5. **Quali slice di state vengono trasferiti OOTB e come se ne aggiunge uno?** site-context, site-theme, cms, product (core-libs/core/src/cms/store/cms-store.module.ts:L21-L31 e moduli analoghi); se ne aggiunge uno con `state.ssrTransfer.keys: { chiave: StateTransferType.TRANSFER_STATE }` (core-libs/core/src/state/config/state-config.ts:L20-L39).
6. **Perché un utente loggato non beneficia dello state trasferito?** Il meta-reducer browser esegue il `deepMerge` solo se `isLoggedIn` è falso, valutato come presenza di `token.access_token` nello storage (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L81-L90, core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L119-L121).
7. **Perché Spartacus usa `withNoHttpTransferCache()` e che effetto ha?** Perché ha un proprio meccanismo di state transfer (docs/migration/221121_7/migration.md:L207-L209), dichiarato temporaneo (core-libs/schematics/src/add-ssr/update-app-config-in-ssr.ts:L47-L50); l'effetto è che le richieste HTTP fatte in SSR vengono ripetute nel browser (angular:packages/platform-browser/src/hydration.ts:L64-L76).
8. **Come impedisce Spartacus che SSR serializzi la pagina mentre un componente CMS lazy si sta ancora caricando?** Registrando l'`import()` come `PendingTasks.add()` e rimuovendolo con `finalize`, così `whenStable()` non risolve prima (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71, angular:packages/core/src/application/application_ref.ts:L326-L350).
9. **Nel browser, da cosa dipende l'attivazione dell'hydration?** Dalla presenza di `__nghData__` in `TransferState` (angular:packages/core/src/hydration/api.ts:L211-L228), che il server scrive sempre durante `annotateForHydration` (angular:packages/core/src/hydration/annotate.ts:L283-L301).
10. **Cosa fa `CxCommonEngine` che `CommonEngine` non fa?** Fornisce `PROPAGATE_ERROR_TO_SERVER`, memorizza il primo errore propagato dall'app e, a render completato, rigetta la promise con quell'errore invece di restituire l'HTML (core-libs/setup/ssr/engine/cx-common-engine.ts:L41-L66).

TAVOLA 07.1 **4.1 Richiesta SSR end-to-end (demo)**

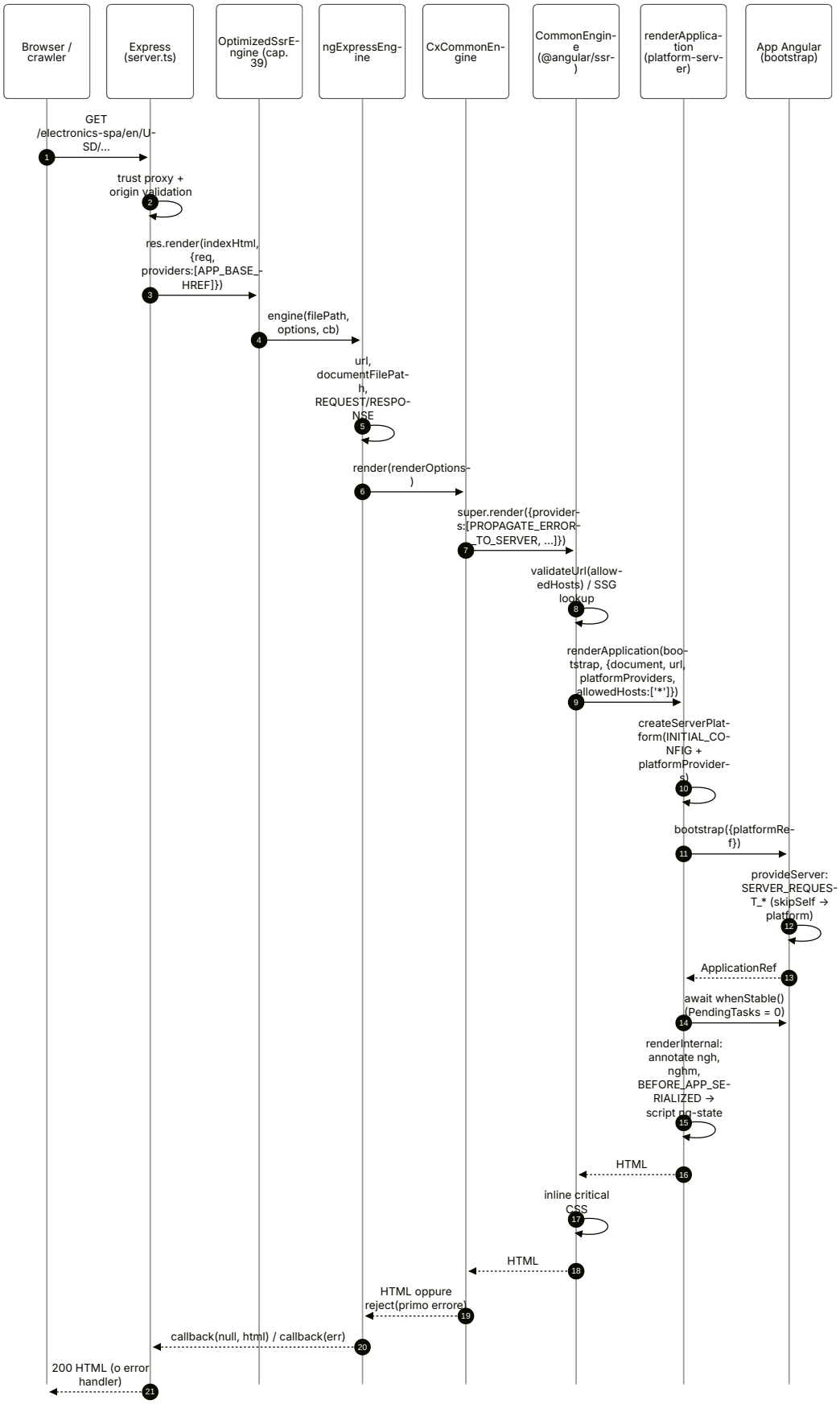


TAVOLA 07.2 **4.2 Decisioni di CommonEngine.render**

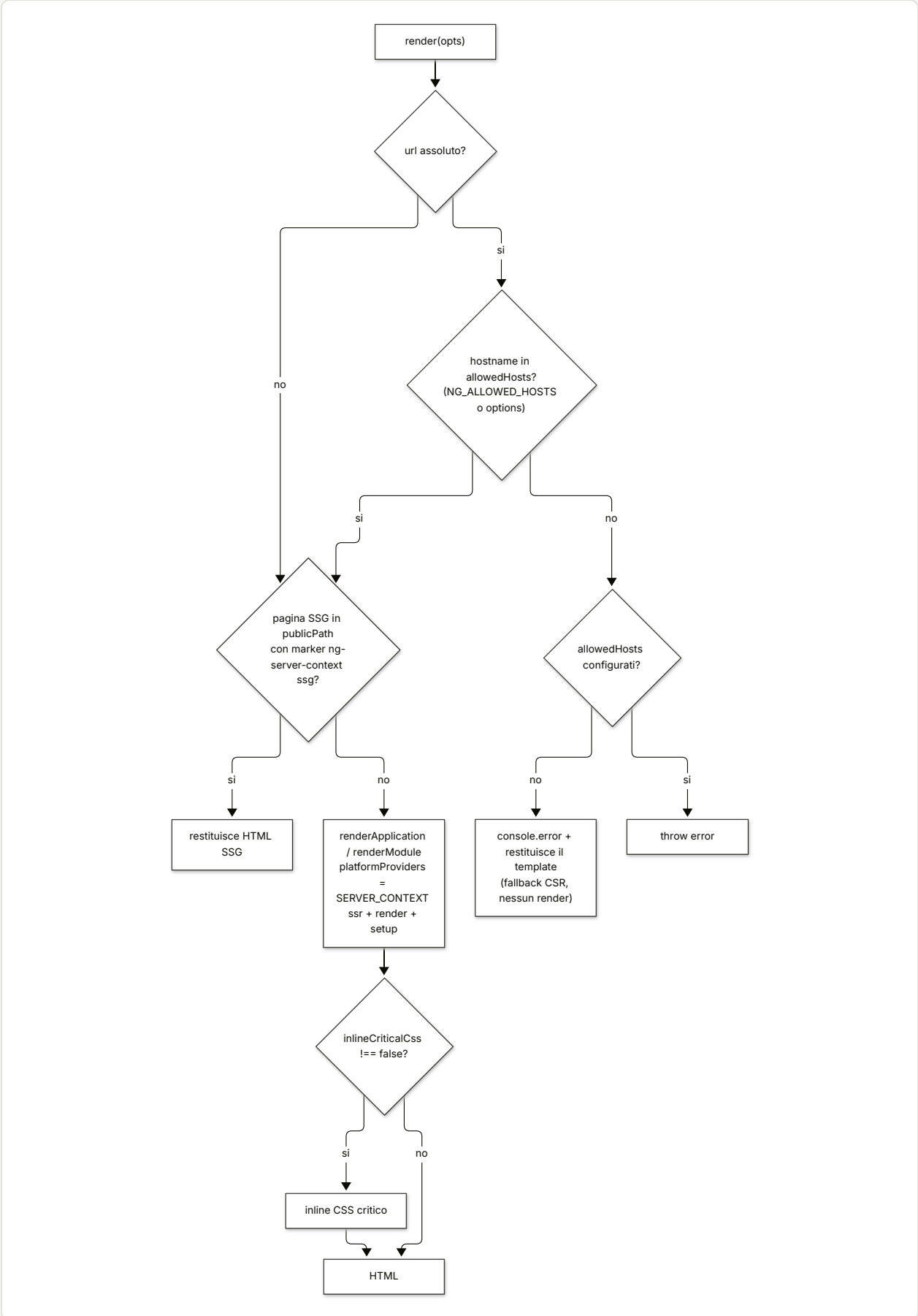
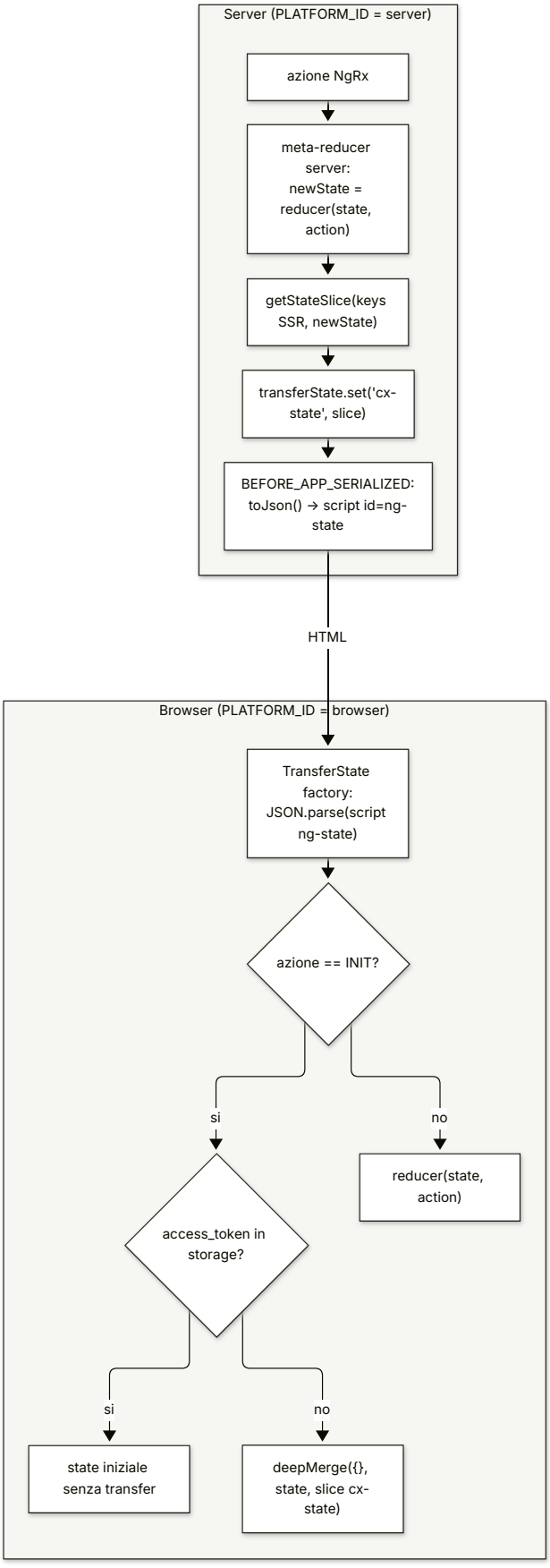


TAVOLA 07.3 **4.3 Transfer dello state NgRx**



08

Schematics e builder

Trasformazioni su `Tree` per `ng add`/`ng update` e builder Architect eseguiti da Nx.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Angular @ `v21.2.23` (prefisso `angular:`). Il devkit (`@angular-devkit/schematics` 21.2.23, `@angular-devkit/architect` 0.2102.23, `@schematics/angular` 21.2.23, `@angular/cli` 21.2.23) **non** sta in `/tmp/angular`: proviene dal repo `angular/angular-cli` ed è letto dai pacchetti npm pubblicati (codice JS compilato e `.d.ts`), con prefisso `deps:devkit/...` (path relativi a `/tmp/deps`).

---

1. A cosa serve e perché esiste

Un framework come Spartacus non vive solo di codice runtime: deve anche **installarsi** in un'applicazione Angular esistente, **aggiornarla** tra una versione e l'altra e farsi **compilare** in pacchetti npm. Angular offre per questo due meccanismi distinti, entrambi nati nel repo `angular/angular-cli`:

1. **Schematics** — trasformazioni di codice dichiarative su un file system *virtuale* (`Tree`). Sono il motore dietro `ng add`, `ng generate` e `ng update`. Una schematic è una funzione `RuleFactory<T> = (options: T) => Rule`, e una `Rule` è `(tree, context) => Tree | Observable<Tree> | Rule | Promise<...> | void` (`deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/interface.d.ts:L137-L137`, `deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/interface.d.ts:L153-L154`). Le modifiche restano in memoria fino alla fine e vengono scritte su disco solo se tutto è andato a buon fine e non si è in dry-run (`deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L77-L100`).
2. **Builder (Architect)** — l'astrazione con cui un *target* di un progetto (`build`, `test`, `serve`, …) viene eseguito: una funzione `(options, context) => BuilderOutput | Promise | Observable | AsyncIterable` trasformata in handler da `createBuilder` (`deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js:L18-L21`, `deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js:L137-L149`).

Dove sta il codice. In `/tmp/angular` (repo `angular/angular`) ci sono solo gli *schematics dei pacchetti framework*: `packages/core/schematics` (migrazioni `ng update` e generatori `ng generate @angular/core:*`) e `packages/localize/schematics` (`ng add @angular/localize`) (`angular:packages/core/schematics/collection.json:L1-L85`, `angular:packages/localize/schematics/collection.json:L1-L10`). Il **motore** degli schematics, il runner dei builder, la collection del CLI (`component`, `module`, …) e la logica dei comandi `ng add`/`ng update` stanno invece nel repo `angular/angular-cli`, pubblicati come `@angular-devkit/schematics`, `@angular-devkit/architect`, `@schematics/angular`, `@angular/cli`. Gli schematics di core lo dichiarano esplicitamente lasciando il devkit *esterno* al bundle: lo usano dal progetto ospite (`angular:packages/core/schematics/rollup.config.js:L56-L58`).

Spartacus usa entrambi i meccanismi:

- pubblica il pacchetto `@spartacus/schematics`, che si aggancia a `ng add` (installazione), `ng generate` (generatori) e `ng update` (migrazioni) tramite i campi `schematics`, `ng-add` e `ng-update` del proprio `package.json` (`core-libs/schematics/package.json:L43-L90`); ogni feature/integration lib ha in più una propria piccola collection (`feature-libs/cart/schematics/collection.json:L3-L17`);
- nel monorepo, i target delle librerie e dell'app demo sono eseguiti da Nx tramite `project.json` con chiave `executor`, che punta a builder Architect ufficiali, di terze parti o **locali** (`tools/build-lib`) (`nx.json:L6-L23`, `core-libs/core/project.json:L8-L19`, `tools/build-lib/builders.json:L1-L14`).

Questo capitolo spiega il **meccanismo** (devkit, CLI, Architect) e mostra *a grandi linee* come Spartacus lo sfrutta. I dettagli di ogni schematic Spartacus (`add-spartacus`, `add-ssr`, `wrapper-module`, dati di deprecazione, `modernize-app-*`, `ai-context`) sono in 43 — Schematics e migrazioni di Spartacus; i test e il tooling Nx in 44 — Strategia di test e tooling del monorepo; l'SSR aggiunto da `add-ssr` in 07 — SSR, hydration e TransferState.

2. Mappa dei file

PATH	RUOLO
angular:packages/core/package.json	campi schematics, ng-update.migrations, exports ./schematics/* di @angular/core
angular:packages/core/schematics/collection.json	14 schematics ng generate @angular/core:*
angular:packages/core/schematics/migrations.json	6 migrazioni ng update versione 21.0.0
angular:packages/core/schematics/BUILD.bazel, angular:packages/core/schematics/rollup.config.js	bundling dei sorgenti in bundles/*.cjs (devkit esterno)
angular:packages/core/schematics/migrations/application-config-core/index.ts	esempio di migrazione: migrate(): Rule che delega a Tsurge
angular:packages/core/schematics/utils/tsurge/helpers/angular_devkit/run_in_devkit.ts	runMigrationInDevkit: analyze → combine → migrate → tree.beginUpdate
angular:packages/core/schematics/utils/project_tsconfig_paths.ts	tsconfig dei target build/test letti dal workspace
angular:packages/core/schematics/utils/typescript/compiler_host.ts	ts.Program costruito sul Tree, canMigrateFile
angular:packages/core/schematics/ng-generate/inject-migration/index.ts, schema.json	esempio di generatore con x-prompt
angular:packages/localize/schematics/ng-add/index.ts	ng add @angular/localize con le utility di @schematics/angular
deps:devkit/angular-devkit-schematics-21.2.23/package/src/tree/interface.d.ts	Tree, UpdateRecorder, MergeStrategy
deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/interface.d.ts	tipi Rule, Source, RuleFactory, SchematicContext
deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js, call.js, schematic.js	chain, noop, mergeWith, branchAndMerge, normalizzazione dei risultati, schematic/externalSchematic
deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js, deps:devkit/angular-devkit-schematics-21.2.23/package/src/sink/host.js	esecuzione, sink (dry-run, errori, disco), post-task
deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js	createSchematic (privati), scheduler delle task
deps:devkit/angular-devkit-schematics-21.2.23/package/tools/node-module-engine-host.js, export-ref.js, file-system-engine-host-base.js	risoluzione collection da node_modules, factory path#export, alias, privati/hidden
deps:devkit/angular-devkit-schematics-21.2.23/package/tasks/package-manager/install-task.js, deps:devkit/angular-devkit-schematics-21.2.23/package/tasks/run-schematic/task.js	NodePackageInstallTask, RunSchematicTask
deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js	logica ng add
deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js, deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js, deps:devkit/angular-cli-21.2.23/package/src/commands/update/schematic/index.js	logica ng update: package group, selezione e ordinamento migrazioni

PATH	RUOLO
deps:devkit/schematics-angular-21.2.23/package/collection.json, deps:devkit/schematics-angular-21.2.23/package/utility/workspace-models.d.ts	collection del CLI, enum dei builder ufficiali
deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js, deps:devkit/angular-devkit-architect-0.2102.23/package/builders/concat.js	createBuilder, builder concat
package.json (root)	versioni devkit/CLI/Nx, script build:*, build:schematics, generate:deps, test:schematics
nx.json	targetDefaults Nx
core-libs/schematics/package.json	metadati schematics / ng-add / ng-update, peer deps devkit
core-libs/schematics/src/collection.json	registro schematics Spartacus (ng add, ng generate)
core-libs/schematics/src/migrations/migrations.json	registro migrazioni ng update Spartacus
core-libs/schematics/src/ng-add/index.ts	entry di ng add @spartacus/schematics
core-libs/schematics/src/shared/utils/lib-utils.ts, feature-utils.ts, new-module-utils.ts	SchematicConfig, task di install, addFeatures, ensureModuleExists
core-libs/schematics/src/shared/utils/program.ts, tree-file-system.ts	ts-morph sopra il Tree
core-libs/schematics/src/shared/utils/workspace-utils.ts	lettura di angular.json, progetto di default
core-libs/schematics/index.ts	utility esportate agli schematics delle altre librerie (non stabile)
feature-libs/cart/schematics/collection.json, feature-libs/cart/ng-package.json	esempio di collection di feature lib e suo impacchettamento
core-libs/*/project.json, projects/storefrontapp/project.json	target Nx ed executor
tools/build-lib/package.json, tools/build-lib/builders.json, tools/build-lib/declaration-merging/index.ts, tools/build-lib/augmented-types/index.ts	builder Architect locali
projects/storefrontapp/esbuild/plugins.ts	plugin esbuild per @angular-builders/custom-esbuild

3. Come funziona passo per passo

3.1 Come un pacchetto npm "si dichiara" al CLI

Il CLI non conosce a priori gli schematics di un pacchetto: li scopre leggendo il suo package.json.

- **Collection.** Il campo schematics indica il file collection.json. L'engine host risolve require.resolve('<nome>/package.json'), legge schematics e, se è un path relativo, lo risolve rispetto alla cartella del pacchetto; se il campo manca lancia NodePackageDoesNotSupportSchematics (deps:devkit/angular-devkit-schematics-21.2.23/package/tools/node-module-engine-host.js:L46-L62). @angular/core dichiara "schematics": "./schematics/collection.json" (angular:packages/core/package.json:L39-L49); Spartacus "schematics": "./src/collection.json" (core-libs/schematics/package.json:L90-L90).

- **Migrazioni.** Il campo `ng-update.migrations` punta a un secondo file di collection, usato solo da `ng update`. Con `--migrate-only` (metodo `migrateOnly`) il CLI rifiuta path assoluti o che escono dalla root del pacchetto (deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L224-L224, deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L262-L270); nel flusso normale il path viene risolto rispetto alla cartella del pacchetto (o con `require.resolve`) senza questi controlli, e lo schematic interno verifica solo che sia una stringa (deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L479-L488, deps:devkit/angular-cli-21.2.23/package/src/commands/update/schematic/index.js:L308-L315). Core: `./schematics/migrations.json` (angular:packages/core/package.json:L39-L42); Spartacus: `./src/migrations/migrations.json` (core-libs/schematics/package.json:L46-L47).
- **Package group.** `ng-update.packageGroup` (array o mappa) elenca i pacchetti che vanno aggiornati insieme al "capogruppo" (deps:devkit/angular-cli-21.2.23/package/src/commands/update/schematic/index.js:L284-L306, deps:devkit/angular-cli-21.2.23/package/src/commands/update/schematic/index.js:L358-L375). `@spartacus/schematics` elenca 39 pacchetti `@spartacus/*` (core-libs/schematics/package.json:L48-L88).
- **Salvataggio di `ng add`.** `ng-add.save` decide dove finisce la dipendenza: il CLI legge `manifest['ng-add']?.save` e installa come devDependency quando vale `'devDependencies'` (deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L425-L426, deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L473-L473). Sia `@angular/localize` sia `@spartacus/schematics` usano `devDependencies` (angular:packages/localize/package.json:L25-L31, core-libs/schematics/package.json:L43-L45).

3.2 La collection: factory, schema, alias, privati

Ogni voce di `collection.json` ha almeno `factory` e (opzionalmente) `schema`, `aliases`, `private`, `hidden`:

- `factory` è nella forma `"<percorso>#<export>"`: `ExportStringRef` divide sulla `#`, risolve il modulo con `require.resolve` e prende l'export nominato o `default` se manca (deps:devkit/angular-devkit-schematics-21.2.23/package/tools/export-ref.js:L16-L27). Gli schematics di core puntano a bundle CommonJS (`./bundles/inject-migration.cjs#migrate`) generati da Rollup in Bazel (angular:packages/core/schematics/collection.json:L2-L84, angular:packages/core/schematics/BUILD.bazel:L63-L67, angular:packages/core/schematics/BUILD.bazel:L142-L187); lo `ng-add` di Spartacus usa `"./ng-add/index"` senza `#`, quindi l'export `default` (core-libs/schematics/src/collection.json:L4-L11).
- `schema` è un JSON Schema. Con il CLI, `NodeWorkflow` è creato con un `CoreSchemaRegistry` proprio e `schemaValidation: true`, quindi registra una trasformazione che valida le opzioni contro lo schema; il CLI aggiunge poi `addUndefinedDefaults` al registry (il ramo di base.js:L48 vale solo se il chiamante non fornisce un registry) (deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L160-L164, deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L177-L177, deps:devkit/angular-devkit-schematics-21.2.23/package/tools/workflow/node-workflow.js:L56-L58, deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L43-L49). Gli schema possono contenere `x-prompt` per chiedere interattivamente le opzioni (angular:packages/core/schematics/ng-generate/inject-migration/schema.json:L1-L32, core-libs/schematics/src/add-spartacus/schema.json:L94-L96). Il CLI registra lo smart default `projectName` (usato da `$default.$source: projectName`) e installa il *prompt provider* per `x-prompt` solo se la sessione è interattiva e c'è un TTY (deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L179-L179, deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L194-L195).
- `aliases` vengono risolti prima del nome (deps:devkit/angular-devkit-schematics-21.2.23/package/tools/file-system-engine-host-base.js:L146-L153): per questo `ng g @spartacus/schematics:spa` equivale ad `add-spartacus` (core-libs/schematics/src/collection.json:L12-L17).
- `private` e `hidden`: `listSchematicNames` esclude i privati e (salvo richiesta) gli hidden (deps:devkit/angular-devkit-schematics-21.2.23/package/tools/file-system-engine-host-base.js:L87-L102);

`createSchematic` lancia `PrivateSchematicException` se lo schematic è privato e non è stato chiesto `allowPrivate` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L226-L256). Il workflow consente i privati se richiesto esplicitamente o se il chiamante appartiene alla stessa collection (deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L108-L110).

3.3 Il modello di esecuzione: Tree, Rule, chain, merge

1. **Tree virtuale.** Il workflow crea un `new HostTree(this._host)` sul workspace (deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L121-L121). `HostTree` registra tutte le modifiche in un `CordHost` sopra un host di sola lettura: nessuna regola tocca il disco (deps:devkit/angular-devkit-schematics-21.2.23/package/src/tree/host-tree.js:L85-L86). L'interfaccia `Tree` offre lettura (`read`, `readText`, `readJson`, `exists`, `get`, `getDir`, `visit`), scrittura (`create`, `overwrite`, `delete`, `rename`), aggiornamenti per offset (`beginUpdate`/`commitUpdate` con `UpdateRecorder`), `branch`/`merge` e l'elenco `actions` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/tree/interface.d.ts:L38-L77, deps:devkit/angular-devkit-schematics-21.2.23/package/src/tree/interface.d.ts:L82-L86).
2. **Normalizzazione del risultato di una Rule.** `callRuleAsync` esegue la regola; finché il risultato è una funzione la richiama (una `Rule` può restituire un'altra `Rule`), `undefined` significa "tree invariato", un Observable viene ridotto all'ultimo valore, altrimenti serve un `Tree` o parte `InvalidRuleResultException` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/call.js:L76-L92).
3. **Composizione.** `chain(rules)` esegue le regole in sequenza passando a ciascuna il tree prodotto dalla precedente, e restituisce a sua volta una funzione `() => intermediateTree` che il ciclo del punto 2 risolve (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js:L46-L61). `noop()` non restituisce nulla, quindi lascia il tree invariato (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js:L76-L78). `mergeWith(source, strategy)` fonde un tree generato (es. da template); `branchAndMerge(rule, strategy)` esegue la regola su `tree.branch()` e poi fa `merge` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js:L71-L75, deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js:L92-L96).
4. **Conflitti.** `MergeStrategy` stabilisce come risolverli: `Default=0`, `Error=1`, `AllowOverwriteConflict=2` (= `ContentOnly`), `AllowCreationConflict=4`, `AllowDeleteConflict=8`, `Overwrite=14` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/tree/interface.d.ts:L10-L18).
5. **Schematics che chiamano schematics.** `schematic(name, options)` invoca uno schematic della *stessa* collection (privati inclusi) su un ramo del tree e fa merge con `AllowOverwriteConflict` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/schematic.js:L38-L49). `externalSchematic(collection, name, options)` crea la collection per nome (risolta come pacchetto npm), non passa `allowPrivate` e fa lo stesso merge (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/schematic.js:L22-L31).
6. **Context.** `SchematicContext` espone `debug`, `engine`, `logger`, `schematic`, `strategy`, `interactive` e `addTask(task, dependencies?)` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/interface.d.ts:L119-L132).

3.4 Fine della schematic: sink e post-task

1. Il tree finale passa per tre "sink" in sequenza: un `DryRunSink` che emette il report degli eventi, un sink che fallisce con `UnsuccessfulWorkflowExecution` se è stato segnalato un errore, e infine un `HostSink` che scrive su disco **solo se non è dry-run** (deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L77-L100, deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L121-L124).
2. Il `HostSink` applica nell'ordine cancellazioni, rinomine, creazioni, aggiornamenti (deps:devkit/angular-devkit-schematics-21.2.23/package/src/sink/host.js:L71-L71).

3. Solo dopo il commit, e mai in dry-run, il workflow esegue `engine.executePostTasks()` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L125-L133). Le task sono schedulate da `TaskScheduler.schedule` con id progressivo e priorità calcolata dalle dipendenze, **senza deduplicazione** (deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L89-L137), ed eseguite in sequenza con `concatMap`, riusando un executor per nome (deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L283-L297).
4. Due task standard: `NodePackageInstallTask` (`{name: NodePackageName, options: {command: 'install', ...}}`, default `quiet`/`hideOutput` `true`, `allowScripts` `false`) (deps:devkit/angular-devkit-schematics-21.2.23/package/tasks/package-manager/install-task.js:L12-L58) e `RunSchematicTask(collection?, schematic, options)`, che pianifica l'esecuzione di uno schematic *dopo* il commit; con due argomenti la collection è `null`, cioè la stessa (deps:devkit/angular-devkit-schematics-21.2.23/package/tasks/run-schematic/task.js:L12-L33).

3.5 I comandi del CLI

- `ng add <pkg>`. Usa lo schematic `ng-add` con `allowPrivateSchematics = true`, mentre gli altri comandi schematic hanno `false` (deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L98-L99, deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L112-L112, deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L341-L341). Dopo l'installazione verifica l'esistenza di `ng-add` con `collection.createSchematic('ng-add', true)` perché "Most `ng-add` schematics are marked as private" (deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L216-L225).
- `ng generate <collection>:<name>`. Passa per `SchematicsCommandModule` con `allowPrivate: this.allowPrivateSchematics = false` (deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L339-L343).
- `ng update <pkg>`. Lo schematic interno di update aggiorna `package.json` (inclusi i pacchetti del group) e passa al CLI, tramite `global.externalMigrations`, l'elenco `{package, collection, from, to}`; il CLI esegue le migrazioni dopo l'installazione (deps:devkit/angular-cli-21.2.23/package/src/commands/update/schematic/index.js:L252-L272, deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L440-L501). `executeMigrations` seleziona le voci con `version` nell'intervallo semver `(from, to]`, le divide in obbligatorie e `optional`, le ordina per versione e poi per **nome** (`localeCompare`) (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L105-L142). Le opzionali sono proposte con un prompt a scelta multipla se c'è un TTY; altrimenti vengono solo elencate (con il comando `ng update <pkg> --name <migrazione>`) e non eseguite (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L232-L255, deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L235-L243, deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L137-L140). Dopo ogni migrazione i file modificati sono formattati con Prettier e, con `--create-commits`, si crea un commit per migrazione (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L166-L190). Il blocco "un major alla volta" vale solo per `@angular/*` e `@nguniversal/*` (deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L353-L373, deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/constants.js:L15-L15).
- Le migrazioni del CLI stesso stanno in `@schematics/angular/migrations/migration-collection.json`, collegate da `@angular/cli` via `ng-update.migrations` (deps:devkit/angular-cli-21.2.23/package/package.json:L47-L51, deps:devkit/schematics-angular-21.2.23/package/migrations/migration-collection.json:L1-L28).

3.6 Cosa c'è in `/tmp/angular`: gli schematics di `@angular/core` e `@angular/localize`

- **Collection e migrazioni.** `@angular/core` registra 14 schematics "generate" (es. `standalone-migration`, `inject-migration`, `signal-input-migration`, `control-flow-migration`, `cleanup-unused-imports`) (angular:packages/core/schematics/collection.json:L2-L84) e 6 migrazioni `21.0.0` (`control-flow-migration`, `router-current-navigation` opzionale, `router-last-successful-navigation`, `application-config-core`, `add-bootstrap-context-to-server-main`, `bootstrap-options-migration`) (angular:packages/core/schematics/migrations.json:L2-L34). `control-flow-migration` è sia generatore sia migrazione (angular:packages/core/schematics/collection.json:L61-L66, angular:packages/core/schematics/migrations.json:L3-L7).
- **Struttura.** Sorgenti TS in `migrations/<nome>/index.ts` e `ng-generate/<nome>/index.ts`; Bazel+Rollup li impacchettano in `bundles/<nome>.cjs` (CommonJS, senza sourcemap) (angular:packages/core/schematics/BUILD.bazel:L142-L187), lasciando esterni `typescript`, `tslib`, `@angular-devkit/*` e `@angular/*` (angular:packages/core/schematics/rollup.config.js:L56-L58).
- **Tsurge.** Una migrazione è `migrate(): Rule` che delega a `runMigrationInDevkit` (angular:packages/core/schematics/migrations/application-config-core/index.ts:L9-L20). Tsurge divide le migrazioni in analyze, merge e migrate, pensate anche per worker separati, con le varianti `TsurgeFunnelMigration` e `TsurgeComplexMigration` (angular:packages/core/schematics/utils/tsurge/migration.ts:L18-L44, angular:packages/core/schematics/utils/tsurge/migration.ts:L58-L100). `runMigrationInDevkit` ricava i tsconfig dal workspace (lanciando `SchematicsException` se non ne trova), crea un programma per ciascuno, assegna ogni file sorgente a una sola unità di compilazione, esegue `analyze`, combina, calcola `globalMeta`, ottiene le `replacements` e le applica al `Tree` (angular:packages/core/schematics/utils/tsurge/helpers/angular_devkit/run_in_devkit.ts:L60-L139, angular:packages/core/schematics/utils/tsurge/helpers/angular_devkit/run_in_devkit.ts:L63-L67, angular:packages/core/schematics/utils/tsurge/helpers/angular_devkit/run_in_devkit.ts:L141-L173).
- **TypeScript sul Tree.** `getProjectTsConfigPaths` legge il workspace dal `Tree` e raccoglie i `tsConfig` dei soli target `build` e `test` (angular:packages/core/schematics/utils/project_tsconfig_paths.ts:L16-L51); il `ts.Program` viene costruito sul contenuto del `Tree` sovrascrivendo `host.readFile` (angular:packages/core/schematics/utils/typescript/compiler_host.ts:L71-L103); `canMigrateFile` esclude `.d.ts`, `.ngtypecheck.ts`, librerie esterne e file fuori dalla root (angular:packages/core/schematics/utils/typescript/compiler_host.ts:L111-L131).
- **Localize.** `ng add @angular/localize` è una `chain` di tre regole (types nei tsconfig, polyfill `@angular/localize/init`, eventuale spostamento in `dependencies`) (angular:packages/localize/schematics/ng-add/index.ts:L168-L182) costruita con le utility pubbliche di `@schematics/angular/utility` e con l'enum `AngularBuilder` per riconoscere i builder dei target (angular:packages/localize/schematics/ng-add/index.ts:L11-L19, angular:packages/localize/schematics/ng-add/index.ts:L27-L63, deps:devkit/schematics-angular-21.2.23/package/utility/workspace-models.d.ts:L18-L38).
- **Test.** Le migrazioni di core si testano con `SchematicTestRunner` puntato su `migrations.json` (angular:packages/core/schematics/test/application_config_core_spec.ts:L30-L34).

3.7 Spartacus: il pacchetto `@spartacus/schematics`

1. **Dipendenze.** Il monorepo installa `@angular-devkit/schematics ~21.2.23`, `@angular-devkit/core ~21.2.23`, `@schematics/angular ~21.2.23`, `@angular/cli ^21.2.23`, `@angular-devkit/build-angular 21.2.23`, `@angular-builders/custom-esbuild ^21.0.3`, `ng-packagr ^21.2.3`, `nx ^22.7.6` (package.json:L201-L221, package.json:L271-L273). Il pacchetto `@spartacus/schematics` dichiara il devkit come `peerDependencies`: il motore è quello del progetto ospite (core-libs/schematics/package.json:L26-L34); a runtime porta `@angular/pwa`, `@angular/ssr`, `semver`, `ts-morph`, `tslib` (core-libs/schematics/package.json:L19-L25).

2. **Build e pubblicazione.** Non passa per ng-packagr: è compilato con `tsc -p ./tsconfig.schematics.json` dopo `generate:deps` (core-libs/schematics/package.json:L12-L14), in CommonJS, escludendo template, spec e dati di test (core-libs/schematics/tsconfig.schematics.json:L2-L29). `.npmignore` esclude i `.ts` non dichiarazione: si pubblica il JS accanto ai sorgenti (core-libs/schematics/.npmignore:L1-L14). `src/dependencies.json` (pacchetto → peerDependencies) è generato da `tools/schematics/dependency-collector.ts` (tools/schematics/dependency-collector.ts:L21-L21, package.json:L64-L64, core-libs/schematics/src/shared/utils/package-utils.ts:L21-L21).
3. **Collection.** 9 voci: `ng-add` (private, hidden, alias `install`), `add-spartacus` (`spa`), `add-ssr` (`ssr`), `add-cms-component` (`cms`), `wrapper-module` (`wrapper`), `ai-context` (`ai`) e tre `modernize-app-*` (core-libs/schematics/src/collection.json:L3-L53). `ng-add`, `add-spartacus` e `add-ssr` condividono `./add-spartacus/schema.json` (core-libs/schematics/src/collection.json:L4-L23); `wrapper-module` e `modernize-app-*` non hanno `schema` (core-libs/schematics/src/collection.json:L30-L52).
4. `ng-add`. È una `chain` di `schematic('add-spartacus')` e, se `options.ssr`, `schematic('add-ssr')` (core-libs/schematics/src/ng-add/index.ts:L17-L24). Lo schema prende `project` dal progetto corrente, ha `features` con `x-prompt`, `lazy` (default `true`), `ssr` (default `false`), `aiTools` (default `["agents"]`) (core-libs/schematics/src/add-spartacus/schema.json:L3-L13, core-libs/schematics/src/add-spartacus/schema.json:L338-L342, core-libs/schematics/src/add-spartacus/schema.json:L381-L385, core-libs/schematics/src/add-spartacus/schema.json:L390-L414); il tipo TS è scritto a mano (core-libs/schematics/src/add-spartacus/schema.ts:L7-L12).
5. `addSpartacus`. Una lunga `chain`: struttura app, app module/config, store, `SpartacusModule`, features module, configurazione, feature toggles, stili, budget, feature, AI context, dipendenze, install, `finalizeInstallation` (core-libs/schematics/src/add-spartacus/index.ts:L531-L592). L'install npm è una task: `installPackageJsonDependencies()` → `context.addTask(new NodePackageInstallTask())` (core-libs/schematics/src/shared/utils/lib-utils.ts:L730-L741).
6. **Composizione con altri schematics.** I moduli mancanti si creano con `externalSchematic('@schematics/angular', 'module', {…, flat: true, typeSeparator: '.'})`, oppure `noop()` se il file ha già un `@NgModule` (core-libs/schematics/src/shared/utils/new-module-utils.ts:L40-L77, core-libs/schematics/src/shared/constants.ts:L10-L10); `add-cms-component` usa `@schematics/angular:module|component` (core-libs/schematics/src/add-cms-component/index.ts:L412-L424); `addSSR` invoca `externalSchematic('@angular/ssr', 'ng-add', {project})` e poi fonde i template con `MergeStrategy.Overwrite` (core-libs/schematics/src/add-ssr/index.ts:L437-L465, core-libs/schematics/src/shared/constants.ts:L15-L15).
7. **Feature senza schematics di libreria.** `addFeatures` *non* chiama gli schematics delle feature lib: per ogni feature prende il `SchematicConfig` da `featureSchematicConfigMapping`, applica in-process `addLibraryFeature` e, per gli `importAfter`, `externalSchematic('@spartacus/schematics', 'wrapper-module', …)`; una feature senza configurazione provoca `SchematicsException` (core-libs/schematics/src/shared/utils/feature-utils.ts:L102-L153, core-libs/schematics/src/shared/utils/feature-utils.ts:L125-L132). `SchematicConfig` descrive una feature in modo dichiarativo (core-libs/schematics/src/shared/utils/lib-utils.ts:L115-L199); `LibraryOptions` estende `Partial<ExecutionOptions>` del devkit (core-libs/schematics/src/shared/utils/lib-utils.ts:L85-L110).
8. **Task concatenate.** `scheduleAiContext` pianifica un `NodePackageInstallTask` e poi `RunSchematicTask('ai-context', …)` *dipendente* dal suo id, così la copia delle skill avviene dopo l'install di `@spartacus/skills` (core-libs/schematics/src/add-spartacus/ai-context.ts:L50-L74). La migrazione `add-or-update-ai-skills` usa lo stesso schema (core-libs/schematics/src/migrations/add-or-update-ai-skills/ai-context.utils.ts:L107-L114).
9. **ts-morph sopra il Tree.** Invece del `ts.Program` "nudo" di core, Spartacus crea un `Project` ts-morph il cui file system è `TreeFileSystem`, che inoltra letture e scritture al `Tree` (core-libs/schematics/src/shared/utils/program.ts:L11-L35, core-libs/schematics/src/shared/utils/tree-file-system.ts:L12-L157);

`saveAndFormat` organizza gli import, formatta e salva (cioè scrive nel `Tree`) (core-libs/schematics/src/shared/utils/program.ts:L37-L48).

10. **Workspace.** `getWorkspace` legge `/angular.json` o `/.angular.json` con `jsonc-parser` e lancia `SchematicsException` se non li trova (core-libs/schematics/src/shared/utils/workspace-utils.ts:L31-L82); `getDefaultProjectNameFromWorkspace` restituisce la prima chiave di `projects` (core-libs/schematics/src/shared/utils/workspace-utils.ts:L111-L115); `getProjectTsConfigPaths` di Spartacus deriva dall'utility Angular (copyright Google LLC) e legge sia `targets` sia `architect` (core-libs/schematics/src/shared/utils/project-tsconfig-paths.ts:L1-L6, core-libs/schematics/src/shared/utils/project-tsconfig-paths.ts:L17-L96).
11. **API per le altre librerie.** `core-libs/schematics/index.ts` esporta le utility per gli schematics delle feature lib, dichiarandole non stabili (core-libs/schematics/index.ts:L7-L13).

3.8 Spartacus: migrazioni `ng update`

- `migrations.json` ha 25 voci su 4 versioni: `2211.19.0` (10), `2211.36.0` (4), `221121.7.0` (10), `221121.22.0` (1, `add-or-update-ai-skills`) (core-libs/schematics/src/migrations/migrations.json:L2-L128).
- I nomi hanno prefissi numerici `00-`, `01-`, …: poiché il CLI ordina per versione e poi per nome (§3.5), i prefissi **fissano l'ordine** (core-libs/schematics/src/migrations/migrations.json:L3-L52, deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L127-L127).
- Molte migrazioni per versione (rename symbol, deprecazioni, dependency management) sono wrapper sottili che passano dati generati a un "mechanism" generico, es. `migrateRenamedSymbols(tree, GENERATED_RENAMED_SYMBOLS_DATA)` (core-libs/schematics/src/migrations/221121_7/rename-symbol/rename-symbol.ts:L7-L15), che con ts-morph sostituisce gli import e chiama `saveAndFormat` (core-libs/schematics/src/migrations/mechanism/rename-symbol/rename-symbol.ts:L14-L85). I dati sono file generati ("Generated file, don't update manually.") (core-libs/schematics/src/migrations/221121_7/rename-symbol/data/generated-rename-symbols.migration.ts:L8-L17). Altre contengono la logica direttamente (punto seguente).
- Alcune migrazioni modificano direttamente il workspace, es. `221121_7/angular-config` rimuove `index` dal target `build` di `angular.json` con `tree.overwrite` (core-libs/schematics/src/migrations/221121_7/angular-config/index.ts:L20-L71); `migrateDependencies` aggiorna `package.json` e pianifica `installPackageJsonDependencies()` (core-libs/schematics/src/migrations/mechanism/dependency-management/dependency-management.ts:L28-L55).
- Nessuna feature/integration lib dichiara `ng-update`: tutto è centralizzato in `@spartacus/schematics`, che con `packageGroup` trascina l'aggiornamento delle librerie (core-libs/schematics/package.json:L46-L89).
- I test creano un `SchematicTestRunner('migrations', <migrations.json>)` e invocano `runSchematic('<nome>', {}, tree)` (core-libs/schematics/src/migrations/221121_7/angular-config/index_spec.ts:L7-L50).

Il catalogo completo delle migrazioni e dei mechanism è in 43 — Schematics e migrazioni di Spartacus.

3.9 Spartacus: schematics delle feature/integration lib

32 librerie (20 feature-libs, 12 integration-libs) hanno una propria `schematics/collection.json` (glob `feature-libs/*/schematics/collection.json`, `integration-libs/*/schematics/collection.json`). Quella di `@spartacus/cart` ha `ng-add` (private, hidden) e `add`, entrambi su `./add-cart/index#addCartFeatures` (feature-libs/cart/schematics/collection.json:L3-L17). `addCartFeatures` valida l'installazione e riusa le utility di `@spartacus/schematics` (`analyzeApplication`, `addFeatures`, `addPackageJsonDependenciesForLibrary`, `finalizeInstallation`) (feature-libs/cart/schematics/add-cart/index.ts:L13-L43). Gli schematics di libreria sono compilati con un `tsc` separato (`build:schematics`) e inclusi nel pacchetto ng-packagr come **asset**

(schematics/**/*.json, schematics/**/*.js) (feature-libs/cart/package.json:L25-L27, feature-libs/cart/ng-package.json:L7-L7); gli script root build:<lib> eseguono prima build:schematics e poi nx build <lib> (package.json:L23-L23). build:libs costruisce core e storefrontlib, poi in parallelo build:schematics e build:user, quindi il resto (package.json:L35-L35).

3.10 Builder ed executor nel monorepo

1. **Niente** angular.json. I target sono definiti nei project.json Nx con chiave executor; nx.json imposta targetDefaults (cache, build.dependsOn: ["^build"]) e useInferencePlugins: false (nx.json:L6-L23). DA VERIFICARE: come Nx risolva ed esegua un executor Architect (incluso il path relativo ./tools/build-lib:...) non è stato verificato sul codice Nx, non scaricato.
2. **Censimento.** Nei 42 project.json con executor: nx:run-commands 70, @angular-eslint/builder:lint 41, @angular-devkit/build-angular:ng-packagr 19, ./tools/build-lib:declaration-merging 18, @angular-devkit/build-angular:karma 4, @angular-devkit/architect:concat 2, @angular-builders/custom-esbuild:application 1, @angular-builders/custom-esbuild:dev-server 1, @nx/jest:jest 1 (grep "executor" su core-libs feature-libs integration-libs projects tools, dossier F118).
3. **Librerie.** core, storefrontlib, cart e altre usano il builder locale ./tools/build-lib:declaration-merging; setup e assets usano direttamente @angular-devkit/build-angular:ng-packagr (core-libs/core/project.json:L8-L19, feature-libs/cart/project.json:L8-L19, core-libs/setup/project.json:L8-L19, core-libs/assets/project.json:L9-L9). I test unitari passano per nx:run-commands (npx vitest run --config vitest.config.ts, npm run test:schematics) (core-libs/core/project.json:L20-L26, feature-libs/cart/project.json:L20-L33). @spartacus/schematics non ha target build Nx: solo lint, test-jest, test-jest-update (core-libs/schematics/project.json:L6-L27); il build è lo script build:schematics (package.json:L50-L50).
4. **Builder locali.** tools/build-lib è un pacchetto di builder Architect ("builders": "builders.json") con augmented-types e declaration-merging (tools/build-lib/package.json:L2-L6, tools/build-lib/builders.json:L1-L14). declaration-merging è creato con createBuilder, esegue @angular-devkit/build-angular:ng-packagr via context.scheduleBuilder(…, {target: context.target}) e, se ha successo, riscrive nei .d.ts prodotti le declare module '<path relativo>' in declare module './<typings>' (tools/build-lib/declaration-merging/index.ts:L20-L60, tools/build-lib/declaration-merging/index.ts:L86-L137), come workaround del bug ng-packagr #3085 (tools/build-lib/declaration-merging/index.ts:L22-L28). augmented-types segue lo stesso schema e sposta il blocco tra /** AUGMENTABLE_TYPES_START */ e /** AUGMENTABLE_TYPES_END */ nel typings radice (tools/build-lib/augmented-types/index.ts:L20-L23, tools/build-lib/augmented-types/index.ts:L54-L64, tools/build-lib/augmented-types/index.ts:L90-L159).
5. **Architect.** createBuilder normalizza il risultato della funzione (BuilderOutput → of, AsyncIterable → fromAsyncIterable, altro → from) e restituisce un oggetto marcato con BuilderSymbol (deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js:L137-L149, deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js:L172-L178); se l'Observable si completa senza emettere l'output è { success: false } (deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js:L155-L156). BuilderContext offre scheduleBuilder (per nome di builder) e scheduleTarget (per target del workspace) (deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js:L75-L96, deps:devkit/angular-devkit-architect-0.2102.23/package/src/api.d.ts:L116-L168). Il pacchetto fornisce i builder true, false, allOf (parallelo) e concat (sequenziale) (deps:devkit/angular-devkit-architect-0.2102.23/package/builders/builders.json:L3-L24).
6. **App demo.** build usa @angular-builders/custom-esbuild:application con plugins: ["./projects/storefrontapp/esbuild/plugins.ts"], server e ssr.entry; serve usa custom-esbuild:dev-server (projects/storefrontapp/project.json:L9-L13, projects/storefrontapp/project.json:L142-L146, projects/storefrontapp/project.json:L218-L229). Il plugin copia le variabili CX_* in

`define['buildProcess.env']`, convertendo `'true'`/`'false'` in booleani (projects/storefrontapp/esbuild/plugins.ts:L10-L31). `build-csr` e `prerender` usano `@angular-devkit/architect:concat` per rieseguire `storefrontapp:build` con configurazioni combinate (projects/storefrontapp/project.json:L190-L217): `concat` divide la stringa in `project:target:configuration` e chiama `scheduleTarget` in sequenza, fermandosi al primo `success: false` (deps:devkit/angular-devkit-architect-0.2102.23/package/builders/concat.js:L12-L45). Il target `test` dell'app usa `@angular-devkit/build-angular:karma` (projects/storefrontapp/project.json:L230-L253).

---

4. Diagrammi

4.1 Dove vive cosa

■ **TAVOLA 08.1** diagramma a tutta pagina, in fondo al capitolo

Ogni freccia corrisponde a §3.6 (core/localize, angular:packages/core/schematics/rollup.config.js:L56-L58, angular:packages/localize/schematics/ng-add/index.ts:L11-L19), §3.7 (core-libs/schematics/package.json:L26-L34, core-libs/schematics/src/shared/utils/new-module-utils.ts:L67-L75), §3.9 (feature-libs/cart/schematics/add-cart/index.ts:L13-L43) e §3.10 (tools/build-lib/declaration-merging/index.ts:L20-L60).

4.2 `ng add @spartacus/schematics`

■ **TAVOLA 08.2** diagramma a tutta pagina, in fondo al capitolo

Passi 2-3: deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L425-L426, :L473-L473; 4: deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L216-L225; 5-6: deps:devkit/angular-devkit-schematics-21.2.23/package/tools/node-module-engine-host.js:L46-L62, deps:devkit/angular-devkit-schematics-21.2.23/package/tools/export-ref.js:L16-L27; 7-8: deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L108-L124; 9-13: core-libs/schematics/src/ng-add/index.ts:L17-L24, core-libs/schematics/src/shared/utils/new-module-utils.ts:L40-L77, core-libs/schematics/src/add-spartacus/index.ts:L531-L592, core-libs/schematics/src/add-spartacus/ai-context.ts:L54-L74, core-libs/schematics/src/add-ssr/index.ts:L437-L465; 15: deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L77-L100; 16: deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L125-L133, deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L283-L297.

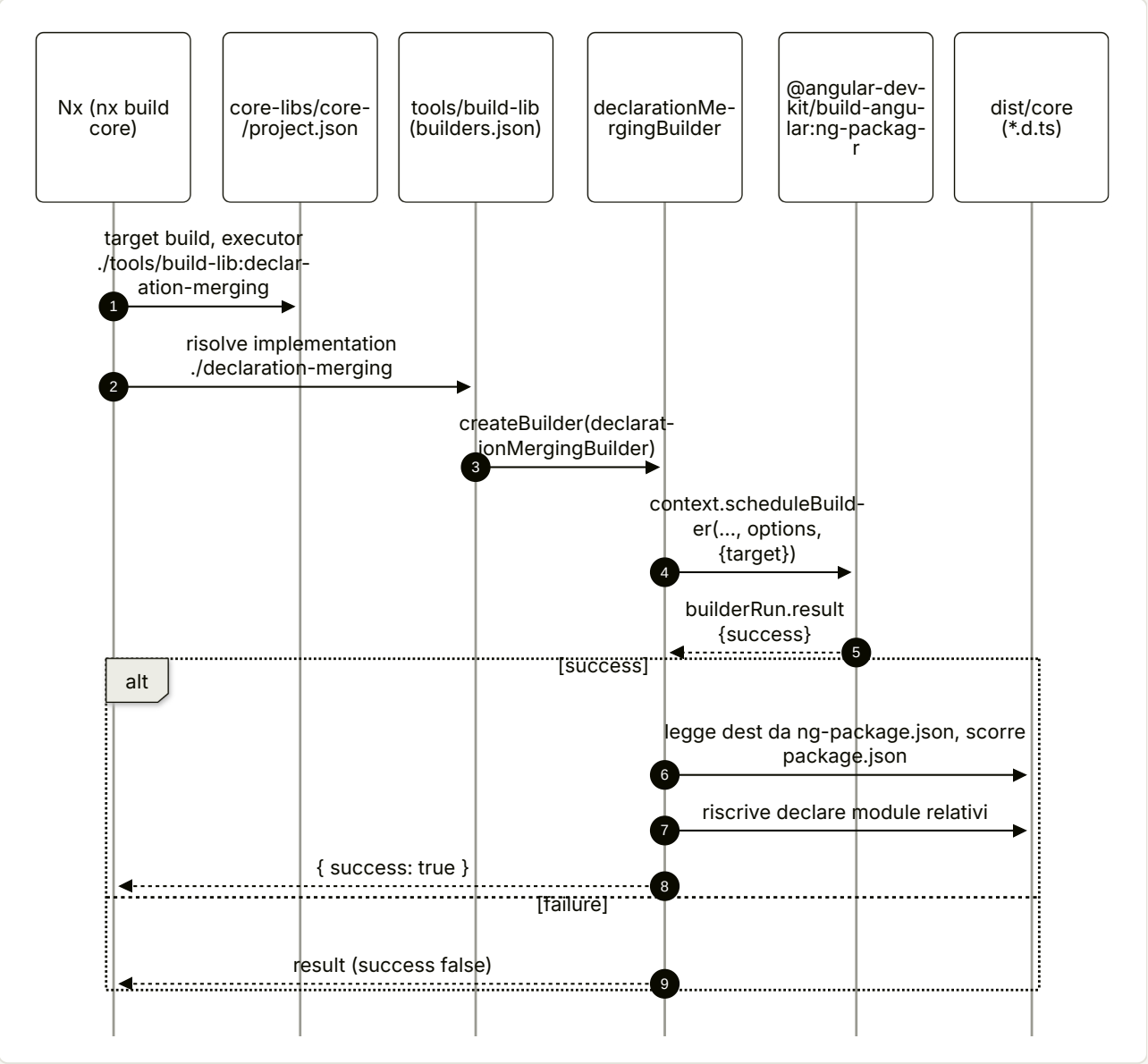
4.3 `ng update`: selezione e ordine delle migrazioni

■ **TAVOLA 08.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: B-C deps:devkit/angular-cli-21.2.23/package/src/commands/update/schematic/index.js:L252-L272, :L358-L375; D-E deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L440-L501; F-S2 deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L105-L142, :L232-L255, :L235-L243 (ramo no TTY), :L137-L140; M core-libs/schematics/src/migrations/221121_7/

rename-symbol/rename-symbol.ts:L11-L15, core-libs/schematics/src/shared/utils/program.ts:L11-L35; P
deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L166-L190.

4.4 `nx build core`: builder locale sopra ng-packagr



Riferimenti: core-libs/core/project.json:L8-L19, tools/build-lib/builders.json:L8-L12, tools/build-lib/package.json:L6-L6, tools/build-lib/declaration-merging/index.ts:L20-L60, tools/build-lib/declaration-merging/index.ts:L65-L137. Il passo 2 (risoluzione da parte di Nx) è DA VERIFICARE sul codice Nx (vedi §3.10).

---

5. Estratti di codice reale

5.1 Il contratto Rule / Source / Context

deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/interface.d.ts:L119-L154

```ts
export interface TypedSchematicContext<CollectionMetadataT extends object, SchematicMetadataT extends object> {
    readonly debug: boolean;
    readonly engine: Engine<CollectionMetadataT, SchematicMetadataT>;
    readonly logger: logging.LoggerApi;
    readonly schematic: Schematic<CollectionMetadataT, SchematicMetadataT>;
    readonly strategy: MergeStrategy;
    readonly interactive: boolean;
    addTask<T extends object>(task: TaskConfigurationGenerator<T>, dependencies?: Array<TaskId>): TaskId;
}
// ...
export type RuleFactory<T extends object> = (options: T) => Rule;
// ...
export type Source = (context: SchematicContext) => Tree | Observable<Tree>;
export type Rule = (tree: Tree, context: SchematicContext) => Tree | Observable<Tree> | Rule | Promise<void | Tree | Rule> | void;
```

5.2 Come viene normalizzato il valore di una Rule

deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/call.js:L76-L92

```js
async function callRuleAsync(rule, tree, context) {
    let result = await rule(tree, context);
    while (typeof result === 'function') {
        // This is considered a Rule, chain the rule and return its output.
        result = await result(tree, context);
    }
    if (typeof result === 'undefined') {
        return tree;
    }
    if ((0, rxjs_1.isObservable)(result)) {
        result = await (0, rxjs_1.lastValueFrom)(result.pipe((0, rxjs_1.defaultIfEmpty)(tree)));
    }
    if (result && interface_1.TreeSymbol in result) {
        return result;
    }
    throw new InvalidRuleResultException(result);
}
```

5.3 `chain` e noop

deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js:L46-L78

```js
function chain(rules) {
    return async (initialTree, context) => {
        let intermediateTree;
```

```
        if (Symbol.asyncIterator in rules) {
            for await (const rule of rules) {
                intermediateTree = (0, call_1.callRule)(rule, intermediateTree ?? initialTree, context);
            }
        }
        else {
            for (const rule of rules) {
                intermediateTree = (0, call_1.callRule)(rule, intermediateTree ?? initialTree, context);
            }
        }
        return () => intermediateTree;
    };
}
// ...
function noop() {
    return () => { };
}
```

5.4 `schematic()` vs `externalSchematic()`

deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/schematic.js:L22-L49

JS

```
function externalSchematic(collectionName, schematicName, options, executionOptions) {
    return (input, context) => {
        const collection = context.engine.createCollection(collectionName,
context.schematic.collection);
        const schematic = collection.createSchematic(schematicName);
        return schematic.call(options, (0, rxjs_1.of)((0, static_1.branch)(input)), context,
executionOptions).pipe((0, rxjs_1.last)(), (0, rxjs_1.map)((x) => {
            input.merge(x, interface_1.MergeStrategy.AllowOverwriteConflict);
            return input;
        }));
    };
}
// ...
function schematic(schematicName, options, executionOptions) {
    return (input, context) => {
        const collection = context.schematic.collection;
        const schematic = collection.createSchematic(schematicName, true);
        return schematic.call(options, (0, rxjs_1.of)((0, static_1.branch)(input)), context,
executionOptions).pipe((0, rxjs_1.last)(), (0, rxjs_1.map)((x) => {
            // We allow overwrite conflict here because they're the only merge conflict we particularly
            // don't want to deal with; the input tree might have an OVERWRITE which the sub
            input.merge(x, interface_1.MergeStrategy.AllowOverwriteConflict);
            return input;
        }));
    };
}
```

5.5 Sink: dry-run, errori, disco

deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L77-L100

```JS
    _createSinks() {
        let error = false;
        const dryRunSink = new dryrun_1.DryRunSink(this._host, this._force);
        const dryRunSubscriber = dryRunSink.reporter.subscribe((event) => {
            this._reporter.next(event);
            error = error || event.kind == 'error';
        });
        // We need two sinks if we want to output what will happen, and actually do the work.
        return [
            dryRunSink,
            // Add a custom sink that clean ourselves and throws an error if an error happened.
            {
                commit() {
                    dryRunSubscriber.unsubscribe();
                    if (error) {
                        return (0, rxjs_1.throwError)(new exception_1.UnsuccessfulWorkflowExecution());
                    }
                    return (0, rxjs_1.of)();
                },
            },
            // Only add a HostSink if this is not a dryRun.
            ...(!this._dryRun ? [new host_1.HostSink(this._host, this._force)] : []),
        ];
    }
```

5.6 Selezione e ordinamento delle migrazioni

deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L105-L128

```JS
async function executeMigrations(workflow, logger, packageName, collectionPath, from, to, commit =
false) {
    const collection = workflow.engine.createCollection(collectionPath);
    const migrationRange = new semver.Range('>' + (semver.prerelease(from) ? from.split('-')[0] + '-0' :
from) + ' <=' + to.split('-')[0]);
    const requiredMigrations = [];
    const optionalMigrations = [];
    for (const name of collection.listSchematicNames()) {
        const schematic = workflow.engine.createSchematic(name, collection);
        const description = schematic.description;
        description.version = (0, cli_version_1.coerceVersionNumber)(description.version);
        if (!description.version) {
            continue;
        }
        if (semver.satisfies(description.version, migrationRange, { includePrerelease: true })) {
            (description.optional ? optionalMigrations : requiredMigrations).push(description);
```

```
        }
    }
    // ...
        requiredMigrations.sort((a, b) => semver.compare(a.version, b.version) ||
a.name.localeCompare(b.name));
```

5.7 Una migrazione di `@angular/core`

angular:packages/core/schematics/migrations/application-config-core/index.ts:L9-L20

```ts
import {Rule} from '@angular-devkit/schematics';
import {ApplicationConfigCoreMigration} from './application-config-core-migration';
import {runMigrationInDevkit} from '../../utils/tsurge/helpers/angular_devkit';

export function migrate(): Rule {
  return async (tree) => {
    await runMigrationInDevkit({
      tree,
      getMigration: () => new ApplicationConfigCoreMigration(),
    });
  };
}
```

angular:packages/core/schematics/utils/tsurge/helpers/angular_devkit/run_in_devkit.ts:L128-L136

```ts
  for (const [file, changes] of replacementsPerFile) {
    const recorder = config.tree.beginUpdate(file);
    for (const c of changes) {
      recorder
        .remove(c.data.position, c.data.end - c.data.position)
        .insertRight(c.data.position, c.data.toInsert);
    }
    config.tree.commitUpdate(recorder);
  }
```

5.8 Il "manifesto" di `@spartacus/schematics`

core-libs/schematics/package.json:L43-L90

```json
  "ng-add": {
    "save": "devDependencies"
  },
  "ng-update": {
    "migrations": "./src/migrations/migrations.json",
    "packageGroup": [
      "@spartacus/core",
      "@spartacus/storefront",
```

```
      // ...
      "@spartacus/user"
    ]
  },
  "schematics": "./src/collection.json"
```

core-libs/schematics/src/collection.json:L3-L17

```json
  "schematics": {
    "ng-add": {
      "factory": "./ng-add/index",
      "description": "Install and configure spartacus",
      "schema": "./add-spartacus/schema.json",
      "private": true,
      "hidden": true,
      "aliases": ["install"]
    },
    "add-spartacus": {
      "description": "Add spartacus to Angular project.",
      "factory": "./add-spartacus/index#addSpartacus",
      "schema": "./add-spartacus/schema.json",
      "aliases": ["spa"]
    },
```

5.9 ng-add di Spartacus

core-libs/schematics/src/ng-add/index.ts:L17-L24

```ts
export default function (options: SpartacusOptions): Rule {
  return (host: Tree, context: SchematicContext) => {
    return chain([
      schematic('add-spartacus', options),
      options.ssr ? schematic('add-ssr', options) : noop(),
    ])(host, context);
  };
}
```

5.10 Task dipendenti: install, poi `ai-context`

core-libs/schematics/src/add-spartacus/ai-context.ts:L50-L69

```ts
/**
 * Runs the `ai-context` copy schematic only after the install task, so
 * `@spartacus/skills` is present in `node_modules`.
 */
export function scheduleAiContext(options: SpartacusOptions): Rule {
  return (_tree: Tree, context: SchematicContext): void => {
```

```
  const targets = normalize(options.aiTools);
  if (targets.length === 0) {
    return;
  }

  const installTaskId = context.addTask(new NodePackageInstallTask());

  context.addTask(
    new RunSchematicTask('ai-context', {
      aiTools: targets,
      debug: options.debug,
    }),
    [installTaskId]
  );
```

5.11 Delegare a `@schematics/angular`

core-libs/schematics/src/shared/utils/new-module-utils.ts:L50-L76

```ts
return (host: Tree): Rule => {
  const modulePath = `${getSourceRoot(host, { project: options.project })}/${
    options.path
  }`;
  const filePath = `${modulePath}/${dasherize(options.name)}.module.ts`;
  if (host.exists(filePath)) {
    const moduleFile = getTsSourceFile(host, filePath);
    const metadata = getDecoratorMetadata(
      moduleFile,
      'NgModule',
      ANGULAR_CORE
    )[0];

    if (metadata) {
      return noop();
    }
  }
  return externalSchematic(ANGULAR_SCHEMATICS, 'module', {
    project: options.project,
    name: dasherize(options.name),
    flat: true,
    commonModule: false,
    path: modulePath,
    module: options.module,
    typeSeparator: '.',
  });
};
```

5.12 ts-morph sul Tree

core-libs/schematics/src/shared/utils/program.ts:L11-L35

```ts
export function createProgram(
  tree: Tree,
  basePath: string,
  tsconfigPath: string
): { program: Project; appSourceFiles: SourceFile[] } {
  const fs = new TreeFileSystem(tree, basePath);
  const program = new Project({
    tsConfigFilePath: tsconfigPath,
    fileSystem: fs,
  });
  program.getTypeChecker();

  const appSourceFiles = program.getSourceFiles().filter((sourceFile) => {
    return (
      !sourceFile.isDeclarationFile() &&
      !sourceFile.isFromExternalLibrary() &&
      !sourceFile.isInNodeModules()
    );
  });

  return {
    program,
    appSourceFiles,
  };
}
```

5.13 Una migrazione Spartacus: wrapper + mechanism

core-libs/schematics/src/migrations/221121_7/rename-symbol/rename-symbol.ts:L7-L15

```ts
import { Rule, Tree } from '@angular-devkit/schematics';
import { migrateRenamedSymbols } from '../../mechanism/rename-symbol/rename-symbol';
import { GENERATED_RENAMED_SYMBOLS_DATA } from './data/generated-rename-symbols.migration';

export function migrate(): Rule {
  return (tree: Tree) => {
    return migrateRenamedSymbols(tree, GENERATED_RENAMED_SYMBOLS_DATA);
  };
}
```

5.14 Il builder locale `declaration-merging`

tools/build-lib/declaration-merging/index.ts:L20-L60

```ts
export default createBuilder(declarationMergingBuilder);

/**
 * Builder that runs default ng-packagr builder ('@angular-devkit/build-angular:ng-packagr')
 * and performs additional post step to fix paths in declaration merges.
 *
 * It's a workaround to fix paths in declaration merges, reference issues:
 *   - https://github.com/ng-packagr/ng-packagr/issues/3085
 */
function declarationMergingBuilder(
  options: NgPackagrBuilderOptions & JsonObject,
  context: BuilderContext
): Observable<BuilderOutput> {
  return from(ngPackagrBuild(context, options)).pipe(
    switchMap((result) =>
      result.success
        ? from(
            declarationMergingPostStep(
              context,
              options as NgPackagrBuilderOptions
            )
          )
        : of(result)
    )
  );
}

/**
 * Run ng packager build step as is
 */
async function ngPackagrBuild(
  context: BuilderContext,
  options: NgPackagrBuilderOptions & JsonObject
): Promise<BuilderOutput> {
  const builderRun = await context.scheduleBuilder(
    '@angular-devkit/build-angular:ng-packagr',
    options,
    { target: context.target }
  );
  return await builderRun.result;
}
```

tools/build-lib/builders.json:L1-L14

```json
{
  "builders": {
    "augmented-types": {
      "implementation": "./augmented-types",
```

```
      "schema": "./augmented-types/schema.json",
      "description": "Propagate typing exports to main d.ts file to make them augmentable."
    },
    "declaration-merging": {
      "implementation": "./declaration-merging",
      "schema": "./declaration-merging/schema.json",
      "description": "Fix relative paths used for declaration merging in flattened .d.ts file"
    }
  }
}
```

5.15 Il target Nx che lo usa

core-libs/core/project.json:L8-L19

```json
    "build": {
      "executor": "./tools/build-lib:declaration-merging",
      "options": {
        "tsConfig": "core-libs/core/tsconfig.lib.json",
        "project": "core-libs/core/ng-package.json"
      },
      "configurations": {
        "production": {
          "tsConfig": "core-libs/core/tsconfig.lib.prod.json"
        }
      }
    },
```

5.16 Il builder `concat` e il suo uso nell'app demo

deps:devkit/angular-devkit-architect-0.2102.23/package/builders/concat.js:L12-L21

```js
const builder = (0, src_1.createBuilder)((options, context) => {
    const allRuns = [];
    context.reportProgress(0, (options.targets ? options.targets.length : 0) +
        (options.builders ? options.builders.length : 0));
    if (options.targets) {
        allRuns.push(...options.targets.map(({ target: targetStr, overrides }) => {
            const [project, target, configuration] = targetStr.split(/:/g, 3);
            return () => context.scheduleTarget({ project, target, configuration }, overrides || {});
        }));
    }
```

projects/storefrontapp/project.json:L190-L200

```json
"build-csr": {
  "executor": "@angular-devkit/architect:concat",
  "configurations": {
    "development": {
      "targets": [{ "target": "storefrontapp:build:development,noSsr" }]
    },
    "production": {
      "targets": [{ "target": "storefrontapp:build:production,noSsr" }]
    }
  },
  "defaultConfiguration": "production"
},
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`package.json#schematics`	string (path collection)	— (assente = pacchetto senza schematics)	`core-libs/schematics/package.json:L90-L90`; `angular:packages/core/package.json:L49-L49`	`deps:devkit/angular-devkit-schematics-21.2.23/package/tools/node-module-engine-host.js:L48-L57`
`package.json#ng-update.migrations`	string	—	`core-libs/schematics/package.json:L46-L47`	`deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L262-L270` (solo `--migrate-only`), `deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L479-L488` (flusso normale)
`package.json#ng-update.packageGroup`	string[] | map	—	`core-libs/schematics/package.json:L48-L88`	`deps:devkit/angular-cli-21.2.23/package/src/commands/update/schematic/index.js:L284-L306`
`package.json#ng-add.save`	`'devDependencies'` | bool	dependencies	`core-libs/schematics/package.json:L43-L45`	`deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L426-L426`
collection `factory`	`"path#export"`	export `default`	`core-libs/schematics/src/collection.json:L5-L5`	`deps:devkit/angular-devkit-schematics-21.2.23/package/tools/export-ref.js:L17-L22`
collection `schema`	path JSON Schema	nessuna validazione	`core-libs/schematics/src/collection.json:L7-L7`	`deps:devkit/angular-devkit-schematics-21.2.23/package/tools/workflow/node-workflow.js:L56-L58`, `deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L160-L164`
collection `aliases`	string[]	[]	`core-libs/schematics/src/collection.json:L10-L10`	`deps:devkit/angular-devkit-schematics-21.2.23/package/tools/file-system-engine-host-base.js:L146-L153`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
collection `private` / `hidden`	boolean	false	`core-libs/schematics/src/collection.json:L8-L9`	`deps:devkit/angular-devkit-schematics-21.2.23/package/tools/file-system-engine-host-base.js:L90-L90`, `deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L249-L251`
migrations `version`	semver string	— (senza versione la voce è ignorata)	`core-libs/schematics/src/migrations/migrations.json:L4-L4`	`deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L113-L119`
migrations `optional`	boolean	false	`angular:packages/core/schematics/migrations.json:L12-L12`	`deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L118-L118`
schema `x-prompt`	string | object	—	`core-libs/schematics/src/add-spartacus/schema.json:L94-L96`	`deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L194-L195` (solo interattivo + TTY)
schema `$default.$source: projectName`	—	progetto corrente	`core-libs/schematics/src/add-spartacus/schema.json:L10-L12`	`deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L179-L179`
option `ssr` (Spartacus)	boolean	false	`core-libs/schematics/src/add-spartacus/schema.json:L381-L385`	`core-libs/schematics/src/ng-add/index.ts:L21-L21`
option `aiTools`	`('agents'|'claude')[]`	`["agents"]`	`core-libs/schematics/src/add-spartacus/schema.json:L390-L414`	`core-libs/schematics/src/add-spartacus/ai-context.ts:L56-L59`
`builders.json` (`implementation`, `schema`)	—	—	`tools/build-lib/builders.json:L3-L12`	dichiarato da `tools/build-lib/package.json:L6-L6`
`project.json#targets.<t>.executor`	`"pkg:builder"` o `"./path:builder"`	—	`core-libs/core/project.json:L9-L9`	Nx **DA VERIFICARE** (codice Nx non letto)
`nx.json#targetDefaults.build.dependsOn`	string[]	—	`nx.json:L7-L11`	Nx **DA VERIFICARE**
opzioni `plugins` del builder custom-esbuild	string[]	—	`projects/storefrontapp/project.json:L13-L13`	`@angular-builders/custom-esbuild` (non scaricato)

7. Commenti e note notevoli dal sorgente

- "Most `ng-add` schematics are marked as private." — `deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L218-L218`. Spiega perché `ng add` usa `createSchematic(…, true)` invece di `listSchematicNames`.
- "We obfuscate the context of Source and Rule because the schematic implementation should not know which types is the schematic or collection metadata, as they are both tooling specific." — `deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/interface.d.ts:L150-L151`.
- "This is used by the Schematics implementations in order to avoid needing to have typing from the tooling. Schematics are not specific to a tool." — `deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/interface.d.ts:L129-L130`. Gli schematics sono pensati indipendenti dal CLI: lo stesso `Tree`/`Rule` funziona con il CLI, i test runner, Nx.

- "We allow overwrite conflict here because they're the only merge conflict we particularly don't want to deal with" — deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/schematic.js:L43-L44.
- "Our migrations are set up to create a Program from the project's tsconfig … We can't migrate files outside of the project, because our file system interactions go through the CLI's Tree" — angular:packages/core/schematics/utils/typescript/compiler_host.ts:L125-L129.
- "Migrations should never have 2+ compilation units with overlapping source files" — angular:packages/core/schematics/utils/tsurge/helpers/angular_devkit/run_in_devkit.ts:L141-L173.
- "NOTE: it exports utils for other Spartacus libs' schematics, but is not intended as a public API for external use. It may change without deprecation." — core-libs/schematics/index.ts:L9-L10.
- "Runs the ai-context copy schematic only after the install task, so @spartacus/skills is present in node_modules." — core-libs/schematics/src/add-spartacus/ai-context.ts:L50-L53.
- "TODO:#schematics - fix the interactivity for the CDS / ASM, etc." — core-libs/schematics/src/shared/utils/feature-utils.ts:L134-L134.
- "Force installing versions of dependencies used by Spartacus. E.g. ng13 uses rxjs 7, but Spartacus uses rxjs 6." — core-libs/schematics/src/add-spartacus/index.ts:L579-L580. Commento datato (vedi §9).
- "To avoid introducing to the @spartacus/schematics a new library @angular-devkit/build-angular … as any was used" — core-libs/schematics/src/migrations/221121_7/angular-config/index.ts:L34-L35.
- "Generated file, don't update manually." — core-libs/schematics/src/migrations/221121_7/rename-symbol/data/generated-rename-symbols.migration.ts:L8-L10.
- "It's a workaround to make TS types augmentable" (TypeScript #9532, #18877) — tools/build-lib/augmented-types/index.ts:L25-L32.
- "It's a workaround to fix paths in declaration merges" (ng-packagr #3085) — tools/build-lib/declaration-merging/index.ts:L22-L28.

8. Extension point

@spartacus/schematics non ha un public_api.ts Angular: la sua superficie "pubblica" è core-libs/schematics/index.ts, dichiarata non stabile (core-libs/schematics/index.ts:L7-L13). Le API del devkit sono di terze parti (Angular CLI).

EXTENSION POINT	COME	RIFERIMENTO
Collection propria di una libreria	campo schematics nel package.json + collection.json con ng-add privato	feature-libs/cart/package.json:L54-L54, feature-libs/cart/schematics/collection.json:L1-L18
Riuso delle utility Spartacus negli schematics di libreria	addFeatures, addPackageJsonDependenciesForLibrary, finalizeInstallation, LibraryOptions importati da @spartacus/schematics	core-libs/schematics/index.ts:L13-L13, feature-libs/cart/schematics/add-cart/index.ts:L13-L43
Descrivere una feature installabile	oggetto SchematicConfig	core-libs/schematics/src/shared/utils/lib-utils.ts:L115-L199
Invocazione da riga di comando	ng g @spartacus/schematics:wrapper-module, :add-cms-component (cms), :ai-context (ai)	core-libs/schematics/src/collection.json:L24-L40
Comporre schematics esterni	externalSchematic('@schematics/angular', …), externalSchematic('@angular/ssr', 'ng-add', …)	core-libs/schematics/src/shared/utils/new-module-utils.ts:L67-L75, core-libs/schematics/src/add-ssr/index.ts:L437-L465

EXTENSION POINT	COME	RIFERIMENTO
Lavoro dopo l'install	`context.addTask(new NodePackageInstallTask())` + `RunSchematicTask` con dipendenze	core-libs/schematics/src/add-spartacus/ai-context.ts:L61-L69
Builder Architect custom	pacchetto con `builders.json` + `createBuilder` + `scheduleBuilder`	tools/build-lib/builders.json:L1-L14, tools/build-lib/declaration-merging/index.ts:L20-L60
Plugin esbuild nell'app	opzione `plugins` di `@angular-builders/custom-esbuild:application`	projects/storefrontapp/project.json:L13-L13, projects/storefrontapp/esbuild/plugins.ts:L10-L31
Sequenze di target	`@angular-devkit/architect:concat` con `targets` o `builders`	projects/storefrontapp/project.json:L190-L217, deps:devkit/angular-devkit-architect-0.2102.23/package/builders/concat.js:L12-L26

Ricette pertinenti in Parte III:

- 54 — API pubblica vs interna, upgrade, anti-pattern, checklist: come le migrazioni `ng update` di `@spartacus/schematics` (rename symbol, deprecazioni di costruttori/componenti) impattano il codice applicativo e cosa resta a carico dello sviluppatore.
- 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni: la struttura (root module, feature module, lazy chunk) che `addLibraryFeature`/`wrapper-module` generano in un'app quando si fa `ng add @spartacus/<lib>`.
- 50 — Ricette: configurazione, DI, interceptor, eventi: la configurazione `provideConfig` che gli schematics scrivono nei moduli `*-feature.module.ts`.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `ng-add` **non si invoca con** `ng generate`**.** È `private`: non compare in `listSchematicNames` e `ng generate` usa `allowPrivate = false`. Per ripetere l'installazione si usa `add-spartacus` (alias `spa`) (core-libs/schematics/src/collection.json:L8-L8, deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L112-L112, deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L249-L251).
2. **Il disco non cambia finché lo schematic non è finito.** Le regole lavorano su `HostTree`/`CordHost`; i sink scrivono alla fine e le task girano dopo (deps:devkit/angular-devkit-schematics-21.2.23/package/src/tree/host-tree.js:L85-L86, deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L77-L133). Conseguenza: durante le regole `node_modules` *non* contiene i pacchetti appena aggiunti a `package.json`, ed è per questo che `ai-context` è una `RunSchematicTask` dipendente dall'install (core-libs/schematics/src/add-spartacus/ai-context.ts:L50-L53).
3. **In dry-run nessuna task.** `--dry-run` non scrive e non esegue neppure `npm install` né `RunSchematicTask` (deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L125-L133).
4. **Install multipli.** Le task non sono deduplicate: `addSpartacus` pianifica `NodePackageInstallTask` in `installPackageJsonDependencies()` e di nuovo in `scheduleAiContext` (core-libs/schematics/src/add-spartacus/index.ts:L583-L587, core-libs/schematics/src/add-spartacus/ai-context.ts:L61-L61, deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L117-L130). `DA VERIFICARE` se l'executor del package manager del CLI accorpi esecuzioni ripetute (executor non letto).

5. **Sovrascritture silenziose.** `schematic()`/`externalSchematic()` fanno merge con `AllowOverwriteConflict`: le modifiche del figlio sugli stessi file vincono senza errore (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/schematic.js:L27-L27, deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/schematic.js:L45-L45).
6. **Ordine delle migrazioni = ordine alfabetico a parità di versione.** I prefissi `00-`…`09-` sono essenziali; `add-or-update-ai-skills` funziona senza prefisso solo perché è l'unico della sua versione (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L127-L127, core-libs/schematics/src/migrations/migrations.json:L123-L127). Attenzione: `localeCompare` con `10-` verrebbe prima di `2-` se si usassero prefissi non a due cifre.
7. **Nessun "un major alla volta" per Spartacus.** Il blocco del CLI vale solo per `@angular/*`/`@nguniversal/*`; per `@spartacus/*` contano solo le versioni nell'intervallo `(from, to]` (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/constants.js:L15-L15). Una voce senza `version` è ignorata (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L113-L116).
8. **Solo il primo progetto.** Le migrazioni Spartacus usano `getDefaultProjectNameFromWorkspace` = prima chiave di `projects`: in un workspace multi-progetto viene migrato solo il primo (core-libs/schematics/src/shared/utils/workspace-utils.ts:L111-L115, core-libs/schematics/src/migrations/mechanism/rename-symbol/rename-symbol.ts:L18-L20, core-libs/schematics/src/migrations/221121_7/angular-config/index.ts:L26-L29).
9. **Solo workspace Angular CLI.** Gli helper leggono solo `/angular.json` o `/.angular.json` (core-libs/schematics/src/shared/utils/workspace-utils.ts:L31-L31); il monorepo Spartacus stesso non ha `angular.json` (Nx), e i test generano un workspace con `@schematics/angular:workspace` (core-libs/schematics/src/shared/utils/generate-default-workspace.ts:L45-L60).
10. **File fuori dai tsconfig.** Core migra i file raggiunti dai tsconfig dei target `build` e `test` (angular:packages/core/schematics/utils/tsurge/helpers/angular_devkit/run_in_devkit.ts:L63-L68); il mechanism Spartacus `rename-symbol` usa solo il tsconfig di `build` del progetto di default (gli spec raggiunti solo dal tsconfig di test non sono toccati) (core-libs/schematics/src/migrations/mechanism/rename-symbol/rename-symbol.ts:L18-L24); i file fuori dalla root sono saltati (core) o letti come vuoti (`TreeFileSystem.readFileSync` restituisce `''`) (angular:packages/core/schematics/utils/typescript/compiler_host.ts:L111-L131, core-libs/schematics/src/shared/utils/tree-file-system.ts:L67-L78). `TreeFileSystem` non implementa `copySync`, `glob`, `globSync` e `mkdirSync` non crea nulla (core-libs/schematics/src/shared/utils/tree-file-system.ts:L100-L105, core-libs/schematics/src/shared/utils/tree-file-system.ts:L118-L121, core-libs/schematics/src/shared/utils/tree-file-system.ts:L149-L155).
11. **Schematics senza schema.** `wrapper-module` e `modernize-app-*` non dichiarano `schema`: niente validazione né default da JSON Schema (core-libs/schematics/src/collection.json:L30-L52); restano solo i default eventualmente configurati nel workspace (`angular.json` → `schematics`), che il CLI fonde nelle opzioni (deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L165-L173, deps:devkit/angular-devkit-schematics-21.2.23/package/tools/workflow/node-workflow.js:L51-L55).
12. **Regex "sbagliata" in** `declaration-merging`**.** `/declare module \'([^\@spartacus].+)\'/g` usa una classe di caratteri: esclude i moduli il cui primo carattere è uno tra `@ s p a r t c u`, non il prefisso `@spartacus` (tools/build-lib/declaration-merging/index.ts:L109-L111).
13. **Builder locali compilati a mano.** I builder di `tools/build-lib` sono usati tramite `index.js` versionati; nessuno script CI li ricompila: modificando `index.ts` serve rilanciare `tsc` (`"build": "tsc"`) (tools/build-lib/package.json:L7-L9). Il pacchetto dichiara solo `@angular-devkit/architect` ma importa anche `@angular-devkit/build-angular`, `@angular-devkit/core`, `glob`, `rxjs`, risolti dal `node_modules` di root (tools/build-lib/package.json:L18-L20, tools/build-lib/declaration-merging/index.ts:L7-L18).
14. `augmented-types` **non usato.** È definito ma nessun `project.json` lo referenzia; le librerie usano `declaration-merging`, pur con i marker `AUGMENTABLE_TYPES_START` nei public api (tools/build-lib/

builders.json:L3-L7, core-libs/core/public_api.ts:L38-L38).

15. **Schematics delle feature lib come asset.** Se `build:schematics` non gira prima di `nx build <lib>`, il pacchetto esce senza JS degli schematics (feature-libs/cart/ng-package.json:L7-L7, package.json:L23-L23).
16. **Target `test` assente nelle librerie core.** `core`/`storefrontlib` hanno `test-vitest`, `lint`, `build`, ma non `test`: il comando `nx run storefrontlib:test` citato in `.claude/CLAUDE.md` non corrisponde a un target attuale (core-libs/core/project.json:L7-L36, core-libs/storefront/project.json:L22-L24). Vedi 44 — Strategia di test e tooling del monorepo.
17. **Commento obsoleto su rxjs 6/ng13** in `addSpartacus` rispetto ad Angular 21 / RxJS 7.8 (core-libs/schematics/src/add-spartacus/index.ts:L579-L580).

---

10. Come fare debug

- **Vedere cosa farebbe uno schematic senza toccare il disco:** `--dry-run` usa solo il `DryRunSink` (report degli eventi) ed esclude `HostSink` e post-task (deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L77-L100, deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L125-L133).
- **Log interni Spartacus:** l'opzione `debug` di `LibraryOptions` abilita messaggi come il grafo delle feature in `addFeatures` (core-libs/schematics/src/shared/utils/lib-utils.ts:L85-L110, core-libs/schematics/src/shared/utils/feature-utils.ts:L107-L113).
- **"[Internal] No feature config found":** la feature non è registrata in `featureSchematicConfigMapping` (core-libs/schematics/src/shared/utils/feature-utils.ts:L125-L132).
- **"Could not find Angular":** manca `angular.json`/`.angular.json` nella root del `Tree` (core-libs/schematics/src/shared/utils/workspace-utils.ts:L31-L82).
- **Errore "does not support schematics":** il pacchetto non ha il campo `schematics` nel `package.json` (deps:devkit/angular-devkit-schematics-21.2.23/package/tools/node-module-engine-host.js:L46-L62).
- **Una migrazione non parte:** controllare `from`/`to` rispetto all'intervallo `(from, to]` e la presenza di `version` (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L105-L119); per eseguirle una per volta e isolarle, `--create-commits` crea un commit per migrazione (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L166-L190).
- **Testare uno schematic/migrazione:** `SchematicTestRunner` su `collection.json`/`migrations.json` e `runSchematic(name, options, tree)` (core-libs/schematics/src/migrations/221121_7/angular-config/index_spec.ts:L7-L50, angular:packages/core/schematics/test/application_config_core_spec.ts:L30-L34). Nel monorepo: `npm run test:schematics` (Jest su `core-libs/schematics`) e `test:all-schematics` per tutte le librerie (package.json:L145-L146), oppure `nx run schematics:test-jest` (core-libs/schematics/project.json:L13-L19).
- **Builder locale:** `declaration-merging` logga via `context.logger` ogni `declare module` trovata e corretta, e gli errori per `package.json` (tools/build-lib/declaration-merging/index.ts:L112-L136). Se il build ng-packagr fallisce, il post-step non gira e viene restituito l'output originale (tools/build-lib/declaration-merging/index.ts:L33-L44).
- `concat` **che restituisce** `success: false` **senza errori visibili:** un target figlio è terminato senza emettere output o con `success: false`; `concat` si ferma lì (deps:devkit/angular-devkit-architect-0.2102.23/package/builders/concat.js:L33-L45), e `createBuilder` trasforma un Observable vuoto in `{ success: false }` (deps:devkit/angular-devkit-architect-0.2102.23/package/src/create-builder.js:L155-L156).

11. Domande di autoverifica

1. **Perché il motore degli schematics non si trova in** `/tmp/angular`**?** Perché `@angular-devkit/schematics`, `@angular-devkit/architect`, `@schematics/angular` e `@angular/cli` vivono nel repo `angular/angular-cli`; il repo `angular/angular` contiene solo gli schematics dei pacchetti framework, che lasciano il devkit esterno al bundle. — angular:packages/core/schematics/rollup.config.js:L56-L58
2. **Come trova il CLI il** `collection.json` **di** `@spartacus/schematics`**?** Risolve `@spartacus/schematics/package.json`, legge il campo `schematics` (`./src/collection.json`) e lo risolve relativamente alla cartella del pacchetto. — deps:devkit/angular-devkit-schematics-21.2.23/package/tools/node-module-engine-host.js:L46-L62, core-libs/schematics/package.json:L90-L90
3. **Cosa succede se una** `Rule` **non restituisce nulla?** `callRuleAsync` interpreta `undefined` come "tree invariato" e restituisce il tree di ingresso; è il principio su cui si basa `noop()`. — deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/call.js:L82-L84, deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js:L76-L78
4. **Perché** `ai-context` **è una** `RunSchematicTask` **e non una semplice regola nella** `chain`**?** Perché le regole girano prima che i file siano scritti e prima di `npm install`; serve che `@spartacus/skills` sia in `node_modules`, quindi la copia è una task con dipendenza dall'id della task di install. — core-libs/schematics/src/add-spartacus/ai-context.ts:L50-L69, deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L125-L133
5. **In che ordine vengono eseguite due migrazioni Spartacus con la stessa** `version`**?** Per nome con `localeCompare`, quindi i prefissi `00-`, `01-`… determinano l'ordine. — deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L127-L127, core-libs/schematics/src/migrations/migrations.json:L3-L52
6. **Perché** `ng generate @spartacus/schematics:ng-add` **fallisce mentre** `ng add` **funziona?** `ng-add` è `private`; `ng add` usa `allowPrivateSchematics = true`, `ng generate` no, e `createSchematic` lancia `PrivateSchematicException`. — deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L98-L99, deps:devkit/angular-cli-21.2.23/package/src/command-builder/schematics-command-module.js:L112-L112, deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L249-L251
7. **Le feature lib hanno bisogno di dichiarare** `ng-update` **per ricevere migrazioni?** No: le migrazioni sono centralizzate in `@spartacus/schematics`, il cui `packageGroup` include le librerie `@spartacus/*`. — core-libs/schematics/package.json:L46-L89
8. **Cosa fa in più il builder** `./tools/build-lib:declaration-merging` **rispetto a ng-packagr?** Esegue ng-packagr con `scheduleBuilder` e, se ha successo, riscrive nei `.d.ts` le `declare module` con path relativo in `declare module './<typings>'` (workaround ng-packagr #3085). — tools/build-lib/declaration-merging/index.ts:L20-L60, tools/build-lib/declaration-merging/index.ts:L86-L137
9. **Come fa** `build-csr` **a riusare il target** `build` **dell'app senza SSR?** Usa `@angular-devkit/architect:concat` con `targets: [{target: "storefrontapp:build:production,noSsr"}]`; `concat` divide la stringa e chiama `scheduleTarget`. — projects/storefrontapp/project.json:L190-L200, deps:devkit/angular-devkit-architect-0.2102.23/package/builders/concat.js:L16-L20
10. **Perché gli schematics Spartacus usano ts-morph invece del** `ts.Program` **di core, e dove scrivono?** `createProgram` crea un `Project` ts-morph con `TreeFileSystem` come file system, che inoltra letture e scritture al `Tree`; `saveAndFormat` salva quindi nel `Tree` virtuale, non su disco. — core-libs/schematics/src/shared/utils/program.ts:L11-L48, core-libs/schematics/src/shared/utils/tree-file-system.ts:L12-L157

TAVOLA 08.1 **4.1 Dove vive cosa**

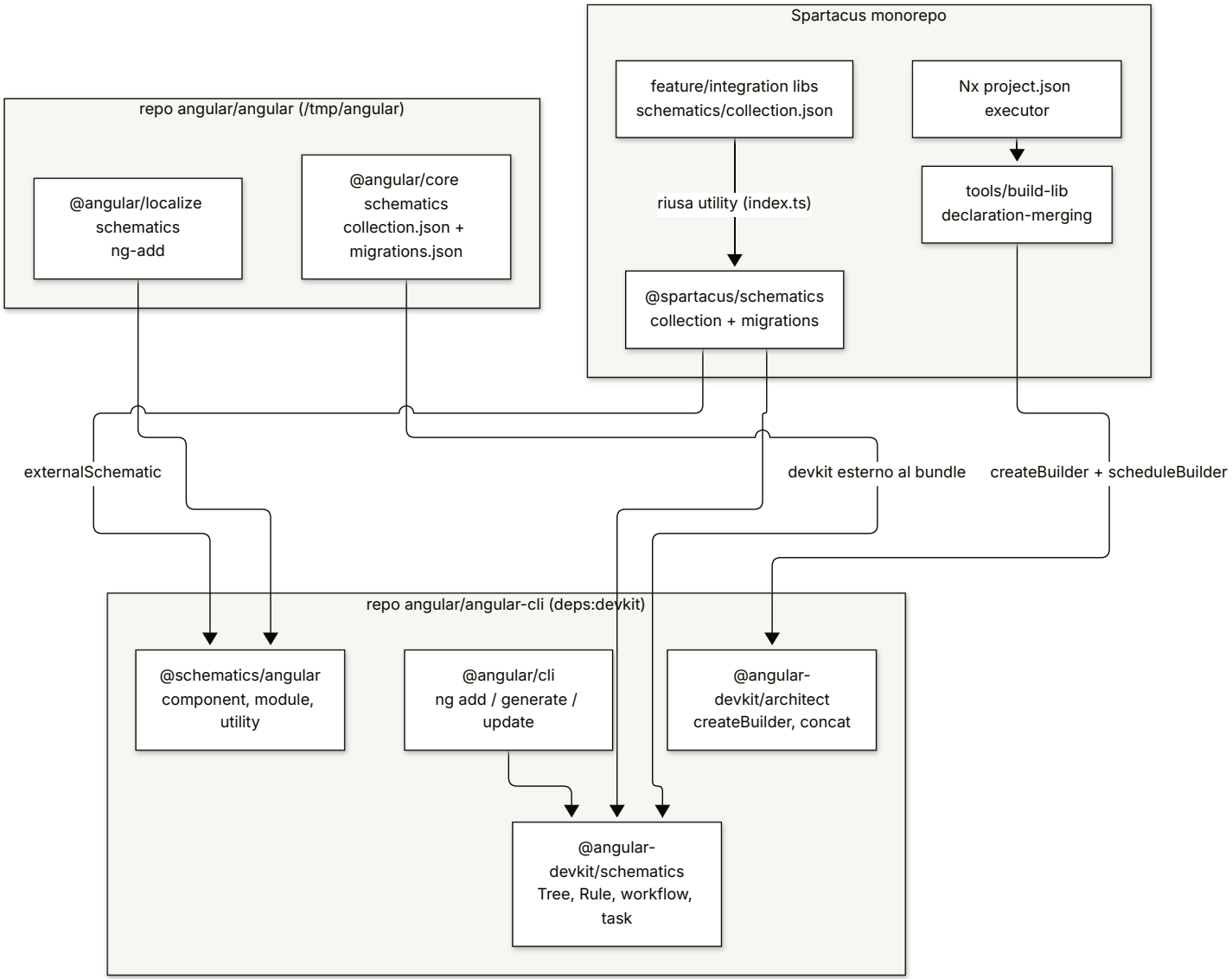


TAVOLA 08.2 **4.2 ng add @spartacus/schematics**

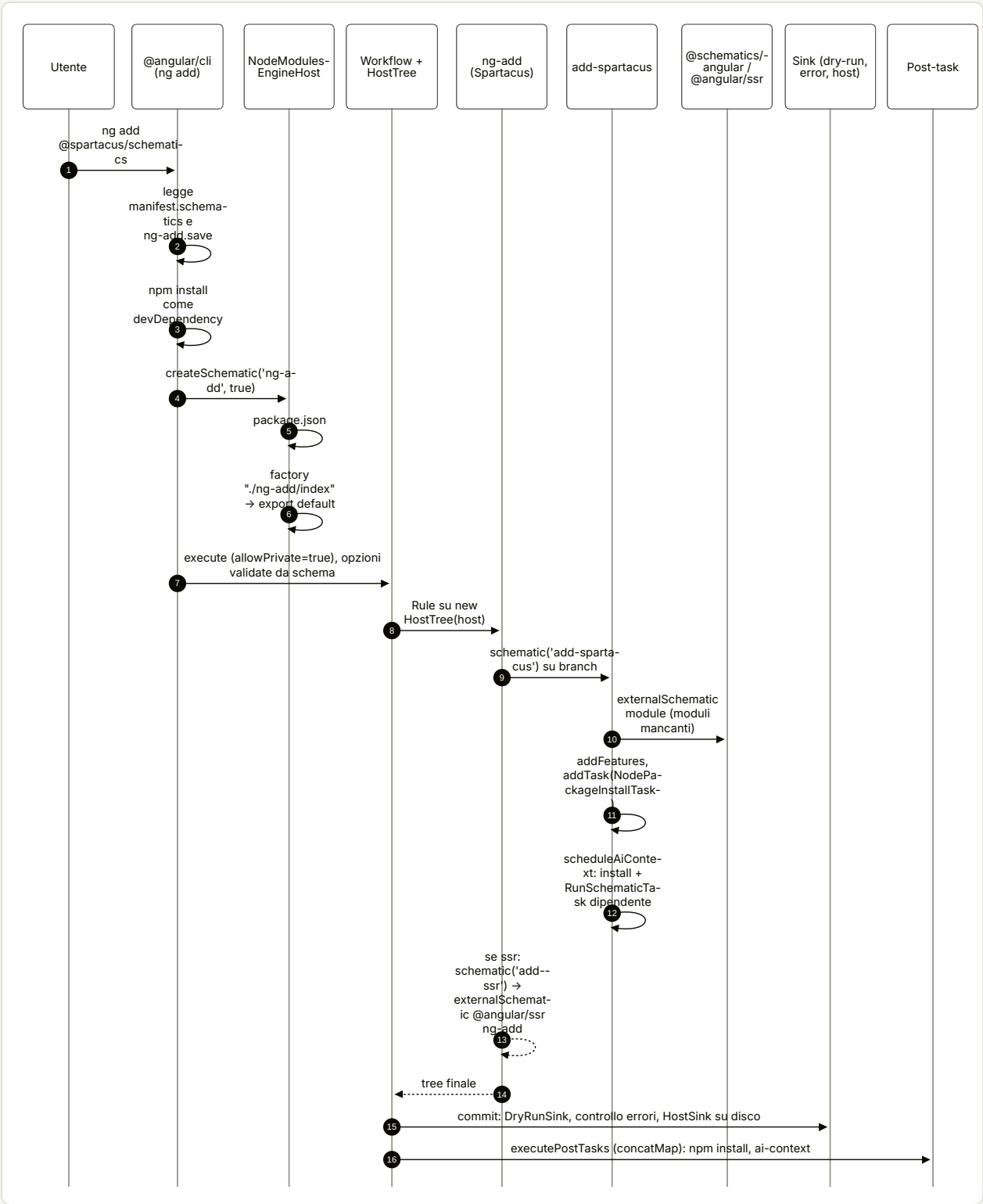
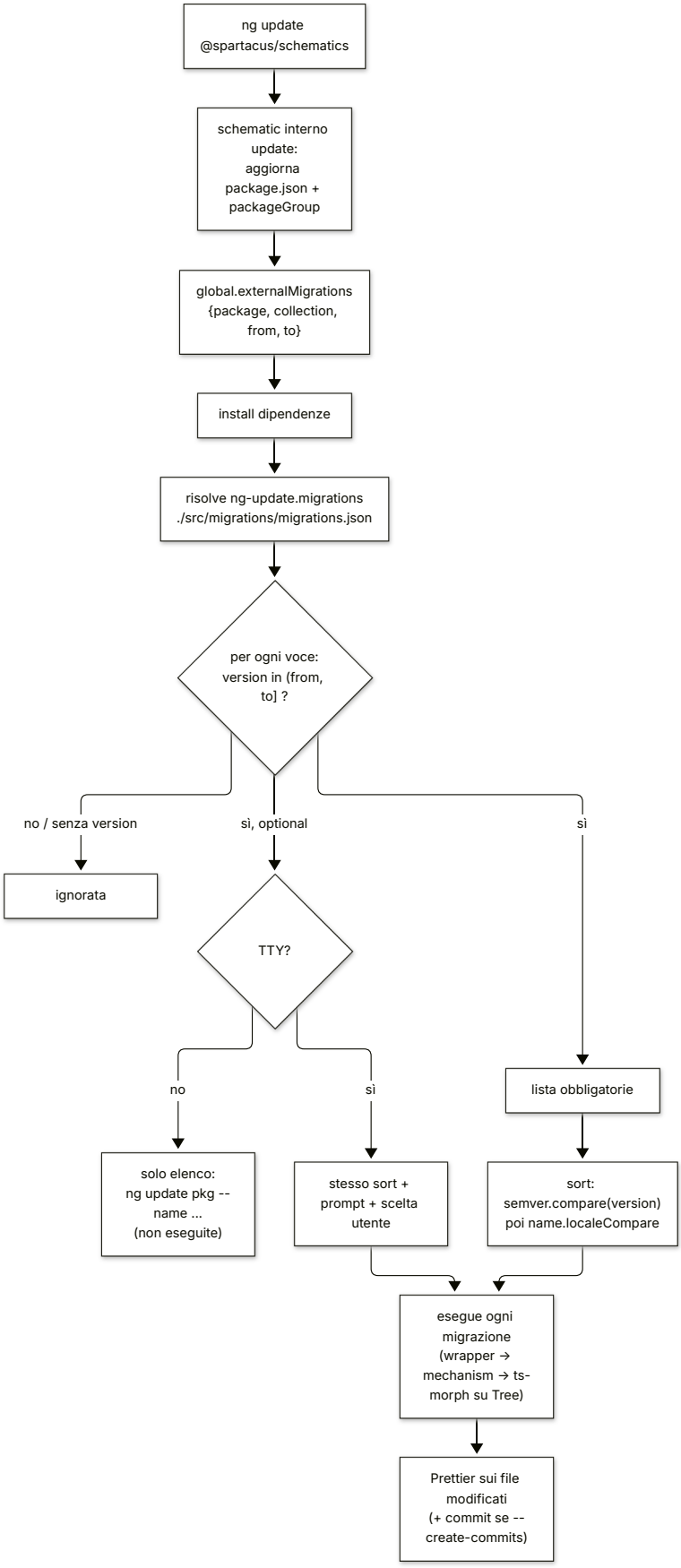


TAVOLA 08.3 **4.3 ng update: selezione e ordine delle migrazioni**



PARTE II

Spartacus sotto il cofano

10 Bootstrap e ordine di inizializzazione

11 Configurazione: provideConfig, default, merge, validator, initializer

12 Feature module e lazy loading in Spartacus

13 Site context e site theme

14 Routing: route semantiche, URL, page guard, contesto

15 CMS (core): pagine, slot, componenti, caricamento

16 CMS (storefront): rendering dinamico delle pagine

17 Outlet

18 Layout, breakpoint, launch dialog e accessibilità

19 State management: NgRx, loader/entity/process state

20 Facade, command & query, eventi

21 Livello OCC: adapter, connector, converter, endpoint

22 Autenticazione: OAuth, token, interceptor, refresh

23 Utente, profilo, consensi

24 Prodotto e ricerca

25 Carrello

26 Checkout

27 Ordini

28 B2B: organization

29 ASM (Assisted Service Module)

30 Product configurator

31 Quote, customer ticketing, subscription billing

32 Pickup in store, store finder, delivery date

33 Tracking, SmartEdit, Qualtrics

34 Integration libs: CDC, CDP, CDS, segment-refs

35 Integration libs: OPF, digital payments, OPPS

36 Integration libs: S/4, OMF, CPQ quote, punchout, EPD

37 Internazionalizzazione (i18n)

38 Stili e theming

10

Bootstrap e ordine di inizializzazione

Cosa deve sapere lo storefront prima di renderizzare (config, site context, rotte, auth) e come lo attende.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). I riferimenti prefissati `angular:` puntano a `/tmp/angular` @ `v21.2.23`.

Questo capitolo segue una storefront Spartacus dal primo byte di JavaScript eseguito (`main.ts`) fino al primo render della pagina CMS. È un capitolo "di attraversamento": tocca configurazione (cap. 11), lazy loading (cap. 12), site context (cap. 13), routing (cap. 14) e rendering CMS (cap. 16), ma solo per quanto serve a capire **in che ordine** le cose accadono e **chi aspetta chi**. I meccanismi Angular di base (injector, `bootstrapApplication`, router) sono trattati in Parte I: cap. 01, cap. 02, cap. 05.

1. A cosa serve e perché esiste

Una storefront Spartacus non può renderizzare nulla finché non sa:

1. **qual è la configurazione definitiva** — che può essere in parte statica (`provideConfig`) e in parte caricata dal backend in modo asincrono (i `CONFIG_INITIALIZER`);
2. **in quale site context si trova** — base site, lingua, valuta, che possono arrivare dalla configurazione statica, dal backend (OCC `basesites`), dall'URL o dallo storage;
3. **quali rotte esistono** — le rotte Angular sono vuote all'inizio (`RouterModule.forRoot([])`) e vengono completate a runtime con i path configurati e con una rotta wildcard CMS;
4. **qual è lo stato di autenticazione** — token persistiti, eventuale callback OAuth nell'URL.

Angular offre già i mattoni per "aspettare prima di partire": `ENVIRONMENT_INITIALIZER`, `APP_INITIALIZER` (oggi `provideAppInitializer`), `LOCATION_INITIALIZED` e la modalità `initialNavigation: 'enabledBlocking'` del router. Spartacus li compone e ci aggiunge tre meccanismi propri:

MECCANISMO SPARTACUS	COSA AGGIUNGE RISPETTO AD ANGULAR	DEFINITO IN
`CONFIG_INITIALIZER` + `ConfigInitializerService`	contributi di config asincroni, divisi per **scope** (`'context'`, `'routing'`, `'i18n'`…), con `getStable(...scopes)` per aspettare solo la parte di config che serve	`core-libs/core/src/config/config-initializer/config-initializer.ts:L18-L67`, `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L35-L175`
`LOCATION_INITIALIZED_MULTI`	trasforma il token singolo `LOCATION_INITIALIZED` di Angular in un multi-token, così più librerie possono trattenere la navigazione iniziale	`core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L30-L62`, `core-libs/core/src/routing/routing.module.ts:L66-L71`
`MODULE_INITIALIZER`	un "APP_INITIALIZER per moduli lazy"; nei moduli eager viene eseguito all'avvio	`core-libs/core/src/lazy-loading/tokens.ts:L9-L20`, `core-libs/core/src/lazy-loading/lazy-loading.module.ts:L18-L53`

Il punto di aggregazione di tutto questo è la coppia `BaseStorefrontModule` → `BaseCoreModule`: importandola, un'applicazione ottiene in un colpo solo l'intera catena di inizializzatori (`core-libs/storefront/base-storefront.module.ts:L26-L49`, `core-libs/core/src/base-core.module.ts:L26-L53`).

Il capitolo copre anche le **recipes** (`core-libs/storefront/recipes`, `core-libs/setup/recipes`) e i `public_api.ts` di `@spartacus/core`, `@spartacus/storefront`, `@spartacus/setup`, perché sono ciò che un'app tipica importa al bootstrap.

2. Mappa dei file

2.1 File assegnati al capitolo

PATH	RUOLO
core-libs/core/public_api.ts	Entry point di @spartacus/core: re-export di tutti i sotto-moduli (incluso ./src/base-core.module) e blocco AUGMENTABLE_TYPES_START/END (core-libs/core/public_api.ts:L10-L58).
core-libs/core/src/base-core.module.ts	BaseCoreModule: aggrega 17 moduli core in un ordine esplicito; forRoot() senza provider propri (core-libs/core/src/base-core.module.ts:L26-L53).
core-libs/storefront/base-storefront.module.ts	BaseStorefrontModule: importa BaseCoreModule.forRoot() per primo, poi infrastruttura CMS/layout/routing/media/outlet; nessun forRoot() (core-libs/storefront/base-storefront.module.ts:L26-L49).
core-libs/storefront/public_api.ts	Entry point di @spartacus/storefront: triple-slash reference a types.d.ts, export di base-storefront.module, recipes, router, blocco AUGMENTABLE_TYPES_START/END (core-libs/storefront/public_api.ts:L7-L36).
core-libs/storefront/recipes/index.ts, core-libs/storefront/recipes/config/index.ts	Barrel delle recipe storefront: default-media.config, layout-config, static-cms-structure (core-libs/storefront/recipes/config/index.ts:L7-L9).
core-libs/storefront/recipes/config/default-media.config.ts	mediaConfig: formati immagine e <picture> (core-libs/storefront/recipes/config/default-media.config.ts:L9-L38).
core-libs/storefront/recipes/config/layout-config.ts	layoutConfig (deprecato) e layoutConfigFactory() (header unificato, niente pageFold) (core-libs/storefront/recipes/config/layout-config.ts:L13-L198).
core-libs/storefront/recipes/config/static-cms-structure.ts	defaultCmsContentProviders: componenti CMS statici in PreHeader/SiteLogin (core-libs/storefront/recipes/config/static-cms-structure.ts:L10-L19).
core-libs/setup/public_api.ts	Entry point di @spartacus/setup: esporta solo ./recipes/index (core-libs/setup/public_api.ts:L7-L7).
core-libs/setup/recipes/index.ts, core-libs/setup/recipes/b2b/index.ts, core-libs/setup/recipes/b2b/config/index.ts	Barrel delle recipe B2B (core-libs/setup/recipes/index.ts:L7-L7, core-libs/setup/recipes/b2b/index.ts:L7-L7, core-libs/setup/recipes/b2b/config/index.ts:L7-L7).
core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts	defaultB2bOccConfig: endpoint OCC B2B (orgUsers/${userId}/...) (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L18-L49).

2.2 File di flusso attraversati (approfonditi in altri capitoli)

PATH	RUOLO NEL BOOTSTRAP
projects/storefrontapp/src/main.ts, projects/storefrontapp/src/main.server.ts	Entry point browser e server (bootstrapApplication).
projects/storefrontapp/src/app/app.config.ts, projects/storefrontapp/src/app/app.config.server.ts	ApplicationConfig browser e server.
projects/storefrontapp/src/app/app.module.ts	AppModule importato con importProvidersFrom.
projects/storefrontapp/src/app/spartacus/spartacus.module.ts e fratelli	Struttura "da schematics": BaseStorefrontModule + feature + configurazione.
projects/storefrontapp/src/app/private/private.providers.ts	Provider solo per sviluppo/e2e.

PATH	RUOLO NEL BOOTSTRAP
core-libs/storefront/router/app-routing.module.ts	AppRoutingModule: RouterModule.forRoot([], { initialNavigation: 'enabledBlocking' }).
core-libs/core/src/config/config-initializer/config-initializer.module.ts	APP_INITIALIZER che esegue i CONFIG_INITIALIZER + LOCATION_INITIALIZED_MULTI che attende la config stabile.
core-libs/core/src/config/config-initializer/config-initializer.service.ts	ConfigInitializerService: initialize(), getStable(), gestione scope.
core-libs/core/src/config/config-validator/config-validator.module.ts	APP_INITIALIZER di validazione (solo dev mode).
core-libs/core/src/routing/routing.module.ts	Override di LOCATION_INITIALIZED, APP_INITIALIZER delle rotte configurabili, CONFIG_INITIALIZER secure portal.
core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts	Token LOCATION_INITIALIZED_MULTI e helper.
core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts	router.resetConfig(...) con i path da routing.routes.
core-libs/storefront/cms-structure/routing/routing.module.ts, core-libs/storefront/cms-structure/routing/cms-route/cms-route.module.ts, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts	RoutingModule storefront e rotta wildcard CMS.
core-libs/core/src/site-context/site-context.module.ts e core-libs/core/src/site-context/providers/*	Provider e APP_INITIALIZER del site context.
core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts	CONFIG_INITIALIZER context da OCC (base sites).
core-libs/core/src/site-context/services/language-initializer.ts, core-libs/core/src/site-context/services/currency-initializer.ts, core-libs/core/src/site-context/services/base-site-initializer.ts, core-libs/core/src/site-context/services/site-context-routes-handler.ts	Impostazione dei valori attivi di contesto.
core-libs/core/src/lazy-loading/lazy-loading.module.ts, core-libs/core/src/lazy-loading/lazy-modules.service.ts, core-libs/core/src/lazy-loading/tokens.ts	MODULE_INITIALIZER.
core-libs/storefront/layout/main/storefront.component.ts, core-libs/storefront/layout/main/storefront.component.html	Root UI cx-storefront.
angular:packages/platform-browser/src/browser.ts, angular:packages/core/src/application/create_application.ts, angular:packages/core/src/platform/bootstrap.ts, angular:packages/core/src/application/application_init.ts, angular:packages/router/src/provide_router.ts	Lato Angular: creazione injector, initializer, bootstrap del root component, navigazione bloccante.

3. Come funziona passo per passo

Il percorso è diviso in otto fasi. Ogni fase indica il file e le righe che la rendono vera.

3.1 Fase 0 — Valutazione dei moduli ES (prima di qualunque DI)

Prima ancora che Angular entri in gioco, il bundler valuta i moduli importati da `main.ts`. Tre cose avvengono prima della DI:

- `enableProdMode()` viene chiamato solo se `environment.production`: non è un effetto della valutazione dei moduli importati, ma codice top-level di `main.ts` eseguito subito prima di `bootstrapApplication` (projects/storefrontapp/src/main.ts:L13-L17).
- `spartacus-configuration.module.ts` esegue `registerLocaleData` per `de`, `ja`, `zh` a livello di modulo, cioè al momento dell'import (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L29-L31).
- La scelta B2B/B2C è già fissata: `environment.b2b` deriva da `buildProcess.env.CX_B2B ?? false` (projects/storefrontapp/src/environments/environment.ts:L27-L27) e seleziona l'array di provider (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L40-L42).

3.2 Fase 1 — `bootstrapApplication` e appiattimento dei provider

1. `main.ts` chiama `bootstrapApplication(AppComponent, appConfig)`: l'app demo è standalone, non usa `bootstrapModule` (projects/storefrontapp/src/main.ts:L17-L23).
2. `appConfig` registra in ordine `provideHttpClient(withFetch(), withInterceptorsFromDi())`, `provideClientHydration(...)`, `provideZoneChangeDetection(...)`, `provideBrowserGlobalErrorListeners()` e infine `importProvidersFrom(AppModule)` (projects/storefrontapp/src/app/app.config.ts:L26-L35). Tutta Spartacus entra quindi attraverso un singolo `importProvidersFrom`.
3. Angular costruisce `{ rootComponent, ...createProvidersConfig(options, context) }` (angular:packages/platform-browser/src/browser.ts:L125-L140), anteponendo `BROWSER_MODULE_PROVIDERS` ai provider utente (angular:packages/platform-browser/src/browser.ts:L169-L175).
4. `importProvidersFrom(AppModule)` attiva il walk dell'albero dei moduli. L'algoritmo è "imports-first": per ogni NgModule vengono visitati prima gli `imports`; i provider dei `ModuleWithProviders` importati sono processati **dopo** tutti gli import del modulo contenitore; poi vengono aggiunti i provider dichiarati sul modulo stesso (angular:packages/core/src/di/provider_collection.ts:L293-L355). Per ogni NgModule viene anche aggiunto un `ENVIRONMENT_INITIALIZER` che fa `inject(defType)` — cioè istanzia eagerly la classe del modulo (angular:packages/core/src/di/provider_collection.ts:L338-L344) — e questo provider è registrato *dopo* gli import ("It's important that this is done after the def's imports", angular:packages/core/src/di/provider_collection.ts:L324-L344).

L'albero che viene percorso nell'app demo è:

- `AppModule` → `BrowserModule`, `StoreModule.forRoot({})`, `AppRoutingModule`, `EffectsModule.forRoot([])`, `SpartacusModule`, più `privateProviders` (projects/storefrontapp/src/app/app.module.ts:L15-L25).
- `AppRoutingModule` → `RouterModule.forRoot([], { anchorScrolling: 'enabled', initialNavigation: 'enabledBlocking' })` + `provideDefaultConfig(defaultOnNavigateConfig)` + APP_INITIALIZER `onNavigateFactory` (core-libs/storefront/router/app-routing.module.ts:L13-L37).
- `SpartacusModule` → `BaseStorefrontModule`, `SpartacusFeaturesModule`, `SpartacusConfigurationModule` (projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L20).
- `BaseStorefrontModule` → `BaseCoreModule.forRoot()` (primo), `RouterModule`, moduli CMS-structure/layout, `PageComponentModule.forRoot()`, `RoutingModule.forRoot()`, `MediaModule.forRoot()`, `OutletModule.forRoot()`, `CmsLcpModule.forRoot()`, `StorefrontComponentModule` (core-libs/storefront/base-storefront.module.ts:L26-L49).

- `BaseCoreModule` → `ErrorHandlingModule.forRoot()` … `SiteThemeModule.forRoot()`, `FederatedLoginModule` (core-libs/core/src/base-core.module.ts:L26-L46).
- Il `RoutingModule` dello storefront importa `CoreRoutingModule.forRoot()` (il `RoutingModule` di core) e `CmsRouteModule` (core-libs/storefront/cms-structure/routing/routing.module.ts:L18-L20): `BaseCoreModule` **non** importa direttamente il routing di core.
- `SpartacusFeaturesModule` apre con `AuthModule.forRoot()`, `LogoutModule`, `LoginRouteModule` e prosegue con le feature (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L176-L214).

3.3 Fase 2 — Creazione dell'environment injector

`internalCreateApplication`:

- in server mode lancia `PLATFORM_NOT_FOUND` se manca il `BootstrapContext` (angular:packages/core/src/application/create_application.ts:L47-L54) — è il motivo per cui `main.server.ts` esporta `(context) => bootstrapApplication(AppComponent, config, context)` (projects/storefrontapp/src/main.server.ts:L23-L26);
- crea l'environment injector con `runEnvironmentInitializers: false`, rimandando gli `ENVIRONMENT_INITIALIZER` a quando la `NgZone` è disponibile (angular:packages/core/src/application/create_application.ts:L66-L85).

3.4 Fase 3 — `ENVIRONMENT_INITIALIZER`: istanziazione eager dei moduli e del `Config`

Dentro `ngZone.run`, `bootstrap()` chiama per prima cosa `r3Injector.resolveInjectorInitializers()` (angular:packages/core/src/platform/bootstrap.ts:L87-L93), che invoca sincronamente ogni funzione `ENVIRONMENT_INITIALIZER` (angular:packages/core/src/di/r3_injector.ts:L426-L448). Poiché ogni NgModule ha il suo `ENVIRONMENT_INITIALIZER` "eager", **tutte le classi NgModule vengono istanziate qui**, prima di qualunque APP_INITIALIZER.

Per Spartacus la conseguenza più importante è in `ConfigModule`: il suo costruttore inietta `ConfigurationService` proprio per forzarne l'istanziazione (core-libs/core/src/config/config.module.ts:L13-L19). `ConfigurationService` a sua volta inietta `RootConfig`, `DefaultConfig` e `Config`, e si sottoscrive subito a `feedUnifiedConfig()` (core-libs/core/src/config/services/configuration.service.ts:L42-L55). Iniettare `Config` significa eseguire la sua factory `deepMerge({}, inject(DefaultConfig), inject(RootConfig))` (core-libs/core/src/config/config-tokens.ts:L10-L21), dove `DefaultConfig` e `RootConfig` sono a loro volta il merge di tutti i `DefaultConfigChunk` e `ConfigChunk` (core-libs/core/src/config/config-tokens.ts:L23-L47). **Al termine di questa fase esiste già un oggetto `Config` globale con tutta la configurazione statica**; i `CONFIG_INITIALIZER` lo muteranno in place più avanti. I dettagli del merge sono nel cap. 11.

3.5 Fase 4 — `APP_INITIALIZER`: il cuore dell'ordine di inizializzazione

Subito dopo, `bootstrap()` registra un pending task e chiama `initStatus.runInitializers()` (angular:packages/core/src/platform/bootstrap.ts:L137-L141). Il comportamento di `runInitializers()` è la chiave di tutto il capitolo (angular:packages/core/src/application/application_init.ts:L239-L275):

- tutti gli APP_INITIALIZER vengono invocati **sincronamente, in sequenza**, in injection context;
- se una funzione restituisce una `Promise` o un `Observable`, questa viene raccolta; i valori di ritorno di altro tipo (incluso `void`) sono ignorati;
- `donePromise` si risolve con `Promise.all` di quanto raccolto; un rifiuto qualunque rigetta il bootstrap.

Quindi **l'ordine nel multi-token decide solo l'ordine in cui le funzioni partono**; ciò che conta per "chi aspetta chi" sono le Promise e le dipendenze interne (scope di config, `LOCATION_INITIALIZED`). L'ordine esatto dei

multi-provider nell'injector dell'app demo non è stato osservato a runtime [DA VERIFICARE] (motivo: `node_modules` assente, ricostruibile solo leggendo il walker di `angular:packages/core/src/di/provider_collection.ts:L293-L355`). La tabella seguente elenca gli APP_INITIALIZER della catena `Base*Module` e se bloccano il bootstrap.

APP_INITIALIZER	REGISTRATO DA	RITORNA	BLOCCA IL BOOTSTRAP?
`configInitializerFactory` → `ConfigInitializerService.initialize(CONFIG_INITIALIZER[])`	`core-libs/core/src/config/config-initializer/config-initializer.module.ts:L24-L30`, `core-libs/core/src/config/config-initializer/config-initializer.module.ts:L51-L59`	`Promise<void>`	Sì, fino alla risoluzione di tutti i `configFactory()`
`configValidatorFactory`	`core-libs/core/src/config/config-validator/config-validator.module.ts:L23-L37`	`void`	No (in dev mode si sottoscrive a `getStable()` e valida in seguito)
`i18nextProviders`	`core-libs/core/src/i18n/i18next/i18next-providers.ts:L13-L28`	`Promise` (`getStable('i18n')` poi `initialize()`)	Sì
`initializeContext`	`core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L33`	`Promise<Config>` (`getStable('context')`)	Sì, fino a scope `context` stabile
`initializeLanguage`, `initializeCurrency`, `initializeBaseSite`	`core-libs/core/src/site-context/providers/context-initializer-providers.ts:L12-L55`	`void` (la Promise di `initialize()` è scartata)	No
`moduleInitializersFactory` (esegue i `MODULE_INITIALIZER` eager)	`core-libs/core/src/lazy-loading/lazy-loading.module.ts:L18-L53`	`Promise.all(...)`	Sì
`initializeTheme`	`core-libs/core/src/site-theme/providers/site-theme-initializer-providers.ts:L10-L23`	`void`	No
`initConfigurableRoutes`	`core-libs/core/src/routing/routing.module.ts:L37-L42`, `core-libs/core/src/routing/routing.module.ts:L93-L95`	`void` (sincrono)	No

APP_INITIALIZER	REGISTRATO DA	RITORNA	BLOCCA IL BOOTSTRAP?
addCmsRoute	core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L18-L25, core-libs/storefront/cms-structure/routing/cms-route/cms-route.module.ts:L10-L20	void (sincrono)	No
registerOutletsFactory	core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L69	void	No
onNavigateFactory	core-libs/storefront/router/app-routing.module.ts:L20-L37	void	No
mediaPreconnectInitializer	core-libs/storefront/shared/components/media/media.module.ts:L14-L18, core-libs/storefront/shared/components/media/media.module.ts:L31-L41	void (addPreconnectLink(): void, core-libs/storefront/cms-structure/seo/media-preconnect.service.ts:L20-L20)	No
htmlLangProvider (setHtmlLangAttribute)	core-libs/storefront/cms-structure/seo/html-lang-provider.ts:L10-L30	void (sottoscrive languageService.getActive(), core-libs/storefront/cms-structure/seo/html-lang-provider.ts:L20-L30)	No
initHtmlDirAttribute	core-libs/storefront/layout/direction/direction.module.ts:L12-L35 (catena base: core-libs/storefront/layout/layout.module.ts:L19-L19)	Promise (DirectionService.initialize() → lastValueFrom(getStable('direction')), core-libs/storefront/layout/direction/direction.module.ts:L12-L19, core-libs/storefront/layout/direction/direction.service.ts:L46-L48)	Sì, fino a scope direction stabile
Router enabledBlocking	angular:packages/router/src/provide_router.ts:L441-L474	Promise (attende LOCATION_INITIALIZED, poi la navigazione iniziale)	Sì

Altri file che registrano APP_INITIALIZER in core/storefront, non approfonditi qui: core-libs/core/src/routing/external-routes/external-routes.module.ts:L21-L21, core-libs/core/src/anonymous-consents/store/anonymous-consents-store.module.ts:L36-L36, core-libs/storefront/cms-structure/pwa/pwa.module.ts:L48-L48, core-libs/storefront/cms-structure/seo/seo.module.ts:L27-L27, core-libs/storefront/cms-structure/seo/structured-data/structured-data.module.ts:L29-L29, core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L38-L38.

Vediamo ora i blocchi principali in dettaglio.

3.5.1 `CONFIG_INITIALIZER`: config asincrona per scope

1. `ConfigInitializerModule.forRoot()` fornisce tre cose: il guard `CONFIG_INITIALIZER_FORROOT_GUARD` (`useValue: true`), l'APP_INITIALIZER `configInitializerFactory` con deps `ConfigInitializerService` e `[new Optional(), CONFIG_INITIALIZER]`, e un `LOCATION_INITIALIZED_MULTI` che attende `getStable()` (core-libs/core/src/config/config-initializer/config-initializer.module.ts:L41-L68).
2. `initialize()` esce subito se `ongoingScopes$.value` è già valorizzato (guard contro la doppia inizializzazione, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L130-L134).
3. Per ogni initializer: salta i `null`, lancia `Error('CONFIG_INITIALIZER should provide scope!')` se mancano gli scope (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L140-L146), avvisa (solo dev mode) se gli scope si sovrappongono a quelli già in corso (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L148-L152).
4. Tutti i `configFactory()` partono **in parallelo**; quando ciascuno si risolve, il risultato è fuso con `deepMerge` sia in `rootConfig` sia nel `Config` globale (mutazione in place), poi i suoi scope sono chiusi con `finishScopes` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L154-L168).
5. Solo dopo aver avviato tutti i factory viene emesso `ongoingScopes$.next(ongoingScopes)`, poi si attende `Promise.all` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L170-L175).

Chi vuole aspettare usa `getStable(...scopes)`: restituisce `of(this.config)` se la config è già stabile, altrimenti filtra `ongoingScopes$` (scartando il valore iniziale `undefined`) finché nessuno degli scope richiesti è in corso (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L35-L70). Senza argomenti aspetta che **non ci sia alcuno scope** in corso (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L91-L103). Due scope si sovrappongono se uno è prefisso dell'altro a confine di `.` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L105-L122). `isStable` è vero se manca il guard (modulo non importato) o se l'elenco degli scope in corso è vuoto (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L42-L44).

I `CONFIG_INITIALIZER` registrati dalla catena `Base*Module` + auth sono:

INITIALIZER	SCOPE	ATTIVO QUANDO	RIFERIMENTI
`SiteContextConfigInitializer`	`['context']`	`context.baseSite` **non** è configurato staticamente	core-libs/core/src/site-context/site-context.module.ts:L30-L41, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L24-L27
`SecurePortalConfigInitializer`	`['routing']`	`routing.protected === undefined` nella config statica	core-libs/core/src/routing/routing.module.ts:L44-L52, core-libs/core/src/routing/routing.module.ts:L96-L101, core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L16-L19
`AuthConfigInitializer`	`['authentication.OAuthLibConfig.redirectUri', 'authentication.client_id']`	feature toggle `asyncAuthConfigInitializer` attivo (default `false`; l'app demo lo attiva, vedi §3.9)	core-libs/core/src/auth/user-auth/user-auth.module.ts:L108-L113, core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L18-L24, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1090-L1090

3.5.2 Site context

`SiteContextModule.forRoot()` fornisce in quest'ordine: default config factory, `contextServiceMapProvider`, `contextServiceProviders`, `siteContextParamsProviders`, il validator `baseSiteConfigValidator`, il `CONFIG_INITIALIZER` `initSiteContextConfig` e `contextInitializerProviders` (core-libs/core/src/site-context/site-context.module.ts:L46-L66).

- La default config (`defaultSiteContextConfigFactory`) definisce solo `context.language` (prima `'en'`) e `context.currency` (prima `'USD'`); **non** definisce `context.baseSite` né `context.urlParameters` (core-libs/core/src/site-context/config/default-site-context-config.ts:L13-L58).
- `initSiteContextConfig` attiva il caricamento da backend solo se manca `context.baseSite` statico (core-libs/core/src/site-context/site-context.module.ts:L30-L41). In quel caso `resolveConfig()` chiede tutti i base site (`baseSiteService.getAll()`), considera l'origin del federated login se attivo, sceglie quello i cui `urlPatterns` (regex Java convertite) matchano l'URL corrente, e **lancia** se nessuno corrisponde (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L46-L75). `getConfig()` costruisce `context.urlParameters` (mappando `storefront`→`baseSite`), `baseSite: [uid]`, `language`/`currency` con il default in testa e `theme` (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L77-L98, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L113-L119).
- `initializeContext` (APP_INITIALIZER) restituisce `lastValueFrom(configInit.getStable('context'))`: è **questo** che blocca il bootstrap sul site context (core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L33).
- I tre APP_INITIALIZER language, currency, baseSite chiamano `initialize()` senza restituirne la Promise (core-libs/core/src/site-context/providers/context-initializer-providers.ts:L12-L55). `LanguageInitializer.initialize()` attende `getStable('context')`, poi in sequenza: lingua dal federated login, `siteContextRoutesHandler.initOnce()`, `languageStatePersistenceService.initSync()`, fallback dal config (core-libs/core/src/site-context/services/language-initializer.ts:L35-L44); il fallback imposta il primo valore di `context.language` solo se la lingua non è già inizializzata (core-libs/core/src/site-context/services/language-initializer.ts:L63-L71). `CurrencyInitializer` fa lo stesso senza il passo federated (core-libs/core/src/site-context/services/currency-initializer.ts:L33-L41); `BaseSiteInitializer` fa `initOnce()` + fallback su `context.baseSite` (core-libs/core/src/site-context/services/base-site-initializer.ts:L31-L65).
- `SiteContextRoutesHandler.initOnce()` esegue `initInternal()` una volta sola: se ci sono parametri di contesto nell'URL li legge **sincronamente** da `location.path(true)` e poi si sottoscrive ai cambi; infine completa un `ReplaySubject` (core-libs/core/src/site-context/services/site-context-routes-handler.ts:L78-L104).
- `siteContextParamsProviders` sostituisce `UrlSerializer` con `SiteContextUrlSerializer` (core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L18); per questo il commento in `BaseCoreModule` chiede che `SiteContextModule.forRoot()` arrivi dopo `RouterModule.forRoot` (core-libs/core/src/base-core.module.ts:L38-L38).

Il dettaglio dei servizi di contesto è nel cap. 13.

3.5.3 Routing: rotte configurabili e rotta CMS

- `initConfigurableRoutes` chiama `ConfigurableRoutesService.init()`, idempotente (flag `initCalled`), che ottiene il `Router` dall'injector — non dal costruttore, "due to cyclic dependency with APP_INITIALIZER" — e fa `router.resetConfig(this.configureRoutes(router.config))` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L31-L52). Ogni `Route` con `data.cxRoute` riceve `path`/`matcher` da `routing.routes[nome]`, ricorsivamente sui `children` (core-libs/core/src/routing/

configurable-routes/configurable-routes.service.ts:L59-L80); la config è letta **sincronamente** da RoutingConfig (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L21-L34).

- addCmsRoute fa `router.config.push({ path: '**', canActivate: [CmsPageGuard], component: PageLayoutComponent })` (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25).
- `RoutingModule.forRoot()` dello storefront aggiunge `provideDefaultConfigFactory(defaultRoutesConfigFactory)` e due `BEFORE_CMS_PAGE_GUARD` (`ProtectedRoutesGuard`, `FederatedLoginGuard`) (core-libs/storefront/cms-structure/routing/routing.module.ts:L22-L39).

Vedi il cap. 14 per route semantiche e `CmsPageGuard`.

3.5.4 `MODULE_INITIALIZER` eager

`LazyLoadingModule.forRoot()` registra un APP_INITIALIZER che prende tutti i `MODULE_INITIALIZER` visibili dal root injector, li esegue con `runModuleInitializerFunctions` e attende con `Promise.all`; in caso di rifiuto logga "MODULE_INITIALIZER promise was rejected during app initialization." e rilancia (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L18-L53). `runModuleInitializerFunctions` conserva solo i risultati con un metodo `then` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L187-L215). Il JSDoc del token conferma: nei moduli eager le funzioni girano all'inizializzazione dell'app (core-libs/core/src/lazy-loading/tokens.ts:L9-L20). Per i moduli lazy vedi §3.8.

3.6 Fase 5 — `LOCATION_INITIALIZED` e navigazione iniziale bloccante

`AppRoutingModule` usa `initialNavigation: 'enabledBlocking'` (core-libs/storefront/router/app-routing.module.ts:L13-L19), per cui `RouterModule.forRoot` aggiunge i provider di `withEnabledBlockingInitialNavigation()` (angular:packages/router/src/router_module.ts:L237-L244). Questi registrano un APP_INITIALIZER che (angular:packages/router/src/provide_router.ts:L441-L474):

1. legge `LOCATION_INITIALIZED` (default `Promise.resolve()`);
2. **dentro il** `.then` di quella Promise chiama `router.initialNavigation()` — quindi sempre in modo asincrono rispetto al ciclo sincrono di `runInitializers()` (angular:packages/router/src/provide_router.ts:L452-L469);
3. si risolve quando la navigazione arriva ad `afterPreactivation` (guard già eseguiti) oppure alla fine se la navigazione è cancellata/in errore; ad `afterPreactivation` restituisce `bootstrapDone`, che tiene ferma l'attivazione finché il root component non è stato creato.

Spartacus sovrascrive `LOCATION_INITIALIZED` nel `RoutingModule.forRoot()` di core con una factory privata che invoca tutti i `LOCATION_INITIALIZED_MULTI` e ritorna `Promise.all` (core-libs/core/src/routing/routing.module.ts:L54-L71, core-libs/core/src/routing/routing.module.ts:L102-L105). I contributori nella catena base sono:

- `ConfigInitializerModule`: `() => lastValueFrom(configInitializer.getStable())` — "Hold on the initial navigation until the Spartacus configuration is stable" (core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64);
- `UserAuthModule`: `authNotificationInitializerFactory` e `authInitializedFactory`; quest'ultimo sincronizza lo stato di auth persistito e controlla i parametri OAuth nell'URL (core-libs/core/src/auth/user-auth/user-auth.module.ts:L70-L92, core-libs/core/src/auth/user-auth/user-auth.module.ts:L119-L120);
- fuori dalla catena base: integration-libs/opf/payment/root/opf-payment-root.module.ts:L42-L42 e integration-libs/opps/root/coupon-codes/opps-coupon-codes.module.ts:L22-L22.

La conseguenza: la navigazione iniziale parte solo quando **tutta** la config è stabile (tutti gli scope) e l'auth è sincronizzata. E poiché parte in un `.then`, a quel punto `initConfigurableRoutes` e `addCmsRoute` (sincroni) sono già stati eseguiti, qualunque sia il loro ordine nel multi-token.

3.7 Fase 6 — Bootstrap del root component e primo render

Quando `donePromise` si risolve, `bootstrap()` imposta il `LOCALE_ID` e chiama `appRef.bootstrap(config.rootComponent)` (angular:packages/core/src/platform/bootstrap.ts:L143-L171).

- `AppComponent` (`app-root`, ospitato in `index.html` come `<app-root>Loading...</app-root>`) ha come template solo `<cx-storefront />` (projects/storefrontapp/src/app/app.component.ts:L14-L19, projects/storefrontapp/src/index.html:L82-L82).
- `StorefrontComponent` è standalone (core-libs/storefront/layout/main/storefront.component.ts:L51-L67); il template è avvolto nell'outlet `StorefrontOutlets.STOREFRONT` e contiene header (`cx-page-layout section="header"` e `"navigation"`), `cx-global-message`, `<main>` con `<router-outlet />` e footer (core-libs/storefront/layout/main/storefront.component.html:L1-L28).
- In `ngOnInit` si sottoscrive a `routingService.isNavigating()` (core-libs/storefront/layout/main/storefront.component.ts:L126-L138).
- Il bootstrap listener del router chiama `router.resetRootComponentType(...)` e completa `BOOTSTRAP_DONE` (angular:packages/router/src/provide_router.ts:L319-L343): l'attivazione della navigazione iniziale — rotta `**` → `PageLayoutComponent` — può terminare e il `<router-outlet />` viene riempito.

3.8 Fase 7 — Dopo il bootstrap: moduli lazy

Quando una feature viene caricata lazy, `runModuleInitializersForModule` legge `MODULE_INITIALIZER` con `{ self: true }` dall'injector del modulo e attende le Promise prima di emettere il `moduleRef` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L175). `OutletModule.forChild()` è un esempio concreto: registra la stessa `registerOutletsFactory` come `MODULE_INITIALIZER` invece che come APP_INITIALIZER (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88). Vedi il cap. 12.

3.9 Il caso dell'app demo: quali scope sono attivi

Nell'app demo:

- `context.baseSite` è configurato staticamente (B2C: elettronica/apparel; B2B: `powertools-spa`, `powertools-standalone`) (projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L11-L31, projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L13-L28) ⇒ `initSiteContextConfig` ritorna `null`, **nessuno scope** `context` (core-libs/core/src/site-context/site-context.module.ts:L37-L40). Lingua, valuta e `urlParameters` vengono dalla config statica e dai default (core-libs/core/src/site-context/config/default-site-context-config.ts:L13-L58).
- `routing.protected` non è impostato (nessun risultato per `protected:` sotto `projects/storefrontapp/src`) ⇒ `SecurePortalConfigInitializer` è **attivo** con scope `routing` (core-libs/core/src/routing/routing.module.ts:L44-L52). Il suo `resolveConfig()` attende `baseSiteService.get()` — cioè un base site **attivo**, perché `get()` senza argomento parte da `getActive()` che filtra i falsy — e contribuisce solo `routing.protected` (core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L31-L50, core-libs/core/src/site-context/facade/base-site.service.ts:L31-L36, core-libs/core/src/site-context/facade/base-site.service.ts:L56-L70).
- Il demo attiva il toggle `asyncAuthConfigInitializer: true` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L297-L298, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L377-L377), quindi è attivo anche `AuthConfigInitializer` con gli scope

`authentication.OAuthLibConfig.redirectUri` e `authentication.client_id` (core-libs/core/src/auth/user-auth/user-auth.module.ts:L108-L113, core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L20-L23). Il suo `resolveConfig()` combina `baseSiteService.getActive()` con `getStable('authentication.initializerOptions')`, scope che non si sovrappone ai due in corso (core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L35-L41).

Gli scope attivi nell'app demo sono quindi tre (`routing` più i due di `authentication`), e tutti attendono un base site attivo. Ne deriva una catena di dipendenze interessante: questi scope si chiudono solo dopo che `BaseSiteInitializer` ha attivato un base site; quest'ultimo aspetta `getStable('context')`, che è subito soddisfatto perché `'context'` non si sovrappone né a `'routing'` né agli scope `authentication.*` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L91-L122). Il `LOCATION_INITIALIZED_MULTI` di config (`getStable()` senza scope) aspetta anche `routing` e gli scope di `authentication`, quindi la navigazione iniziale parte solo dopo l'attivazione del base site (e, per `routing`, la risposta OCC sui base site).

---

4. Diagrammi

4.1 Sequenza end-to-end del bootstrap browser

■ **TAVOLA 10.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti dei passi: 1 projects/storefrontapp/src/main.ts:L17-L17; 2 angular:packages/core/src/application/create_application.ts:L72-L79; 3 angular:packages/core/src/platform/bootstrap.ts:L89-L90, core-libs/core/src/config/config.module.ts:L13-L19; 4 angular:packages/core/src/application/application_init.ts:L239-L256; 5-6 core-libs/core/src/config/config-initializer/config-initializer.module.ts:L24-L30, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L154-L175; 7 core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L21; 8 core-libs/core/src/site-context/providers/context-initializer-providers.ts:L12-L34; 9 core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L48-L52; 10 core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L18-L25; 11-12 angular:packages/router/src/provide_router.ts:L445-L451, core-libs/core/src/routing/routing.module.ts:L66-L71, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64; 14-15 angular:packages/router/src/provide_router.ts:L456-L469; 16-17 angular:packages/core/src/platform/bootstrap.ts:L143-L171; 18 angular:packages/router/src/provide_router.ts:L338-L342; 19 core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L16. L'ordine relativo fra i passi 5-11 dipende dall'ordine dei multi-provider DA VERIFICARE (vedi §3.5).

4.2 Albero dei moduli importati al bootstrap (app demo)

■ **TAVOLA 10.2** diagramma a tutta pagina, in fondo al capitolo

Fonti: projects/storefrontapp/src/app/app.config.ts:L26-L35, projects/storefrontapp/src/app/app.module.ts:L15-L25, core-libs/storefront/router/app-routing.module.ts:L13-L19, projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L20, core-libs/storefront/base-storefront.module.ts:L26-L49, core-libs/storefront/cms-structure/routing/routing.module.ts:L18-L20, core-libs/core/src/base-core.module.ts:L26-L46. Per leggibilità sono mostrati solo i moduli rilevanti per l'inizializzazione.

4.3 Chi aspetta chi: scope di config e Promise bloccanti

■ **TAVOLA 10.3** diagramma a tutta pagina, in fondo al capitolo

Fonti: core-libs/core/src/site-context/site-context.module.ts:L30-L41, core-libs/core/src/routing/routing.module.ts:L44-L52, core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L21, core-libs/core/src/site-context/services/base-site-initializer.ts:L31-L65, core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L31-L41, core-libs/core/src/site-context/facade/base-site.service.ts:L56-L70, core-libs/core/src/auth/user-auth/user-auth.module.ts:L108-L113, core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L35-L41, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64, core-libs/core/src/auth/user-auth/user-auth.module.ts:L119-L120, core-libs/core/src/routing/routing.module.ts:L66-L71, angular:packages/router/src/provide_router.ts:L452-L469, angular:packages/core/src/application/application_init.ts:L263-L273.

---

5. Estratti di codice reale

5.1 Entry point e `ApplicationConfig` dell'app demo

projects/storefrontapp/src/main.ts:L13-L23

```ts
if (environment.production) {
  enableProdMode();
}

bootstrapApplication(AppComponent, appConfig)
  /* eslint-disable-next-line no-console
  --
  It's just an example application file. This message is not crucial
  to be logged using any special logger. Moreover, we don't have
  any special logger available in this context. */
  .catch((err) => console.error(err));
```

projects/storefrontapp/src/app/app.config.ts:L26-L35

```ts
export const appConfig: ApplicationConfig = {
  providers: [
    provideHttpClient(withFetch(), withInterceptorsFromDi()),
    provideClientHydration(withEventReplay(), withNoHttpTransferCache()),
    provideZoneChangeDetection({ eventCoalescing: true }),
    provideBrowserGlobalErrorListeners(),

    importProvidersFrom(AppModule),
  ],
};
```

projects/storefrontapp/src/main.server.ts:L23-L26

```ts
const bootstrap = (context: BootstrapContext) =>
  bootstrapApplication(AppComponent, config, context);

export default bootstrap;
```

5.2 `BaseCoreModule`: l'ordine dichiarato

core-libs/core/src/base-core.module.ts:L26-L53

```ts
@NgModule({
  imports: [
    ErrorHandlingModule.forRoot(), // Import this module before any other interceptor to handle HTTP errors efficiently
    StateModule.forRoot(),
    ConfigModule.forRoot(),
    ConfigInitializerModule.forRoot(),
    ConfigValidatorModule.forRoot(),
    I18nModule.forRoot(),
    CmsModule.forRoot(),
    GlobalMessageModule.forRoot(),
    ProcessModule.forRoot(),
    FeaturesConfigModule.forRoot(),
    SiteContextModule.forRoot(), // should be imported after RouterModule.forRoot, because it overwrites UrlSerializer
    MetaTagConfigModule.forRoot(),
    BaseOccModule.forRoot(),
    LazyLoadingModule.forRoot(),
    HttpModule.forRoot(),
    SiteThemeModule.forRoot(),
    FederatedLoginModule,
  ],
})
export class BaseCoreModule {
  static forRoot(): ModuleWithProviders<BaseCoreModule> {
    return {
      ngModule: BaseCoreModule,
    };
  }
}
```

5.3 `BaseStorefrontModule`

core-libs/storefront/base-storefront.module.ts:L26-L49

```ts
@NgModule({
  imports: [
    BaseCoreModule.forRoot(),
    RouterModule,
    GlobalMessageComponentModule,
    OutletModule,
    OutletRefModule,
    PwaModule,
    PageLayoutModule,
    SeoModule,
    PageComponentModule.forRoot(),
    PageSlotModule,
    SkipLinkModule,
    KeyboardFocusModule,
    LayoutModule,
    RoutingModule.forRoot(),
    MediaModule.forRoot(),
    OutletModule.forRoot(),
    CmsLcpModule.forRoot(),
    StorefrontComponentModule,
  ],
  exports: [LayoutModule, StorefrontComponentModule],
})
export class BaseStorefrontModule {}
```

5.4 Angular: `runInitializers()`

angular:packages/core/src/application/application_init.ts:L239-L275

```ts
  runInitializers() {
    if (this.initialized) {
      return;
    }

    const asyncInitPromises = [];
    for (const appInits of this.appInits) {
      const initResult = runInInjectionContext(this.injector, appInits);
      if (isPromise(initResult)) {
        asyncInitPromises.push(initResult);
      } else if (isSubscribable(initResult)) {
        const observableAsPromise = new Promise<void>((resolve, reject) => {
          initResult.subscribe({complete: resolve, error: reject});
        });
        asyncInitPromises.push(observableAsPromise);
      }
    }
    // ...
    Promise.all(asyncInitPromises)
```

```
      .then(() => {
        complete();
      })
      .catch((e) => {
        this.reject(e);
      });
    // ...
  }
```

5.5 ConfigInitializerModule.forRoot()

core-libs/core/src/config/config-initializer/config-initializer.module.ts:L41-L68

```ts
@NgModule({})
export class ConfigInitializerModule {
  static forRoot(): ModuleWithProviders<ConfigInitializerModule> {
    return {
      ngModule: ConfigInitializerModule,
      providers: [
        {
          provide: CONFIG_INITIALIZER_FORROOT_GUARD,
          useValue: true,
        },
        {
          provide: APP_INITIALIZER,
          multi: true,
          useFactory: configInitializerFactory,
          deps: [
            ConfigInitializerService,
            [new Optional(), CONFIG_INITIALIZER],
          ],
        },
        provideLocationInitializerFactory(() => {
          // Hold on the initial navigation until the Spartacus configuration is stable
          const configInitializer = inject(ConfigInitializerService);
          return () => lastValueFrom(configInitializer.getStable());
        }),
      ],
    };
  }
}
```

5.6 ConfigInitializerService.initialize()

core-libs/core/src/config/config-initializer/config-initializer.service.ts:L130-L175

```ts
  async initialize(initializers?: ConfigInitializer[]) {
    if (this.ongoingScopes$.value) {
      // guard for double initialization
      return;
    }

    const ongoingScopes: string[] = [];

    const asyncConfigs: Promise<void>[] = [];

    for (const initializer of initializers || []) {
      if (!initializer) {
        continue;
      }
      if (!initializer.scopes || !initializer.scopes.length) {
        throw new Error('CONFIG_INITIALIZER should provide scope!');
      }
      // ...
      ongoingScopes.push(...initializer.scopes);

      asyncConfigs.push(
        (async () => {
          const initializerConfig = await initializer.configFactory();
          // contribute configuration to rootConfig
          deepMerge(
            this.rootConfig as Record<string, unknown>,
            initializerConfig
          );
          // contribute configuration to global config
          deepMerge(this.config as Record<string, unknown>, initializerConfig);
          this.finishScopes(initializer.scopes);
        })()
      );
    }
    this.ongoingScopes$.next(ongoingScopes);

    if (asyncConfigs.length) {
      await Promise.all(asyncConfigs);
    }
  }
```

5.7 `RoutingModule.forRoot()` di core: override di `LOCATION_INITIALIZED`

core-libs/core/src/routing/routing.module.ts:L66-L71

```ts
function locationInitializedFactory(): Promise<any> {
  const initializers =
    inject(LOCATION_INITIALIZED_MULTI, { optional: true }) ?? [];
```

```
  const promiseInitializers = initializers.map((initializer) => initializer());
  return Promise.all(promiseInitializers);
}
```

core-libs/core/src/routing/routing.module.ts:L87-L106

```ts
    providers: [
      reducerProvider,
      {
        provide: RouterStateSerializer,
        useClass: CustomSerializer,
      },
      provideAppInitializer(() =>
        initConfigurableRoutes(inject(ConfigurableRoutesService))()
      ),
      provideConfigInitializerFactory(() => {
        return initSecurePortalConfig(
          inject(SecurePortalConfigInitializer),
          inject(RoutingConfig)
        );
      }),
      {
        provide: LOCATION_INITIALIZED,
        useFactory: locationInitializedFactory,
      },
    ],
```

5.8 Angular: navigazione iniziale bloccante

angular:packages/router/src/provide_router.ts:L445-L472

```ts
  provideAppInitializer(() => {
    const injector = inject(Injector);
    const locationInitialized: Promise<any> = injector.get(
      LOCATION_INITIALIZED,
      Promise.resolve(),
    );

    return locationInitialized.then(() => {
      return new Promise((resolve) => {
        const router = injector.get(Router);
        const bootstrapDone = injector.get(BOOTSTRAP_DONE);
        afterNextNavigation(router, () => {
          // Unblock APP_INITIALIZER in case the initial navigation was canceled or errored
          // without a redirect.
          resolve(true);
        });
```

```
          injector.get(NavigationTransitions).afterPreactivation = () => {
            // Unblock APP_INITIALIZER once we get to `afterPreactivation`. At this point, we
            // assume activation will complete successfully (even though this is not
            // guaranteed).
            resolve(true);
            return bootstrapDone.closed ? of(void 0) : bootstrapDone;
          };
          router.initialNavigation();
        });
      });
    }),
```

5.9 Site context: initializer bloccante vs non bloccanti

core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L21

TS

```ts
export function initializeContext(
  configInit: ConfigInitializerService
): () => Promise<Config> {
  return () => {
    return lastValueFrom(configInit.getStable('context'));
  };
}
```

core-libs/core/src/site-context/providers/context-initializer-providers.ts:L20-L26

TS

```ts
export function initializeLanguage(
  languageInitializer: LanguageInitializer
): () => void {
  return () => {
    languageInitializer.initialize();
  };
}
```

core-libs/core/src/site-context/site-context.module.ts:L30-L41

TS

```ts
export function initSiteContextConfig(
  configInitializer: SiteContextConfigInitializer,
  config: SiteContextConfig
): ConfigInitializer | null {
  /**
   * Load config for `context` from backend only when there is no static config for `context.baseSite`
   */
  if (!config.context?.[BASE_SITE_CONTEXT_ID]) {
    return configInitializer;
```

```
  }
  return null;
}
```

5.10 La rotta wildcard CMS

core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25

```ts
const cmsRoute: Route = {
  path: '**',
  canActivate: [CmsPageGuard],
  component: PageLayoutComponent,
};

export function addCmsRoute(injector: Injector): () => void {
  const result = () => {
    // Router could not be injected via `deps[]` due to a cyclic dependency with APP_INITIALIZER:
    const router = injector.get(Router);
    router.config.push(cmsRoute);
  };
  return result;
}
```

5.11 Recipe: `layoutConfigFactory`

core-libs/storefront/recipes/config/layout-config.ts:L187-L198

```ts
/**
 * Factory for layout configuration.
 */
export function layoutConfigFactory(): LayoutConfig {
  const config: LayoutConfig = JSON.parse(JSON.stringify(layoutConfig));

  applyUnifiedHeaderSlots(config);

  applyWithoutPageFold(config);

  return config;
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`context.baseSite`	`string[]`	nessun default nel core (`defaultSiteContextConfigFactory` non lo definisce); app demo: lista B2C o B2B	`core-libs/core/src/site-context/config/default-site-context-config.ts:L13-L58`; `projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L26-L31`, `projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L22-L27`	`core-libs/core/src/site-context/site-context.module.ts:L37-L37` (decide se caricare dal backend); `core-libs/core/src/site-context/services/base-site-initializer.ts:L57-L64`; `core-libs/core/src/site-context/config/base-site-config-validator.ts:L11-L15`
`context.language`	`string[]` (primo = default)	`['en', 'de', 'ja', ...]`	`core-libs/core/src/site-context/config/default-site-context-config.ts:L16-L34`; backend: `core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L82-L85`	`core-libs/core/src/site-context/services/language-initializer.ts:L63-L71`
`context.currency`	`string[]` (primo = default)	`['USD', 'EUR', ...]`	`core-libs/core/src/site-context/config/default-site-context-config.ts:L35-L56`; backend: `core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L86-L89`	`core-libs/core/src/site-context/services/currency-initializer.ts:L33-L41`
`context.urlParameters`	`string[]`	nessuno nel core	app demo `['baseSite','language','currency']` (`projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L28-L28`); backend `core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L80-L80`	`core-libs/core/src/site-context/services/site-context-routes-handler.ts:L92-L100`
`routing.routes[name]`	`RouteConfig`	`defaultRoutesConfigFactory`	`core-libs/storefront/cms-structure/routing/routing.module.ts:L26-L26`; app demo `projects/storefrontapp/src/app/private/private.providers.ts:L34-L41`	`core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L26-L34`
`routing.protected`	`boolean | undefined`	`undefined` ⇒ `SecurePortalConfigInitializer` attivo	backend: `core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L43-L50`	`core-libs/core/src/routing/routing.module.ts:L48-L48`
`enableResetViewOnNavigate.active`	`boolean`	`true` (insieme a `ignoreQueryString: false`, `ignoreRoutes: []`, `selectedHostElement: 'body'`)	`core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16` (fornita in `core-libs/storefront/router/app-routing.module.ts:L21-L21`)	`core-libs/storefront/router/on-navigate.service.ts:L69-L73`
`i18n.resources` / `i18n.chunks` / `i18n.fallbackLang`	oggetti	app demo statica (`fallbackLang: 'en'`)	`projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L55-L67`	scope `'i18n'` atteso in `core-libs/core/src/i18n/i18next/i18next-providers.ts:L19-L22`
`features.level`	`string`	app demo `'*'`	`projects/storefrontapp/src/app/private/private.providers.ts:L44-L44`	vedi cap. 42
feature toggle `asyncAuthConfigInitializer`	`boolean`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1090-L1090`	`core-libs/core/src/auth/user-auth/user-auth.module.ts:L108-L113`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
layoutSlots	Layout Config	layoutConfig Factory()	core-libs/storefront/recipes/config/layout-config.ts:L187-L198; app demo projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L46-L46	vedi cap. 18
mediaFormats, media.picture ElementFormats, media.picture FormatsOrder	MediaConfig	mediaConfig	core-libs/storefront/recipes/config/default-media.config.ts:L9-L38	vedi cap. 45
backend.occ. endpoints.* (B2B)	Occ Endpoints	defaultB2bOccConfig	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L38-L49	vedi cap. 21
environment.b2b	boolean (build time, solo app demo)	CX_B2B ?? false	projects/storefrontapp/src/environments/environment.ts:L27-L27	projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L40-L42
environment.epd Visualization	boolean (build time, solo app demo)	CX_EPD_VISUALIZATION ?? false	projects/storefrontapp/src/environments/environment.ts:L32-L32	projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L21-L23, projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L15-L17
RouterModule.for Root initial Navigation	Extra Options	'enabledBlocking' (in AppRoutingModule)	core-libs/storefront/router/app-routing.module.ts:L15-L18	angular:packages/router/src/router_module.ts:L237-L244

6.1 Le recipes in breve

- mediaConfig — formati mobile 400, tablet 770, desktop 1140, widescreen 1400, cartIcon 65, thumbnail 96, product 284, zoom 515, più media.pictureElementFormats con media query e pictureFormatsOrder (core-libs/storefront/recipes/config/default-media.config.ts:L9-L38).
- layoutConfig (deprecato) — slot di header (con variante lg), navigation, footer e di 16 page template, con pageFold su LandingPage2Template, CategoryPageTemplate, ProductDetailsPageTemplate (core-libs/storefront/recipes/config/layout-config.ts:L32-L133, page template in core-libs/storefront/recipes/config/layout-config.ts:L55-L133). Il JSDoc indica di usare layoutConfigFactory con provideConfigFactory(layoutConfigFactory) (core-libs/storefront/recipes/config/layout-config.ts:L13-L27).
- layoutConfigFactory() — deep copy JSON di layoutConfig, poi applyUnifiedHeaderSlots (elimina header.lg e mette in header.slots gli 8 slot completi, core-libs/storefront/recipes/config/layout-config.ts:L136-L156) e applyWithoutPageFold (rimuove pageFold per evitare regressioni CLS, core-libs/storefront/recipes/config/layout-config.ts:L158-L185).
- defaultCmsContentProviders — provideCmsStructure per HamburgerMenuComponent nello slot PreHeader e LoginComponent in SiteLogin (core-libs/storefront/recipes/config/static-cms-structure.ts:L10-L19).
- defaultB2bOccConfig — endpoint orgUsers/${userId}/... per user, addEntries, placeOrder, scheduleReplenishmentOrder, reorder, e users/${userId} per profilo/chiusura account (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L18-L49).

L'app demo usa le recipe così: provideConfigFactory(layoutConfigFactory), provideConfig(mediaConfig), ...defaultCmsContentProviders (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L44-L72) e, in B2B, provideConfig(defaultB2bOccConfig) (projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L13-L28).

7. Commenti e note notevoli dal sorgente

- "Import this module before any other interceptor to handle HTTP errors efficiently" — su `ErrorHandlingModule.forRoot()` (core-libs/core/src/base-core.module.ts:L28-L28).
- "should be imported after RouterModule.forRoot, because it overwrites UrlSerializer" — su `SiteContextModule.forRoot()` (core-libs/core/src/base-core.module.ts:L38-L38).
- "Hold on the initial navigation until the Spartacus configuration is stable" (core-libs/core/src/config/config-initializer/config-initializer.module.ts:L61-L61).
- `@deprecated since 2211.22 - should not be a public API` sulla `locationInitializedFactory` esportata da `config-initializer.module.ts` (core-libs/core/src/config/config-initializer/config-initializer.module.ts:L32-L39).
- `@internal` "Not a part of a public API, used by APP_INITIALIZER to initialize all provided CONFIG_INITIALIZERS" su `initialize()` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L129).
- "Should stay private in 1.x as forRoot() is used internally by ConfigInitializerModule — issue: #5279" su `ConfigValidatorModule` (core-libs/core/src/config/config-validator/config-validator.module.ts:L39-L44).
- CAUTION sulla perdita dell'injection context nei `LocationInitializer` asincroni (core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L13-L28) e divieto di usare il Router nelle implementazioni (core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L30-L38).
- "Router could not be injected in constructor due to cyclic dependency with APP_INITIALIZER" (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L49-L49) e la nota gemella in `add-cms-route.ts` (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L20-L20).
- "workaround for AOT compilation" su `initConfigurableRoutes` (core-libs/core/src/routing/routing.module.ts:L40-L40).
- "functions below should not be exposed in public API" su `siteContextParamsProviders` (core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L12).
- `@deprecated since 2211.32. Use initOnce() instead` su `SiteContextRoutesHandler.init()` (core-libs/core/src/site-context/services/site-context-routes-handler.ts:L51-L56).
- Iniezione di `ConfigurationService` nel costruttore di `ConfigModule` "To make sure ConfigurationService will be instantiated" (core-libs/core/src/config/config.module.ts:L13-L19); sottoscrizione eager in `ConfigurationService` (core-libs/core/src/config/services/configuration.service.ts:L51-L54).
- Motivazione CLS della rimozione di `pageFold` (core-libs/storefront/recipes/config/layout-config.ts:L158-L171).
- Import per la type augmentation di `OccEndpoints` nella recipe B2B (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L7-L17).
- Triple-slash reference a `types.d.ts` in `@spartacus/storefront` per le augmentation globali (core-libs/storefront/public_api.ts:L7-L14).
- "CAUTION: IT IS NOT MEANT FOR CUSTOMERS' APPLICATIONS!" in `privateProviders` (projects/storefrontapp/src/app/private/private.providers.ts:L17-L17) e "DO NOT USE IN CUSTOMERS APPS" in `app.config.server.ts` (projects/storefrontapp/src/app/app.config.server.ts:L24-L24).
- B2B/B2C scelto a build time solo nel repo; nei progetti cliente deciso da `ng add @spartacus/schematics` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L33-L39).

- Lato Angular: `APP_INITIALIZER` "is now deprecated" a favore di `provideAppInitializer()` (angular:packages/core/src/application/application_init.ts:L150-L203). Spartacus usa ancora prevalentemente `APP_INITIALIZER` con `useFactory`/`deps`, con l'eccezione di `initConfigurableRoutes` registrato via `provideAppInitializer` (core-libs/core/src/routing/routing.module.ts:L93-L95).

8. Extension point

EXTENSION POINT	USO TIPICO	RIFERIMENTO / EXPORT	RICETTA
`CONFIG_INITIALIZER`, `ConfigInitializer`, `provideConfigInitializer`, `provideConfigInitializerFactory`	caricare una parte di config dal backend prima dell'avvio, dichiarando uno scope	core-libs/core/src/config/config-initializer/config-initializer.ts:L18-L67; export via core-libs/core/src/config/config-initializer/index.ts:L7-L12 → core-libs/core/public_api.ts:L15-L15	Ricette config e DI
`ConfigInitializerService.getStable(...scopes)`	in un APP_INITIALIZER o in un servizio che gira presto, aspettare solo la config che serve	core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70; export core-libs/core/src/config/config-initializer/index.ts:L14-L14	Ricette config e DI
`LOCATION_INITIALIZED_MULTI`, `provideLocationInitializer`, `provideLocationInitializerFactory`	trattenere la navigazione iniziale (es. leggere parametri dall'URL prima che il router li consumi)	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L22-L62; export core-libs/core/src/routing/index.ts:L11-L11 → core-libs/core/public_api.ts:L30-L30	Ricette config e DI
`MODULE_INITIALIZER`	inizializzazione di una feature lazy al momento del caricamento	core-libs/core/src/lazy-loading/tokens.ts:L19-L20; export core-libs/core/src/lazy-loading/index.ts:L9-L9 → core-libs/core/public_api.ts:L24-L24	Ricette feature lib lazy
`SiteContextConfigInitializer` (metodi `protected resolveConfig`, `getConfig`)	cambiare come si sceglie il base site dall'URL	core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L24-L98; export core-libs/core/src/site-context/config/index.ts:L7-L7	Ricette config e DI
`LanguageInitializer`, `CurrencyInitializer`, `BaseSiteInitializer` (metodi `protected`)	cambiare la priorità URL/persistenza/default	core-libs/core/src/site-context/services/language-initializer.ts:L35-L79, core-libs/core/src/site-context/services/base-site-initializer.ts:L31-L65; export core-libs/core/src/site-context/services/index.ts:L7-L10	Ricette config e DI
`ContextServiceMap` e costanti `*_CONTEXT_ID`	registrare un contesto custom	core-libs/core/src/site-context/providers/context-service-map.ts:L18-L33, core-libs/core/src/site-context/providers/context-ids.ts:L7-L10; export core-libs/core/src/site-context/providers/index.ts:L7-L8	vedi cap. 13

EXTENSION POINT	USO TIPICO	RIFERIMENTO / EXPORT	RICETTA
`BaseCoreModule`, `BaseStorefrontModule`	composizione dell'app; base per app senza alcune parti	core-libs/core/public_api.ts:L12-L12, core-libs/storefront/public_api.ts:L20-L20	API pubblica e upgrade
`AppRoutingModule`, `OnNavigateService`	router root con navigazione bloccante	core-libs/storefront/router/app-routing.module.ts:L13-L37; export core-libs/storefront/router/index.ts:L7-L9	Ricette route e layout
Recipes `mediaConfig`, `layoutConfigFactory`, `defaultCmsContentProviders`	punto di partenza da estendere con `provideConfig`	core-libs/storefront/recipes/index.ts:L7-L7 → core-libs/storefront/public_api.ts:L27-L27	Ricette route e layout
Recipe `defaultB2bOccConfig`	endpoint B2B	core-libs/setup/public_api.ts:L7-L7	Ricette OCC
Outlet di root `StorefrontOutlets.STOREFRONT`, `cx-header`, `cx-footer`	sostituire/arricchire la shell della pagina	core-libs/storefront/layout/main/storefront.component.html:L1-L28	Ricette outlet

Non esportati (quindi non da usare in un'app): `CONFIG_INITIALIZER_FORROOT_GUARD` (assente da core-libs/core/src/config/config-initializer/index.ts:L7-L12), `contextInitializerProviders`, `contextServiceProviders`, `siteContextParamsProviders` (core-libs/core/src/site-context/providers/index.ts:L7-L8 esporta solo `context-ids` e `context-service-map`). Vedi cap. 54.

8.1 Schema d'uso di un `CONFIG_INITIALIZER` (illustrativo)

Esempio **non presente nel repo**, costruito solo con simboli esportati e con la firma di `ConfigInitializer` (core-libs/core/src/config/config-initializer/config-initializer.ts:L18-L28) e `provideConfigInitializer` (core-libs/core/src/config/config-initializer/config-initializer.ts:L46-L52):

```ts
// ESEMPIO ILLUSTRATIVO (non presente nel repo)
@Injectable({ providedIn: 'root' })
export class MyFeatureConfigInitializer implements ConfigInitializer {
  readonly scopes = ['myFeature'];          // scope obbligatorio, altrimenti Error
  readonly configFactory = () =>
    Promise.resolve({ myFeature: { enabled: true } } as Config);
}

// in un modulo/array di provider dell'app
provideConfigInitializer(MyFeatureConfigInitializer);

// chi deve aspettare solo questa parte di config:
// configInitializerService.getStable('myFeature')
```

9. Trappole, casi limite, comportamenti controintuitivi

1. **Gli initializer di language/currency/baseSite non bloccano il bootstrap.** Ritornano `void`, quindi Angular non ne attende la Promise (core-libs/core/src/site-context/providers/context-initializer-providers.ts:L15-L17, angular:packages/core/src/application/application_init.ts:L246-L256). Il boo-

tstrap è bloccato da `initializeContext` (config `context` stabile), non dall'impostazione dei valori attivi. Codice in un APP_INITIALIZER che legge `languageService.getActive()` in modo sincrono può non trovare ancora un valore.

2. **Molti APP_INITIALIZER sono "fire and forget".** `initializeTheme`, `onNavigateFactory`, `addCmsRoute`, `initConfigurableRoutes` e il validator ritornano `void` (core-libs/core/src/site-theme/providers/site-theme-initializer-providers.ts:L12-L14, core-libs/storefront/router/app-routing.module.ts:L32-L37, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L18-L25, core-libs/core/src/routing/routing.module.ts:L37-L42, core-libs/core/src/config/config-validator/config-validator.module.ts:L23-L37).
3. **Le rotte sono configurate con la config statica.** `initConfigurableRoutes` legge `routing.routes` sincronamente durante `runInitializers` (core-libs/core/src/routing/routing.module.ts:L93-L95, core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L26-L27). Un `CONFIG_INITIALIZER` che contribuisse `routing.routes` in modo asincrono non verrebbe applicato alle `Route`. Il `SecurePortalConfigInitializer` del core contribuisce solo `routing.protected`, quindi non è toccato (core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L43-L50).
4. **Un `configFactory` che rigetta blocca tutto.** `finishScopes` non viene chiamato, `initialize()` rigetta, `donePromise` rigetta e il bootstrap fallisce; ogni `getStable` sullo scope resta in attesa (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L156-L174, angular:packages/core/src/application/application_init.ts:L263-L269).
5. **URL che non matcha alcun base site ⇒ bootstrap fallito.** Senza `context.baseSite` statico, `SiteContextConfigInitializer` lancia `Error: Cannot get base site config! ...` (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L65-L69). `SecurePortalConfigInitializer` invece lancia `Error: Cannot get current base site config .` solo se il base site attivo non è presente nella lista OCC dei base site; se nessun base site viene attivato, resta in attesa indefinita (vedi trappola 21) (core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L32-L37, core-libs/core/src/site-context/facade/base-site.service.ts:L31-L35, core-libs/core/src/site-context/facade/base-site.service.ts:L63-L68).
6. `context.baseSite` **statico disattiva tutto il caricamento** `context` **dal backend**, incluse lingue, valute e `urlParameters` (core-libs/core/src/site-context/site-context.module.ts:L37-L40). Con la sola default config, `urlParameters` resta indefinito (core-libs/core/src/site-context/config/default-site-context-config.ts:L13-L58).
7. **Scope mancanti ⇒ errore; scope sovrapposti ⇒ solo warning in dev.** (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L144-L152).
8. **Guard contro la doppia inizializzazione "tardivo".** `ongoingScopes$` è valorizzato solo dopo il loop sugli initializer; una seconda chiamata a `initialize()` ritorna subito **senza attendere** la prima (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L131-L134, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L170-L170).
9. `finishScopes` **con** `indexOf` **-1.** `splice(newScopes.indexOf(scope), 1)` rimuoverebbe l'ultimo elemento se lo scope non fosse presente; in pratica non accade perché gli scope sono aggiunti in `initialize` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L77-L83).
10. **Senza** `ConfigInitializerModule.forRoot()` **la config è "sempre stabile".** Manca il guard, `isStable` è vero, `getStable()` emette subito e nessuno esegue i `CONFIG_INITIALIZER` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L42-L44, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L47-L50).
11. **Due** `locationInitializedFactory`**.** Una pubblica deprecata in `config-initializer.module.ts` (con parametro) e una privata in `routing.module.ts`, che è quella realmente usata per `LOCATION_INITIALIZED` (core-

libs/core/src/config/config-initializer/config-initializer.module.ts:L35-L39, core-libs/core/src/routing/routing.module.ts:L66-L71).

12. **Ordine dei provider e override di** `UrlSerializer`**.** `SiteContextModule.forRoot()` deve arrivare dopo `RouterModule.forRoot` (core-libs/core/src/base-core.module.ts:L38-L38). Nell'app demo `AppRoutingModule` precede `SpartacusModule` (projects/storefrontapp/src/app/app.module.ts:L16-L22); con l'algoritmo imports-first e MWP dopo gli import del contenitore (angular:packages/core/src/di/provider_collection.ts:L293-L355) i provider di `RouterModule.forRoot` sono raccolti prima di quelli di `SiteContextModule.forRoot`, quindi vince l'override. DA VERIFICARE: deduzione dalla lettura del walker, non da esecuzione. Invertire l'ordine in `AppModule.imports` potrebbe rompere il serializer.
13. **Senza il** `RoutingModule` **di core i** `LOCATION_INITIALIZED_MULTI` **non contano.** L'override di `LOCATION_INITIALIZED` vive lì: un'app che non importa `BaseStorefrontModule` (o il routing di core) partirebbe con la navigazione senza aspettare config e auth (core-libs/core/src/routing/routing.module.ts:L102-L105, angular:packages/router/src/provide_router.ts:L447-L450).
14. `enabledBlocking` **fa aspettare anche i guard.** Il bootstrap del root component attende `LOCATION_INITIALIZED` e la navigazione iniziale fino ad `afterPreactivation`, quindi anche `CmsPageGuard` sulla rotta `**` (angular:packages/router/src/provide_router.ts:L452-L470, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L16).
15. `push` **vs** `resetConfig`**.** `addCmsRoute` muta `router.config` in place, `ConfigurableRoutesService` usa `resetConfig` (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L21-L22, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L50-L51). L'ordine fra i due APP_INITIALIZER non è stato verificato DA VERIFICARE, ma entrambi sono sincroni e la navigazione parte in un `.then` successivo (angular:packages/router/src/provide_router.ts:L452-L469), quindi la navigazione iniziale vede comunque entrambe le modifiche; la rotta `**` non ha `data.cxRoute`, quindi `configureRoutes` la lascerebbe invariata.
16. **Effetti collaterali all'import.** `registerLocaleData` nell'app demo gira al caricamento del modulo ES, non in un initializer (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L29-L31).
17. `layoutConfig` **vs** `layoutConfigFactory`**.** La costante deprecata conserva `pageFold` e l'header `lg`: chi usa `provideConfig(layoutConfig)` ottiene rendering differito sotto il fold e possibile CLS (core-libs/storefront/recipes/config/layout-config.ts:L55-L57, core-libs/storefront/recipes/config/layout-config.ts:L158-L171). La deep copy via `JSON.parse(JSON.stringify(...))` funziona perché `layoutConfig` è puro JSON (core-libs/storefront/recipes/config/layout-config.ts:L191-L191).
18. **Moduli di test nell'app demo.** `TestConfigModule.forRoot({ cookie: 'cxConfigE2E' })` legge la config da cookie ed è marcato "DON'T USE IT IN PRODUCTION!" (projects/storefrontapp/src/app/private/private.providers.ts:L52-L56, core-libs/core/src/config/test-config.module.ts:L70-L106). Non va copiato in progetti reali.
19. `MODULE_INITIALIZER` **in un modulo lazy non eredita quelli del parent.** È letto con `{ self: true }` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L153-L157); nei moduli eager gira invece nell'APP_INITIALIZER di `LazyLoadingModule` (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L44-L48).
20. **Server senza** `BootstrapContext`**.** `bootstrapApplication` lato server senza `context` lancia `PLATFORM_NOT_FOUND` (angular:packages/core/src/application/create_application.ts:L47-L54). Vedi cap. 07 e cap. 39.
21. **Catena nascosta** `routing`**/**`authentication` **→ base site attivo.** Nell'app demo lo scope `routing` del secure portal e gli scope `authentication.*` di `AuthConfigInitializer` (toggle `asyncAuthConfigInitializer` attivo) attendono un base site attivo (core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L35-L41), impostato da un initializer *non atteso* (`BaseSiteInitializer`); se l'attivazione non avviene mai, `getStable()` non emette e la navigazione iniziale resta bloccata (core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L31-L41, core-libs/

core/src/site-context/facade/base-site.service.ts:L31-L36, core-libs/core/src/site-context/services/base-site-initializer.ts:L57-L64).

---

10. Come fare debug

1. **"L'app resta su `Loading...`".** Il root component non è stato creato: `donePromise` non si è risolta (angular:packages/core/src/platform/bootstrap.ts:L143-L171). Le cause tipiche, in ordine di probabilità:
   - un `CONFIG_INITIALIZER` che non risolve: mettere un breakpoint in `ConfigInitializerService.initialize()` e ispezionare `ongoingScopes$` per vedere quale scope non viene chiuso (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L154-L170);
   - una richiesta OCC `basesites` lenta o fallita (usata da `SiteContextConfigInitializer` e, indirettamente, da `SecurePortalConfigInitializer`) — controllare la scheda Network (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L46-L47, core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L31-L32);
   - un `LOCATION_INITIALIZED_MULTI` che non risolve: breakpoint nella `locationInitializedFactory` privata di `routing.module.ts` (core-libs/core/src/routing/routing.module.ts:L66-L71);
   - un `MODULE_INITIALIZER` eager che rigetta: cercare in console "MODULE_INITIALIZER promise was rejected during app initialization." (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L28-L33).
2. **Errore "Cannot get base site config!"** — l'URL corrente non matcha gli `urlPatterns` di nessun base site restituito da OCC: o si configura `context.baseSite` staticamente, o si corregge il backend (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L61-L69).
3. **Errore "CONFIG_INITIALIZER should provide scope!"** — un initializer custom ha `scopes` vuoto (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L144-L146). Warning "More than one CONFIG_INITIALIZER is initializing the same config scope." — due initializer si sovrappongono (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L148-L152).
4. **Validazione config.** In dev mode `validateConfig` gira quando la config è stabile (core-libs/core/src/config/config-validator/config-validator.module.ts:L28-L36); un messaggio come "Please configure context.parameters.baseSite before using storefront library!" viene da `baseSiteConfigValidator` (core-libs/core/src/site-context/config/base-site-config-validator.ts:L11-L15).
5. **Rotta configurata ma non raggiunta.** In dev mode `RoutingConfigService` avvisa "No path was configured for the named route" (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L21-L34). Ricordare la trappola 3: le rotte usano la config statica.
6. **Ispezionare l'ordine degli APP_INITIALIZER.** Un breakpoint nel ciclo `for (const appInits of this.appInits)` di `runInitializers()` (angular:packages/core/src/application/application_init.ts:L245-L255) permette di vedere l'array effettivo e il valore di ritorno di ciascuno (Promise o `void`): è il modo più diretto per sciogliere i "DA VERIFICARE" sull'ordine.
7. **Stato NgRx.** Nell'app demo, fuori produzione, `StoreDevtoolsModule.instrument()` è attivo (projects/storefrontapp/src/app/private/private.providers.ts:L52-L56): le azioni `SetActiveBaseSite` (core-libs/core/src/site-context/facade/base-site.service.ts:L72-L82) mostrano quando il contesto diventa attivo rispetto alla navigazione iniziale.
8. **Config iniettata dagli e2e.** Se il comportamento differisce fra sviluppo e test, verificare il cookie `cxConfigE2E` letto da `TestConfigModule` (projects/storefrontapp/src/app/private/private.providers.

ts:L54-L54).

---

11. Domande di autoverifica

1. **In quale fase vengono istanziate le classi NgModule di Spartacus, e perché questo conta per** `Config`**?** Durante `resolveInjectorInitializers()`, prima degli APP_INITIALIZER, perché ogni NgModule ha un `ENVIRONMENT_INITIALIZER` che fa `inject(defType)`; `ConfigModule` inietta `ConfigurationService`, che materializza `Config` con la sola config statica. — angular:packages/core/src/platform/bootstrap.ts:L89-L90, angular:packages/core/src/di/provider_collection.ts:L338-L344, core-libs/core/src/config/config.module.ts:L13-L19.
2. **Un APP_INITIALIZER che chiama** `service.initialize()` **senza** `return` **blocca il bootstrap?** No: `runInitializers()` raccoglie solo Promise e Observable; `void` è ignorato. È il caso di language/currency/baseSite. — angular:packages/core/src/application/application_init.ts:L245-L256, core-libs/core/src/site-context/providers/context-initializer-providers.ts:L20-L26.
3. **Cosa trattiene la navigazione iniziale finché la config Spartacus non è stabile?** `RouterModule.forRoot` con `enabledBlocking` attende `LOCATION_INITIALIZED`; Spartacus lo sovrascrive con `Promise.all` dei `LOCATION_INITIALIZED_MULTI`, tra cui `getStable()` di `ConfigInitializerModule`. — core-libs/storefront/router/app-routing.module.ts:L13-L19, angular:packages/router/src/provide_router.ts:L445-L452, core-libs/core/src/routing/routing.module.ts:L66-L71, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64.
4. **Quando viene caricata dal backend la config** `context`**?** Solo se `context.baseSite` non è configurato staticamente; altrimenti `initSiteContextConfig` restituisce `null`. — core-libs/core/src/site-context/site-context.module.ts:L30-L41.
5. `getStable('context')` **emette se è ancora in corso lo scope** `'routing'`**? E** `getStable()`**?** `getStable('context')` sì, perché `'context'` e `'routing'` non si sovrappongono; `getStable()` senza argomenti no, perché richiede zero scope in corso. — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L91-L122.
6. **Perché** `ConfigurableRoutesService` **e** `addCmsRoute` **non iniettano il** `Router` **nel costruttore/**`deps`**?** Per una dipendenza ciclica con APP_INITIALIZER (il Router dipende dagli initializer): lo ottengono con `injector.get(Router)` al momento dell'esecuzione. — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L48-L51, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L20-L22.
7. **Che cosa succede se un** `configFactory()` **rigetta?** `initialize()` rigetta, quindi l'APP_INITIALIZER rigetta, `donePromise` rigetta e il bootstrap fallisce; lo scope non viene chiuso. — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L156-L174, angular:packages/core/src/application/application_init.ts:L263-L269.
8. **Che differenza c'è fra** `OutletModule.forRoot()` **e** `OutletModule.forChild()` **al bootstrap?** `forRoot()` registra `registerOutletsFactory` come APP_INITIALIZER (all'avvio), `forChild()` come `MODULE_INITIALIZER` (al caricamento del modulo lazy, letto con `{ self: true }`). — core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157.
9. **Perché in un progetto nuovo conviene** `provideConfigFactory(layoutConfigFactory)` **invece di** `provideConfig(layoutConfig)`**?** `layoutConfig` è deprecato e conserva `pageFold` e l'header `lg`; la factory produce header unificato e niente `pageFold`, evitando regressioni CLS. — core-libs/storefront/recipes/config/layout-config.ts:L13-L27, core-libs/storefront/recipes/config/layout-config.ts:L158-L198.

10. **Perché** `main.server.ts` **esporta una funzione che riceve** `context` **invece di chiamare direttamente** `bootstrapApplication`**?** In server mode Angular richiede un `BootstrapContext` (con `platformRef`), altrimenti lancia `PLATFORM_NOT_FOUND`. — `projects/storefrontapp/src/main.server.ts:L23-L26`, `angular:packages/core/src/application/create_application.ts:L47-L54`.

TAVOLA 10.1 **4.1 Sequenza end-to-end del bootstrap browser**

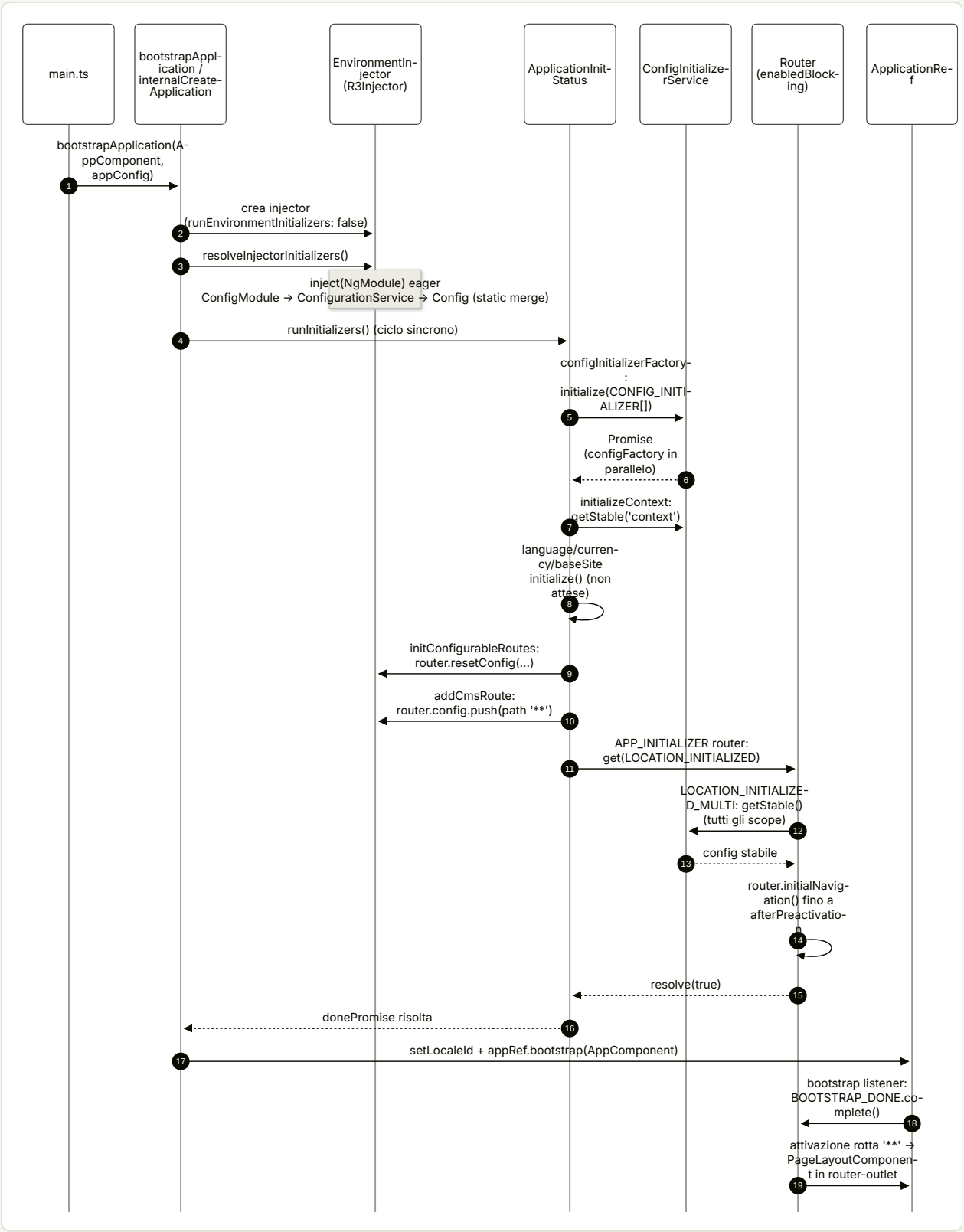


TAVOLA 10.2 **4.2 Albero dei moduli importati al bootstrap (app demo)**

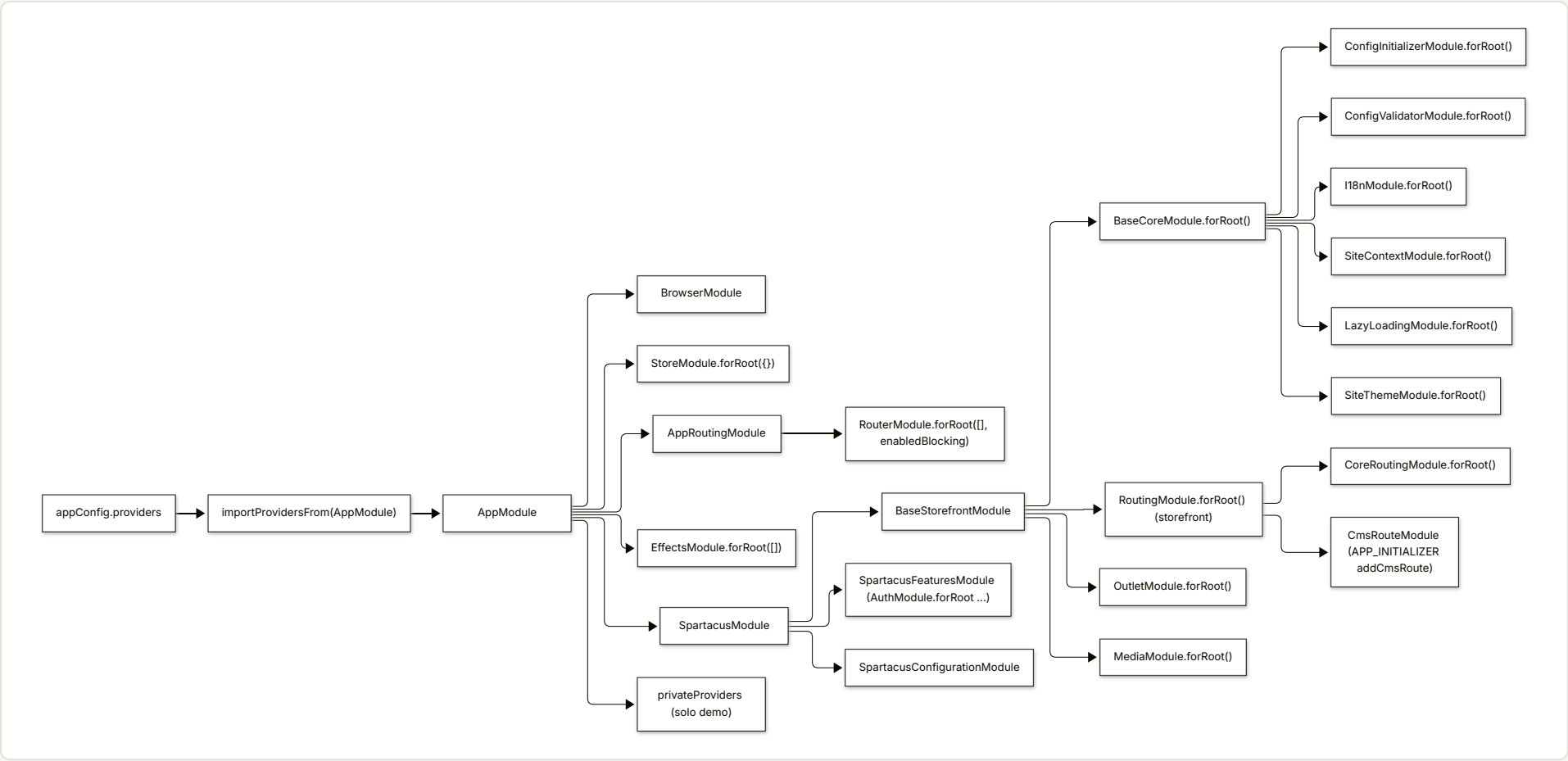
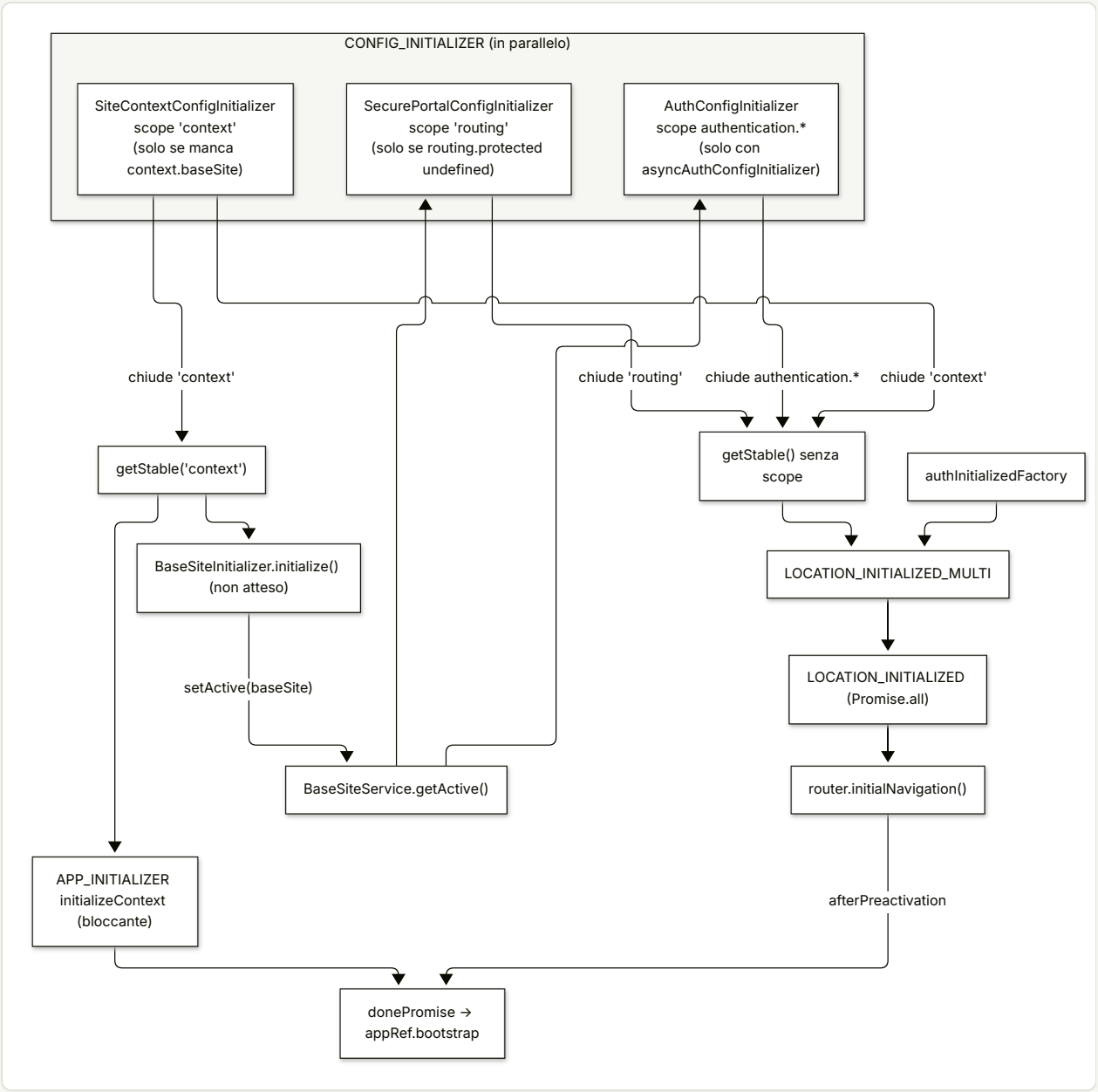


TAVOLA 10.3 **4.3 Chi aspetta chi: scope di config e Promise bloccanti**



11

Configurazione: provideConfig, default, merge, validator, initializer

Chunk di config, deep merge, `CONFIG_INITIALIZER`, config unificata dei moduli lazy, validator.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

In Spartacus quasi ogni comportamento è guidato dalla configurazione: l'URL del backend OCC, i parametri di site context, le route semantiche, il mapping dei componenti CMS, i moduli lazy, le traduzioni, i feature toggle. Il sottosistema descritto in questo capitolo risponde a quattro domande:

1. **Come si contribuisce configurazione?** Con frammenti ("chunk") registrati come multi-provider DI su due token distinti, uno per le librerie (`DefaultConfigChunk`) e uno per l'applicazione (`ConfigChunk`) — core-libs/core/src/config/config-tokens.ts:L49-L62.
2. **Come diventano un unico oggetto?** Con `deepMerge`, un merge ricorsivo che muta il primo argomento — core-libs/core/src/config/utils/deep-merge.ts:L18-L49 — eseguito dalle factory dei token `DefaultConfig`, `RootConfig` e `Config` — core-libs/core/src/config/config-tokens.ts:L10-L47.
3. **Come si gestisce la configurazione che arriva in ritardo?** In due modi:
   - durante il bootstrap, con i `CONFIG_INITIALIZER` (config asincrona, ad esempio dal backend) orchestrati da `ConfigInitializerService` — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L130-L175;
   - dopo il bootstrap, con la "config unificata" di `ConfigurationService`, che raccoglie i chunk dichiarati nei moduli caricati in lazy loading — core-libs/core/src/config/services/configuration.service.ts:L57-L106.
4. **Come si segnala una config sbagliata?** Con i `ConfigValidator`, funzioni che restituiscono un messaggio e vengono eseguite solo in dev mode — core-libs/core/src/config/config-validator/config-validator.module.ts:L23-L37.

Il punto di design centrale è la separazione tra **default** (librerie) e **root** (applicazione): qualsiasi `ConfigChunk` vince su qualsiasi `DefaultConfigChunk`, indipendentemente dall'ordine di import, perché i due insiemi vengono fusi separatamente e poi combinati con il root per ultimo — core-libs/core/src/config/config-tokens.ts:L10-L12. Il JSDoc lo dice in chiaro: "General rule is, that all config provided in libraries should be provided as default config" — core-libs/core/src/config/config-tokens.ts:L58-L58. Una regola ESLint di workspace lo impone sui file `**/*.module.ts` del monorepo (non sull'app demo) — tools/eslint-rules/rules/use-provide-default-config.ts:L22-L49, eslint.config.mjs:L231-L239, projects/storefrontapp/eslint.config.mjs:L12-L13.

Questo capitolo presuppone i concetti di DI Angular (multi provider, `InjectionToken` con `providedIn`, `inject()`) trattati in 01 — Dependency Injection, e si collega all'ordine di bootstrap descritto in 10 — Bootstrap e ordine di inizializzazione.

2. Mappa dei file

Tutti i path sono relativi a `core-libs/core/src/config/` salvo dove indicato.

PATH	RUOLO
core-libs/core/src/config/config-tokens.ts	Classe-token `Config` e token `DefaultConfig`, `RootConfig`, `ConfigChunk`, `DefaultConfigChunk` con le factory di merge (`configFactory`, `defaultConfigFactory`, `rootConfigFactory`)

PATH	RUOLO
core-libs/core/src/config/config-factory.ts	Tipo `ConfigFactory = (...props: any[]) => Config`
core-libs/core/src/config/config-providers.ts	Helper `provideConfig`, `provideConfigFactory`, `provideDefaultConfig`, `provideDefaultConfigFactory`
core-libs/core/src/config/config.module.ts	`ConfigModule` (`withConfig`, `withConfigFactory`, `forRoot`); istanzia `ConfigurationService`
core-libs/core/src/config/index.ts	Barrel pubblico della cartella (riesportato da `core-libs/core/public_api.ts`)
core-libs/core/src/config/config-initializer/config-initializer.ts	Interfaccia `ConfigInitializer`, token `CONFIG_INITIALIZER` e `CONFIG_INITIALIZER_FORROOT_GUARD`, helper `provideConfigInitializer`, `provideConfigInitializerFactory`
core-libs/core/src/config/config-initializer/config-initializer.service.ts	`ConfigInitializerService`: scope in corso, `getStable`, `initialize`
core-libs/core/src/config/config-initializer/config-initializer.module.ts	`ConfigInitializerModule.forRoot()`: guard, `APP_INITIALIZER`, location initializer
core-libs/core/src/config/config-initializer/index.ts	Barrel (non esporta il guard)
core-libs/core/src/config/config-validator/config-validator.ts	`ConfigValidatorToken`, tipo `ConfigValidator`, `provideConfigValidator`, `validateConfig`
core-libs/core/src/config/config-validator/config-validator.module.ts	`ConfigValidatorModule.forRoot()`: `APP_INITIALIZER` di validazione in dev mode
core-libs/core/src/config/config-validator/index.ts	Barrel (`export *` di entrambi i file)
core-libs/core/src/config/services/configuration.service.ts	`ConfigurationService`: config unificata con i chunk dei moduli lazy (`unifiedConfig$`)
core-libs/core/src/config/test-config.module.ts	`TestConfigModule`, `TEST_CONFIG`, `TEST_CONFIG_COOKIE_NAME`, `TestConfig`: config da cookie per i test e2e
core-libs/core/src/config/utils/deep-merge.ts	`deepMerge`, `isObject`
core-libs/core/src/config/utils/get-cookie.ts	`getCookie` (non esportato dal barrel)
core-libs/core/src/config/utils/string-template.ts	`StringTemplate.resolve`: interpolazione `${var}` usata per gli endpoint

File correlati fuori dalla cartella, citati nel capitolo:

PATH	RUOLO NEL FLUSSO DI CONFIG
core-libs/core/src/base-core.module.ts	Importa `ConfigModule.forRoot()`, `ConfigInitializerModule.forRoot()`, `ConfigValidatorModule.forRoot()`
core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts	Token `LOCATION_INITIALIZED_MULTI` e `provideLocationInitializerFactory`
core-libs/core/src/routing/routing.module.ts	Fornisce `LOCATION_INITIALIZED` aggregando i `LOCATION_INITIALIZED_MULTI`; registra `SecurePortalConfigInitializer`

PATH	RUOLO NEL FLUSSO DI CONFIG
core-libs/core/src/lazy-loading/unified-injector.ts	UnifiedInjector: legge un token da root + injector dei moduli lazy
core-libs/core/src/lazy-loading/lazy-modules.service.ts	LazyModulesService.modules$: stream dei moduli lazy istanziati
core-libs/core/src/site-context/site-context.module.ts	Esempio di CONFIG_INITIALIZER condizionale e di validator
core-libs/core/src/i18n/i18n.module.ts	Altro CONFIG_INITIALIZER condizionale
tools/eslint-rules/rules/use-provide-default-config.ts	Regola lint che vieta provideConfig() (attiva come error solo sui file **/*.module.ts)

3. Come funziona passo per passo

3.1 Il token `Config` e i tre livelli di oggetti

`Config` è una classe astratta vuota, decorata `@Injectable({ providedIn: 'root', useFactory: configFactory })` — core-libs/core/src/config/config-tokens.ts:L14-L21. Essendo vuota, la sua forma è data interamente dalla *module augmentation*: ogni area dichiara `interface Config extends XxxConfig {}`. Nel core l'augmentation punta al modulo relativo `config-tokens` — core-libs/core/src/site-context/config/site-context-config.ts:L21-L23 — mentre nelle feature lib punta a `'@spartacus/core'` — feature-libs/cart/base/root/config/cart-config.ts:L30-L32. Per questo `Config` è elencato tra gli "augmentable types" del public API — core-libs/core/public_api.ts:L38-L40.

Le classi di config specifiche (`SiteContextConfig`, `I18nConfig`, ...) sono anch'esse classi astratte iniettabili, ma con `useExisting: Config`: **sono alias dello stesso oggetto globale**, non oggetti separati — core-libs/core/src/site-context/config/site-context-config.ts:L10-L19, core-libs/core/src/i18n/config/i18n-config.ts:L14-L23. Iniettare `SiteContextConfig` significa ottenere l'istanza di `Config` con un tipo più stretto.

Ci sono tre oggetti di configurazione in gioco, tutti singleton in root:

OGGETTO	COSTRUITO DA	CONTENUTO
DefaultConfig	defaultConfigFactory	merge di tutti i DefaultConfigChunk — core-libs/core/src/config/config-tokens.ts:L23-L35
RootConfig	rootConfigFactory	merge di tutti i ConfigChunk — core-libs/core/src/config/config-tokens.ts:L37-L47
Config	configFactory	deepMerge({}, DefaultConfig, RootConfig) — core-libs/core/src/config/config-tokens.ts:L10-L12

Entrambi i chunk token sono `InjectionToken<Config[]>` **senza factory**: se nessuno li fornisce, `inject(..., { optional: true })` restituisce `null` e il codice ripiega su `[]` — core-libs/core/src/config/config-tokens.ts:L26-L26, L38-L38. Il tipo `Config[]` riflette il fatto che sono multi-token (array di chunk) — core-libs/core/src/config/config-tokens.ts:L53-L53, L60-L62.

Le tre factory non sono esportate dal barrel: `index.ts` riesporta da `config-tokens` solo i cinque simboli `Config`, `ConfigChunk`, `DefaultConfig`, `DefaultConfigChunk`, `RootConfig` — core-libs/core/src/config/index.ts:L10-L16.

3.2 Contribuire configurazione: gli helper `provide*`

I quattro helper sono semplici costruttori di provider multi:

HELPER	PROVIDER	TOKEN	RIFERIMENTO
`provideConfig(config = {}, defaultConfig = false)`	`ValueProvider`	`ConfigChunk` (o `DefaultConfigChunk` se `defaultConfig`)	core-libs/core/src/config/config-providers.ts:L18-L27
`provideConfigFactory(factory, deps?, defaultConfig = false)`	`FactoryProvider`	idem	core-libs/core/src/config/config-providers.ts:L37-L48
`provideDefaultConfig(config = {})`	`ValueProvider`	`DefaultConfigChunk`	core-libs/core/src/config/config-providers.ts:L55-L61
`provideDefaultConfigFactory(factory, deps?)`	`FactoryProvider`	`DefaultConfigChunk`	core-libs/core/src/config/config-providers.ts:L69-L79

Il tipo della factory è `ConfigFactory`, una funzione con argomenti arbitrari che restituisce `Config` — core-libs/core/src/config/config-factory.ts:L7-L9. Gli argomenti arrivano da `deps` secondo la semantica classica di `FactoryProvider`; in alternativa la factory può usare `inject()` perché è eseguita in injection context (come fa `TestConfigModule`, core-libs/core/src/config/test-config.module.ts:L99-L102).

Il JSDoc di `provideConfig` ricorda che nelle librerie va usato `provideDefaultConfig` — core-libs/core/src/config/config-providers.ts:L11-L17 — e le due regole ESLint `use-provide-default-config` e `use-provide-default-config-factory` segnalano ogni `CallExpression` di `provideConfig()`/`provideConfigFactory()` — tools/eslint-rules/rules/use-provide-default-config.ts:L22-L49, tools/eslint-rules/rules/use-provide-default-config-factory.ts:L22-L46 — ma sono abilitate come `error` solo per i file `**/*.module.ts` del workspace — eslint.config.mjs:L231-L239 — e disattivate nell'app demo — projects/storefrontapp/eslint.config.mjs:L12-L13. Una chiamata in un `*.providers.ts` o `*.config.ts` non viene segnalata. L'app demo invece, da "consumer", usa `provideConfig` per backend, routing e `features.level` — projects/storefrontapp/src/app/private/private.providers.ts:L19-L50.

3.3 `ConfigModule`: la variante NgModule

`ConfigModule` offre tre metodi statici:

- `withConfig(config)` → `providers: [provideConfig(config)]` — core-libs/core/src/config/config.module.ts:L28-L34;
- `withConfigFactory(factory, deps?)` → `provideConfigFactory(factory, deps)` — core-libs/core/src/config/config.module.ts:L44-L53;
- `forRoot(config = {})` → identico a `withConfig` — core-libs/core/src/config/config.module.ts:L60-L66.

Tutti contengono un `eslint-disable-next-line` per la regola di cui sopra. Il ruolo vero del modulo è un altro: il suo costruttore inietta `ConfigurationService` "to make sure ConfigurationService will be instantiated" — core-libs/core/src/config/config.module.ts:L13-L19. È l'unico punto di libreria che lo inietta (fuori dalla cartella `config/` il grep trova solo una costante delle schematics, core-libs/schematics/src/shared/constants.ts:L328-L328, e una menzione nel JSDoc di `disableConfigUpdates`, core-libs/core/src/features-config/config/features-config.ts:L112-L112).

`BaseCoreModule` importa, in quest'ordine, `ErrorHandlingModule.forRoot()`, `StateModule.forRoot()`, `ConfigModule.forRoot()`, `ConfigInitializerModule.forRoot()`, `ConfigValidatorModule.forRoot()` — core-

libs/core/src/base-core.module.ts:L26-L32. Quindi in una storefront standard i tre moduli di config sono sempre presenti.

3.4 Costruzione sincrona della `Config` globale

La `Config` viene costruita **pigramente**, alla prima iniezione di `Config` o di un suo alias `useExisting`:

1. `inject(Config)` → Angular esegue `configFactory` — core-libs/core/src/config/config-tokens.ts:L17-L21.
2. `configFactory` chiede `DefaultConfig` → `defaultConfigFactory` fonde i `DefaultConfigChunk` nell'ordine dei provider — core-libs/core/src/config/config-tokens.ts:L23-L28.
3. Poi chiede `RootConfig` → `rootConfigFactory` fonde i `ConfigChunk` — core-libs/core/src/config/config-tokens.ts:L37-L39.
4. `deepMerge({}, DefaultConfig, RootConfig)` produce l'oggetto `Config` — core-libs/core/src/config/config-tokens.ts:L11-L11.

L'ordine dei chunk all'interno dello stesso token è l'ordine in cui l'`R3Injector` li ha accumulati nel record multi (`multiRecord.multi!.push(provider)`) — angular:packages/core/src/di/r3_injector.ts:L497-L512. Tra provider dichiarati in `makeEnvironmentProviders` e quelli provenienti da `importProvidersFrom` vale l'ordine di posizione nell'array: `makeEnvironmentProviders` conserva l'array così com'è — angular:packages/core/src/di/provider_collection.ts:L46-L52 — `importProvidersFrom` restituisce l'elenco dei provider raccolti in ordine (`providersOut.push(provider)`, con i `ModuleWithProviders` elaborati dopo i moduli semplici) — angular:packages/core/src/di/provider_collection.ts:L151-L189 — e l'`R3Injector` appiattisce array ed `EnvironmentProviders` con una visita in profondità ordinata — angular:packages/core/src/di/r3_injector.ts:L725-L738. Nell'app demo, quindi, il `provideConfig` di `private.providers.ts` precede il `ConfigChunk` di `TestConfigModule.forRoot()`, che vince — projects/storefrontapp/src/app/private/private.providers.ts:L19-L56.

3.5 L'algoritmo di `deepMerge`

`deepMerge(target = {}, ...sources)`:

1. Se non restano sorgenti restituisce `target` — core-libs/core/src/config/utils/deep-merge.ts:L22-L24.
2. Estrae la prima sorgente con `sources.shift()`; una sorgente falsy diventa `{}` — core-libs/core/src/config/utils/deep-merge.ts:L25-L25.
3. Se la sorgente non è un oggetto "semplice" (array, primitivo) la salta e ricorre sulle restanti — core-libs/core/src/config/utils/deep-merge.ts:L27-L29. `isObject` è vero solo per oggetti non nulli e non array — core-libs/core/src/config/utils/deep-merge.ts:L7-L9.
4. Per ogni chiave enumerabile (`for...in`) — core-libs/core/src/config/utils/deep-merge.ts:L31-L46:
   - salta `__proto__` e `constructor` (protezione da prototype pollution) — core-libs/core/src/config/utils/deep-merge.ts:L11-L16, L32-L34;
   - un `Date` viene copiato per riferimento — L36-L37;
   - un oggetto viene fuso ricorsivamente, creando prima `{}` nel target se lì non c'è un oggetto — L38-L42;
   - tutto il resto (primitivi **e array**) sovrascrive — L43-L45.
5. Ricorre sulle sorgenti rimanenti e restituisce `target` — core-libs/core/src/config/utils/deep-merge.ts:L48-L48.

Due conseguenze pratiche:

- gli array **non si concatenano**: lo spec lo verifica ("should overrite arrays") — core-libs/core/src/config/utils/deep-merge.spec.ts:L86-L86;
- poiché `configFactory` parte da `{}` e il ramo oggetto crea nuovi `{}` (L39-L41), la `Config` globale ha **oggetti annidati propri** ma **array e** `Date` **condivisi** con i chunk sorgente (L36-L37, L43-L45).

`deepMerge` e `isObject` sono API pubbliche — core-libs/core/src/config/index.ts:L27-L27.

3.6 Config asincrona: `CONFIG_INITIALIZER`

Alcune parti di config possono essere note solo a runtime (es. il base site dipende dall'URL e dalla risposta del backend). Il contratto è l'interfaccia `ConfigInitializer` — core-libs/core/src/config/config-initializer/config-initializer.ts:L15-L28:

- `scopes: string[]` — le parti di config che l'initializer risolve, es. `'context.baseSite'`;
- `configFactory: () => Promise<Config>` — la Promise che produce il chunk.

Gli initializer si registrano sul token `CONFIG_INITIALIZER` (`InjectionToken<ConfigInitializer | null>`, multi per convenzione) — core-libs/core/src/config/config-initializer/config-initializer.ts:L30-L35 — con `provideConfigInitializer(Classe)` (`ClassProvider` multi) — L46-L52 — o `provideConfigInitializerFactory(() => initializer | null)` (`FactoryProvider` multi) — L59-L67. La factory può restituire `null` per disattivare l'initializer.

Registrazione del motore. `ConfigInitializerModule.forRoot()` fornisce tre cose — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L41-L68:

1. il guard `CONFIG_INITIALIZER_FORROOT_GUARD` con valore `true` (L47-L50) — un token non esportato dal barrel (core-libs/core/src/config/config-initializer/index.ts:L7-L14) che segnala la presenza del modulo;
2. un `APP_INITIALIZER` multi con `configInitializerFactory`, che riceve `ConfigInitializerService` e l'array opzionale dei `CONFIG_INITIALIZER` (L51-L59) e restituisce `() => configInitializer.initialize(initializers)` (L24-L30);
3. un *location initializer* che trattiene la navigazione iniziale finché `getStable()` non emette (L60-L64) — vedi §3.8.

`initialize`. Il metodo è marcato `@internal` — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L129 — e procede così:

1. Se `ongoingScopes$.value` è già definito esce: "guard for double initialization" — L131-L134.
2. Per ogni initializer — L140-L169:
   - salta i valori `null`/falsy (L141-L143);
   - lancia `Error('CONFIG_INITIALIZER should provide scope!')` se mancano gli scope (L144-L146);
   - in dev mode, se gli scope si sovrappongono a quelli già registrati, logga un warning (L148-L152);
   - aggiunge gli scope a `ongoingScopes` e avvia subito un'IIFE async che attende `configFactory()`, fonde il risultato **prima in** `rootConfig` **poi in** `config`, e chiama `finishScopes` (L154-L168).
3. Solo dopo aver avviato tutti gli initializer emette su `ongoingScopes$` l'elenco completo (L170-L170).
4. Attende `Promise.all` di tutte le IIFE (L172-L174). Se una factory rifiuta, rifiuta `initialize` e quindi l'`APP_INITIALIZER`; Angular raccoglie le Promise di tutti gli initializer e rifiuta l'inizializzazione dell'app — angular:packages/core/src/application/application_init.ts:L239-L269. Lo spec copre il caso ("should fail if one initializer will fail") — core-libs/core/src/config/config-initializer/config-initializer.service.spec.ts:L195-L195.

Il merge in `rootConfig` oltre che in `config` mantiene coerente la config unificata dei moduli lazy, che riapplica `rootConfig` per ultimo (§3.9, core-libs/core/src/config/services/configuration.service.ts:L93-L99): senza questo passaggio un default di un modulo lazy potrebbe sovrascrivere un valore caricato dal backend. (Interpretazione del codice: il sorgente commenta solo "contribute configuration to rootConfig", config-initializer.service.ts:L159-L159.)

3.7 Attendere la stabilità: `getStable` e gli scope

`ConfigInitializerService` mantiene lo stato in `ongoingScopes$`, un `BehaviorSubject<string[] | undefined>` che parte da `undefined` ("initialize non ancora eseguito") — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L35-L37.

- `isStable` è `true` se **il guard è assente** oppure se gli scope in corso sono un array vuoto — `L42-L44`.
- `getStable(...scopes)` restituisce `of(this.config)` se stabile; altrimenti filtra `ongoingScopes$` scartando `undefined` e le emissioni in cui gli scope richiesti non sono pronti, prende la prima (`take(1)`) e la mappa su `this.config` — `L58-L70`. Il JSDoc lo indica come "Recommended way to get config for code that can run before app will finish initialization" — `L46-L57`.
- `areReady(scopes, ongoing)`: senza scope richiesti è vero solo se non c'è nulla in corso; altrimenti è falso se uno scope richiesto si sovrappone a uno in corso — `L91-L103`.
- `scopesOverlap(a, b)`: sovrapposti se uno è prefisso dell'altro con confine a `.` — `'test'`/`'test.a'` sì, `'test'`/`'testA'` no — `L105-L122`.
- `finishScopes` rimuove gli scope conclusi da una copia dell'array ed emette — `L77-L83`.

Il fatto che l'elenco completo venga emesso solo **dopo** l'avvio di tutte le factory (`L170-L170`) abilita le dipendenze tra initializer: `I18nConfigInitializer` (scope `['i18n.fallbackLang']`) dentro la propria `configFactory` chiama `getStable('context.language')` — core-libs/core/src/i18n/config/i18n-config-initializer.ts:L14-L33. In quell'istante `ongoingScopes$` vale ancora `undefined`, `isStable` è falso e il filtro `!!ongoingScopes` lo fa aspettare la lista completa; poi attende che lo scope `context` (registrato da `SiteContextConfigInitializer`) si chiuda.

Consumatori tipici di `getStable` con scope: `CmsComponentsService` (`'cmsComponents'`) — core-libs/storefront/cms-structure/services/cms-components.service.ts:L61-L62; `CmsFeaturesService` (`'featureModules'`) — core-libs/storefront/cms-structure/services/cms-features.service.ts:L57-L58; `DirectionService` (`'direction'`) — core-libs/storefront/layout/direction/direction.service.ts:L48-L48; `OAuthLibWrapperService` (`'authentication'`) — core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L59-L59; la persistenza del carrello (`'context'`) — feature-libs/cart/base/core/cart-persistence.module.ts:L25-L25. Inoltre un `APP_INITIALIZER` del site context (`initializeContext`) attende `getStable('context')` — core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L21 (registrazione come `APP_INITIALIZER` a `L27-L31`).

3.8 Bloccare la navigazione iniziale

Il terzo provider di `ConfigInitializerModule` è `provideLocationInitializerFactory(...)` — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64 — che crea un multi-provider sul token Spartacus `LOCATION_INITIALIZED_MULTI` — core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L36-L51 (pubblico via core-libs/core/src/routing/index.ts:L11-L11).

`RoutingModule.forRoot()` fornisce il token Angular `LOCATION_INITIALIZED` con una factory che esegue tutti i `LOCATION_INITIALIZED_MULTI` e ne fa `Promise.all` — core-libs/core/src/routing/routing.module.ts:L66-L71, `L102-L105`. Lo storefront configura il router con `initialNavigation: 'enabledBlocking'` — core-libs/storefront/router/app-routing.module.ts:L13-L19 — e con questa opzione Angular registra un app initiali-

zer che attende `LOCATION_INITIALIZED` prima di far partire la navigazione iniziale — angular:packages/router/src/provide_router.ts:L441-L465.

Risultato: **nessuna route viene riconosciuta prima che tutta la config Spartacus sia stabile**. Questo è essenziale perché la config di routing stessa può essere asincrona (`SecurePortalConfigInitializer`, scope `['routing']`, core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L16-L19).

3.9 Moduli lazy: `ConfigurationService` e la config unificata

Quando un modulo viene caricato in lazy loading (vedi 12 — Feature module e lazy loading), i suoi `provideDefaultConfig`/`provideConfig` finiscono nel suo injector, non nel root: un multi-provider crea un record proprio nell'`R3Injector` in cui è registrato — angular:packages/core/src/di/r3_injector.ts:L497-L512. La `Config` globale, già costruita, non li vedrebbe. Ci pensa `ConfigurationService`:

1. Viene istanziato dal costruttore di `ConfigModule` — core-libs/core/src/config/config.module.ts:L15-L19.
2. Nel costruttore inietta `RootConfig`, `DefaultConfig`, `UnifiedInjector`, `Config`; crea `unifiedConfig$` come `BehaviorSubject(config)` e si sottoscrive subito a `feedUnifiedConfig()` "to propagate changes to the config from the beginning" — core-libs/core/src/config/services/configuration.service.ts:L42-L55.
3. `feedUnifiedConfig` chiede a `UnifiedInjector` gli stream di `ConfigChunk` e `DefaultConfigChunk` con `notFoundValue []`, li accoppia con `zip` e salta la prima emissione (quella del root injector) — `L57-L74`.
4. `UnifiedInjector.injectors$` parte dal root e aggiunge l'injector di ogni modulo emesso da `LazyModulesService.modules$` — core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35; `get` interroga ogni injector con `{ self: true }` tranne il primo, ed emette una volta per injector quando c'è un `notFoundValue` — core-libs/core/src/lazy-loading/unified-injector.ts:L42-L68. Il flag `Self` fa terminare il lookup sul `NullInjector` invece che sul parent — angular:packages/core/src/di/r3_injector.ts:L374-L376 — quindi si ottengono solo i chunk propri del modulo lazy.
5. `modules$` è un `connectable` con `ReplaySubject` alimentato dagli eventi `ModuleInitializedEvent` — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L68 — che `resolveModuleInstance` dispatcha dopo aver creato il modulo ed eseguito i suoi module initializer — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L107.
6. `processConfig` accumula i default chunk in `ambientDefaultConfig` e i chunk in `ambientConfig` (accumulatori privati inizializzati a `{}`, configuration.service.ts:L37-L38) ed emette solo se almeno uno dei due array non è vuoto — core-libs/core/src/config/services/configuration.service.ts:L76-L90.
7. `emitUnifiedConfig` fonde `defaultConfig → ambientDefaultConfig → ambientConfig → rootConfig` in un oggetto nuovo e lo emette su `unifiedConfig$` — `L92-L100`.
8. Salvo il flag `features.disableConfigUpdates`, la nuova config viene anche fusa **in place** nella `Config` globale ("compatibility mechanism") — `L102-L105`.

Il flag `disableConfigUpdates` è dichiarato in `FeaturesConfig` come `@deprecated`, "disabled by default and recommended to keep it so" — core-libs/core/src/features-config/config/features-config.ts:L101-L116 — e `isFeatureEnabled` lo accetta esplicitamente come secondo argomento — core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L51. Nessun servizio di libreria consuma `unifiedConfig$`: l'unico riferimento extra è nel JSDoc del flag — core-libs/core/src/features-config/config/features-config.ts:L112-L112.

3.10 Validazione in dev mode

- `ConfigValidator` è il tipo `(config: Config) => string | void`: una stringa significa "validazione fallita" — core-libs/core/src/config/config-validator/config-validator.ts:L15-L20.
- `provideConfigValidator(fn)` registra un multi `useValue` su `ConfigValidatorToken` — core-libs/core/src/config/config-validator/config-validator.ts:L11-L13, L27-L35.
- `ConfigValidatorModule.forRoot()` registra un `APP_INITIALIZER` con deps `ConfigInitializerService` e i validator opzionali — core-libs/core/src/config/config-validator/config-validator.module.ts:L45-L63.
- `configValidatorFactory` inietta `LoggerService` con `inject()` e restituisce una funzione **sincrona** che, solo se `isDevMode()`, si sottoscrive a `getStable()` — config-validator.module.ts:L23-L37.
- Alla config stabile `validateConfig` esegue ogni validator e logga con `logger.warn` i messaggi restituiti; non lancia mai — core-libs/core/src/config/config-validator/config-validator.ts:L37-L48.

`isDevMode()` di Angular è vero se `ngDevMode` è `undefined` o truthy — angular:packages/core/src/util/is_dev_mode.ts:L20-L22. Esempio di validator: `baseSiteConfigValidator` restituisce "Please configure context.parameters.baseSite before using storefront library!" se manca il default del base site — core-libs/core/src/site-context/config/base-site-config-validator.ts:L11-L15. Altri validator registrati: `occConfigValidator` — core-libs/core/src/occ/base-occ.module.ts:L31-L31; `baseUrlConfigValidator` — core-libs/core/src/auth/user-auth/user-auth.module.ts:L107-L107; `epdVisualizationConfigValidator` — integration-libs/epd-visualization/root/epd-visualization-root.module.ts:L35-L35; `cdsConfigValidator` — integration-libs/cds/src/cds.module.ts:L42-L42.

3.11 Initializer concreti nel core

INITIALIZER	SCOPE	ATTIVAZIONE	RIFERIMENTI
`SiteContextConfigInitializer`	`['context']`	solo se manca `context.baseSite` statico (`initSiteContextConfig` restituisce `null` altrimenti)	core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L24-L27; core-libs/core/src/site-context/site-context.module.ts:L30-L41, L51-L61
`I18nConfigInitializer`	`['i18n.fallbackLang']`	solo se `i18n.fallbackLang` è `undefined`	core-libs/core/src/i18n/config/i18n-config-initializer.ts:L14-L33; core-libs/core/src/i18n/i18n.module.ts:L30-L53
`SecurePortalConfigInitializer`	`['routing']`	solo se `routing.protected` è `undefined` (`initSecurePortalConfig` restituisce `null` altrimenti, core-libs/core/src/routing/routing.module.ts:L44-L52)	core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L16-L19; core-libs/core/src/routing/routing.module.ts:L96-L101
`AuthConfigInitializer`	`['authentication.OAuthLibConfig.redirectUri', 'authentication.client_id']`	solo con feature toggle `asyncAuthConfigInitializer`	core-libs/core/src/auth/user-auth/user-auth.module.ts:L106-L113; core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L20-L24

Si notano tre stili di registrazione: provider scritto a mano con `useFactory` + `deps` (`site-context.module.ts:L56-L61`), `provideConfigInitializerFactory` con `inject()` (`routing.module.ts:L96-L101`), e un toggle con commento "When removing this feature toggle, switch to `provideConfigInitializer`" (`user-auth.module.ts:L109-L109`).

3.12 Config per i test e2e: `TestConfigModule`

`TestConfigModule.forRoot({ cookie })` — `core-libs/core/src/config/test-config.module.ts:L83-L105`:

1. fornisce il nome del cookie su `TEST_CONFIG_COOKIE_NAME`;
2. `TEST_CONFIG` (root) legge il cookie solo in browser e se il nome è definito, e ne fa il parse — `test-config.module.ts:L35-L55`; `parseConfigJSON` applica `decodeURIComponent` + `JSON.parse` e restituisce `{}` in caso di errore — `L57-L63`;
3. passa `TEST_CONFIG.features` ai feature toggle e l'intero `TEST_CONFIG` come `ConfigChunk` via `provideConfigFactory` — `L95-L102`.

La lettura del cookie usa `getCookie`, che costruisce una RegExp concatenando il nome — `core-libs/core/src/config/utils/get-cookie.ts:L7-L12`. L'app demo importa `TestConfigModule.forRoot({ cookie: 'cxConfigE2E' })` con il commento "Should be imported after other config modules" — `projects/storefrontapp/src/app/private/private.providers.ts:L52-L56`. Il JSDoc avverte: "CAUTION: DON'T USE IT IN PRODUCTION!" — `core-libs/core/src/config/test-config.module.ts:L69-L73`.

3.13 `StringTemplate`

`StringTemplate.resolve(template, variables, encodeVariable?)` sostituisce ogni `${nome}` con il valore, opzionalmente passato per `encodeURIComponent` — `core-libs/core/src/config/utils/string-template.ts:L15-L31`. È nella cartella config perché i template degli endpoint OCC sono configurazione; entrambi i chiamanti di libreria passano `true` — `core-libs/core/src/occ/services/occ-endpoints.service.ts:L122-L122`, `integration-libs/opf/base/core/services/opf-endpoints.service.ts:L47-L47`. Il dettaglio sugli endpoint è in 21 — Livello OCC.

---

4. Diagrammi

4.1 Costruzione della Config e priorità dei livelli

■ **TAVOLA 11.1** diagramma a tutta pagina, in fondo al capitolo

Ogni freccia corrisponde a: §3.2 (`config-providers.ts:L18-L79`), §3.4 (`config-tokens.ts:L10-L47`), §3.6 (`config-initializer.service.ts:L158-L165`), §3.9 (`configuration.service.ts:L92-L105`), §3.1 (`site-context-config.ts:L10-L19`), §3.12 (`test-config.module.ts:L99-L102`).

4.2 Bootstrap con config asincrona

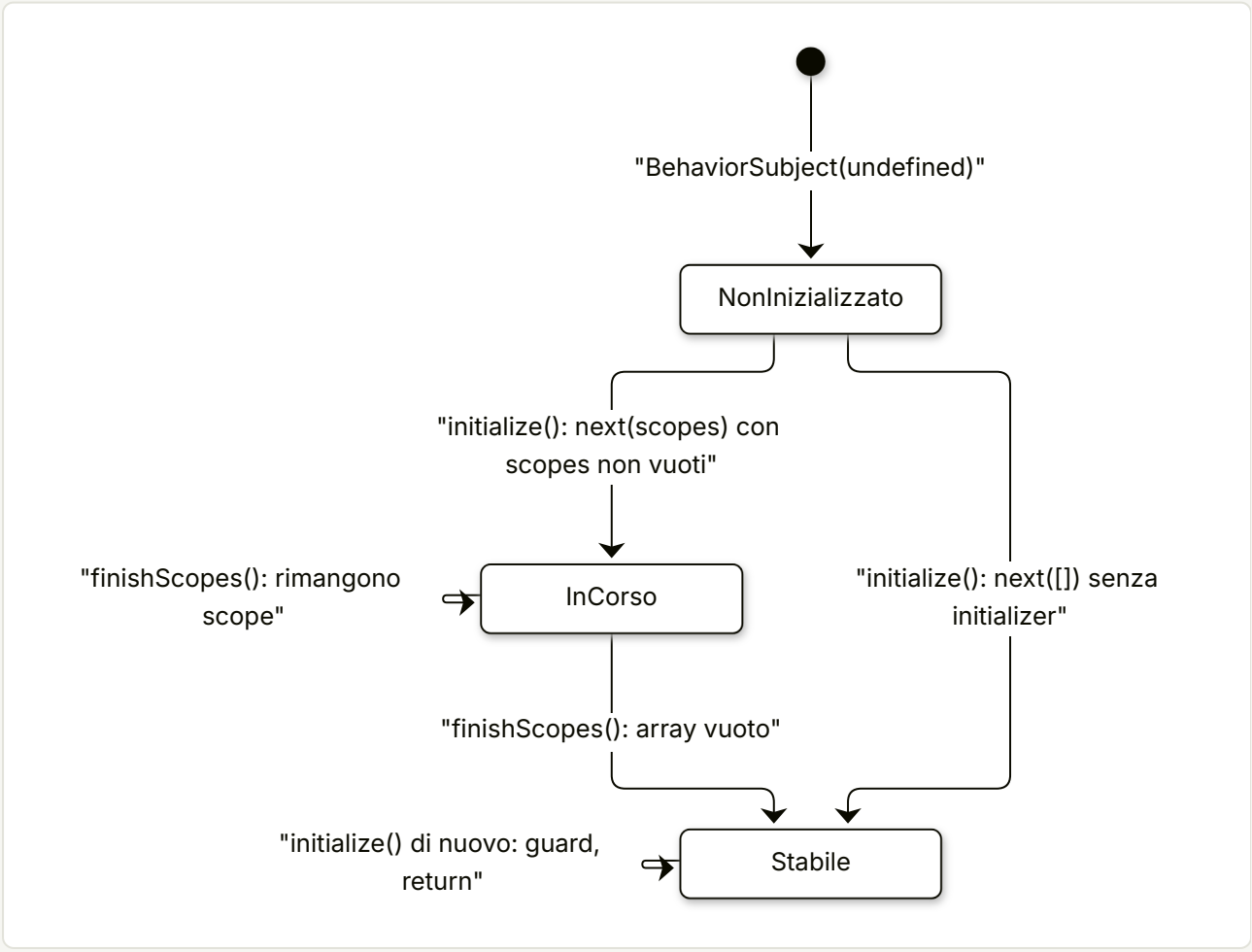
■ **TAVOLA 11.2** diagramma a tutta pagina, in fondo al capitolo

Passi: 1 → `application_init.ts:L245-L246`; 2 → `config-initializer.module.ts:L24-L30`; 3, 5 → `config-initializer.service.ts:L154-L158`; 4 → `i18n-config-initializer.ts:L25-L25` (vedi §3.7); 6 → `config-`

initializer.service.ts:L170-L170; 7 → angular:packages/core/src/application/application_init.ts:L245-L248, angular:packages/router/src/router_module.ts:L240-L241; 8 → angular:packages/router/src/provide_router.ts:L445-L452, routing.module.ts:L66-L71; 9-10 → config-initializer.service.ts:L158-L166; 11 → L58-L70; 13 → L77-L83; 14 → L172-L174; 15 → config-initializer.module.ts:L60-L64, angular:packages/router/src/provide_router.ts:L452-L469; 16-17 → angular:packages/router/src/provide_router.ts:L462-L468, angular:packages/core/src/application/application_init.ts:L263-L269.

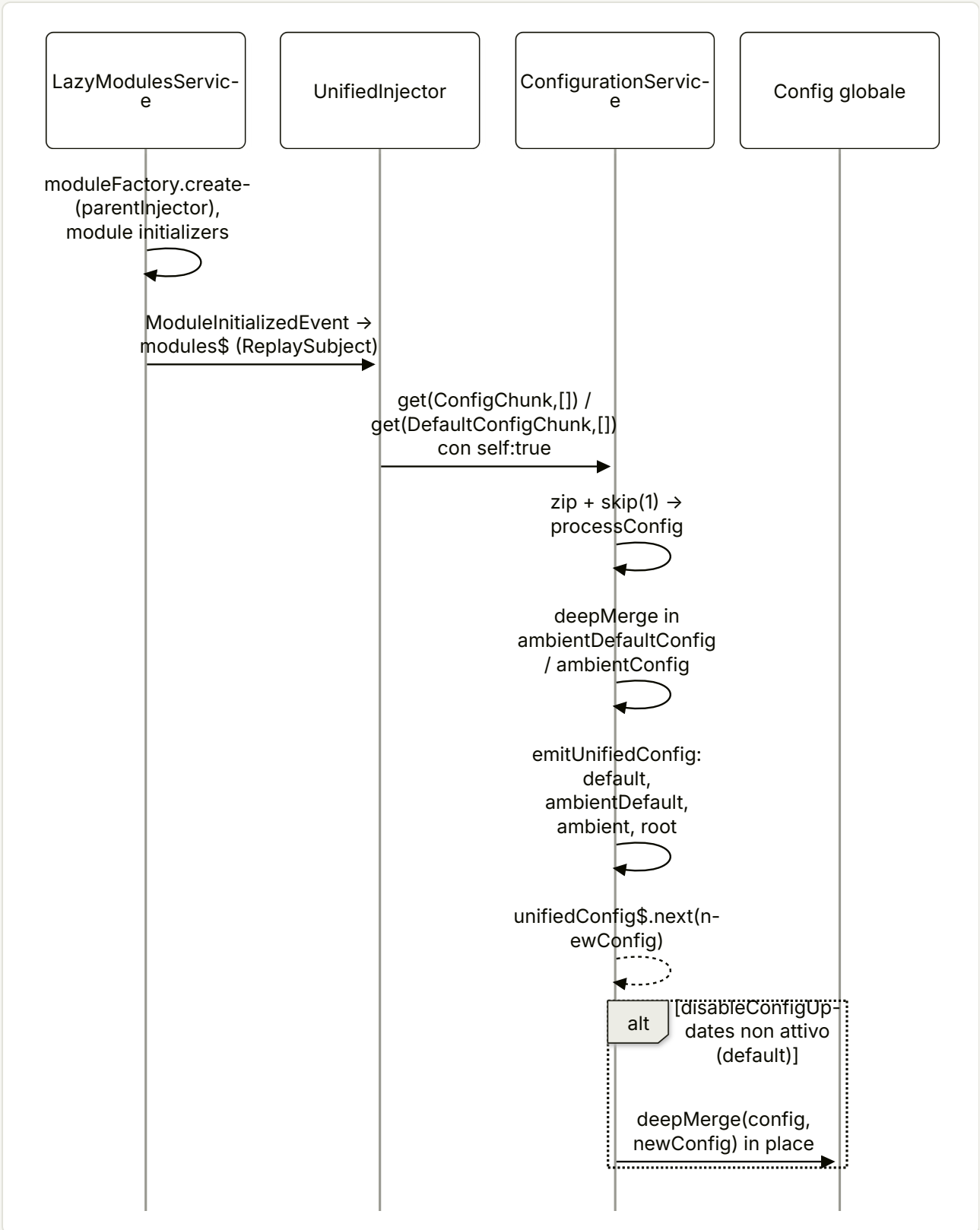
L'ordine dei passi 3-5 riflette l'ordine dei provider `CONFIG_INITIALIZER` nella storefront standard: `BaseCoreModule` importa `I18nModule.forRoot()` prima di `SiteContextModule.forRoot()` — core-libs/core/src/base-core.module.ts:L33-L38 — e Angular elabora i `ModuleWithProviders` nell'ordine di import — angular:packages/core/src/di/provider_collection.ts:L316-L320. Con un ordine diverso il risultato non cambia, perché `ongoingScopes$` viene emesso solo al passo 6. Spartacus usa `RouterModule.forRoot({initialNavigation: 'enabledBlocking'})`, che Angular traduce in `withEnabledBlockingInitialNavigation()` — angular:packages/router/src/router_module.ts:L240-L241: l'attesa di `LOCATION_INITIALIZED` è quindi essa stessa un `APP_INITIALIZER`, avviato nello stesso ciclo `for` di `runInitializers` e concorrente a `configInitializerFactory` (la posizione relativa dei due nel ciclo dipende dall'ordine dei provider ed è irrilevante: entrambi restano pendenti). L'ordine relativo tra la risoluzione di `SiteContextConfigInitializer` e l'emissione del passo 6 è garantito perché la ripresa dopo `await` avviene sempre in un microtask successivo all'esecuzione sincrona del ciclo `for`.

4.3 Stato di `ongoingScopes$`



Riferimenti: `config-initializer.service.ts:L35-L37` (stato iniziale), `L170-L170` (emissione), `L77-L83` (`finishScopes`), `L42-L44` (`isStable`), `L131-L134` (guard). Se il guard `CONFIG_INITIALIZER_FORROOT_GUARD` manca, `isStable` è sempre vero indipendentemente da questo stato (`L43-L43`).

4.4 Config unificata da moduli lazy



Riferimenti: lazy-modules.service.ts:L78-L107, L47-L68; unified-injector.ts:L32-L35, L42-L68; configuration.service.ts:L57-L106.

---

5. Estratti di codice reale

Il cuore della costruzione della config — core-libs/core/src/config/config-tokens.ts:L10-L21

```ts
export function configFactory() {
  return deepMerge({}, inject(DefaultConfig), inject(RootConfig));
}

/**
 * Global Configuration, can be used to inject configuration to any part of the app
 */
@Injectable({
  providedIn: 'root',
  useFactory: configFactory,
})
export abstract class Config {}
```

Factory dei due livelli — core-libs/core/src/config/config-tokens.ts:L23-L47

```ts
export function defaultConfigFactory() {
  return deepMerge(
    {},
    ...(inject(DefaultConfigChunk, { optional: true }) ?? [])
  );
}
// ...
export const DefaultConfig = new InjectionToken('DefaultConfiguration', {
  providedIn: 'root',
  factory: defaultConfigFactory,
});

export function rootConfigFactory() {
  return deepMerge({}, ...(inject(ConfigChunk, { optional: true }) ?? []));
}
// ...
export const RootConfig = new InjectionToken('RootConfiguration', {
  providedIn: 'root',
  factory: rootConfigFactory,
});
```

Un helper provide — core-libs/core/src/config/config-providers.ts:L18-L27

```ts
export function provideConfig(
  config: Config = {},
  defaultConfig = false
): ValueProvider {
  return {
    provide: defaultConfig ? DefaultConfigChunk : ConfigChunk,
    useValue: config,
    multi: true,
  };
}
```

deepMerge — core-libs/core/src/config/utils/deep-merge.ts:L18-L49

```ts
export function deepMerge(
  target: Record<string, unknown> = {},
  ...sources: any[]
): any {
  if (!sources.length) {
    return target;
  }
  const source = sources.shift() || {};

  if (!isObject(source)) {
    return deepMerge(target, ...sources);
  }

  for (const key in source) {
    if (isRestricted(key)) {
      continue;
    }

    if (source[key] instanceof Date) {
      target[key] = source[key];
    } else if (isObject(source[key])) {
      if (!isObject(target[key])) {
        target[key] = {};
      }
      deepMerge(target[key] as Record<string, unknown>, source[key]);
    } else {
      target[key] = source[key];
    }
  }

  return deepMerge(target, ...sources);
}
```

Registrazione del motore degli initializer — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L41-L68

```ts
@NgModule({})
export class ConfigInitializerModule {
  static forRoot(): ModuleWithProviders<ConfigInitializerModule> {
    return {
      ngModule: ConfigInitializerModule,
      providers: [
        {
          provide: CONFIG_INITIALIZER_FORROOT_GUARD,
          useValue: true,
        },
        {
          provide: APP_INITIALIZER,
          multi: true,
          useFactory: configInitializerFactory,
          deps: [
            ConfigInitializerService,
            [new Optional(), CONFIG_INITIALIZER],
          ],
        },
        provideLocationInitializerFactory(() => {
          // Hold on the initial navigation until the Spartacus configuration is stable
          const configInitializer = inject(ConfigInitializerService);
          return () => lastValueFrom(configInitializer.getStable());
        }),
      ],
    };
  }
}
```

initialize — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L130-L175

```ts
  async initialize(initializers?: ConfigInitializer[]) {
    if (this.ongoingScopes$.value) {
      // guard for double initialization
      return;
    }

    const ongoingScopes: string[] = [];

    const asyncConfigs: Promise<void>[] = [];

    for (const initializer of initializers || []) {
      if (!initializer) {
        continue;
      }
      if (!initializer.scopes || !initializer.scopes.length) {
        throw new Error('CONFIG_INITIALIZER should provide scope!');
      }
```

```
      if (isDevMode() && !this.areReady(initializer.scopes, ongoingScopes)) {
        this.logger.warn(
          'More than one CONFIG_INITIALIZER is initializing the same config scope.'
        );
      }

      ongoingScopes.push(...initializer.scopes);

      asyncConfigs.push(
        (async () => {
          const initializerConfig = await initializer.configFactory();
          // contribute configuration to rootConfig
          deepMerge(
            this.rootConfig as Record<string, unknown>,
            initializerConfig
          );
          // contribute configuration to global config
          deepMerge(this.config as Record<string, unknown>, initializerConfig);
          this.finishScopes(initializer.scopes);
        })()
      );
    }
    this.ongoingScopes$.next(ongoingScopes);

    if (asyncConfigs.length) {
      await Promise.all(asyncConfigs);
    }
  }
```

getStable — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70

```ts
  getStable(...scopes: string[]): Observable<Config> {
    if (this.isStable) {
      return of(this.config);
    }
    return this.ongoingScopes$.pipe(
      filter(
        (ongoingScopes) =>
          !!ongoingScopes && this.areReady(scopes, ongoingScopes)
      ),
      take(1),
      map(() => this.config)
    );
  }
```

Initializer condizionale del site context — core-libs/core/src/site-context/site-context.module.ts:L30-L41

```ts
export function initSiteContextConfig(
  configInitializer: SiteContextConfigInitializer,
  config: SiteContextConfig
): ConfigInitializer | null {
  /**
   * Load config for `context` from backend only when there is no static config for `context.baseSite`
   */
  if (!config.context?.[BASE_SITE_CONTEXT_ID]) {
    return configInitializer;
  }
  return null;
```

Validatore in dev mode — core-libs/core/src/config/config-validator/config-validator.module.ts:L23-L37

```ts
export function configValidatorFactory(
  configInitializer: ConfigInitializerService,
  validators: ConfigValidator[]
): () => void {
  const logger = inject(LoggerService);
  return () => {
    if (isDevMode()) {
      configInitializer
        .getStable()
        .subscribe((config) =>
          validateConfig(config, validators || [], logger)
        );
    }
  };
}
```

Config unificata — core-libs/core/src/config/services/configuration.service.ts:L92-L106

```ts
  private emitUnifiedConfig(): void {
    const newConfig: Config = deepMerge(
      {},
      this.defaultConfig,
      this.ambientDefaultConfig,
      this.ambientConfig,
      this.rootConfig
    );
    (this.unifiedConfig$ as BehaviorSubject<Config>).next(newConfig);

    // compatibility mechanism, can be disabled with feature toggle
    if (!isFeatureEnabled(this.config, 'disableConfigUpdates')) {
      deepMerge(this.config as Record<string, unknown>, newConfig);
    }
  }
```

Module augmentation di una feature lib — feature-libs/cart/base/root/config/cart-config.ts:L30-L32

```ts
declare module '@spartacus/core' {
  interface Config extends CartConfig {}
}
```

6. Configurazione

Il sottosistema è il "contenitore" della configurazione; le chiavi che lo governano direttamente sono poche.

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`features.disableConfigUpdates`	`boolean?`	non impostata (= `false`, "disabled by default")	core-libs/core/src/features-config/config/features-config.ts:L101-L116	core-libs/core/src/config/services/configuration.service.ts:L103-L103 via `isFeatureEnabled` (core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L51)
`context.baseSite` (statico)	`string[]`	—	core-libs/core/src/site-context/config/site-context-config.ts:L15-L18	core-libs/core/src/site-context/site-context.module.ts:L36-L38 (se presente, `SiteContextConfigInitializer` non viene attivato)
`i18n.fallbackLang`	`string | false`	`undefined` (esplicito in `defaultI18nConfig`, core-libs/core/src/i18n/config/default-i18n-config.ts:L9-L14)	core-libs/core/src/i18n/config/i18n-config.ts:L19-L23	core-libs/core/src/i18n/i18n.module.ts:L30-L32 (se definito, `I18nConfigInitializer` non viene attivato)
`TestConfigModuleOptions.cookie`	`string`	— (demo: `'cxConfigE2E'`)	core-libs/core/src/config/test-config.module.ts:L65-L67	core-libs/core/src/config/test-config.module.ts:L89-L92, L45-L45
`provideConfig(config, defaultConfig)` 2° argomento	`boolean`	`false`	core-libs/core/src/config/config-providers.ts:L18-L21	core-libs/core/src/config/config-providers.ts:L23-L23
`provideConfigFactory(..., defaultConfig)` 3° argomento	`boolean`	`false`	core-libs/core/src/config/config-providers.ts:L37-L41	core-libs/core/src/config/config-providers.ts:L43-L43
`StringTemplate.resolve(..., encodeVariable)`	`boolean?`	`undefined` (= non codifica)	core-libs/core/src/config/utils/string-template.ts:L15-L19	core-libs/core/src/config/utils/string-template.ts:L25-L27

Priorità di merge, dal più debole al più forte:

1. `DefaultConfigChunk`, nell'ordine dei provider — core-libs/core/src/config/config-tokens.ts:L23-L28.
2. `DefaultConfigChunk` dei moduli lazy (`ambientDefaultConfig`), fusi nella `Config` globale se `disableConfigUpdates` non è attivo — core-libs/core/src/config/services/configuration.service.

ts:L93-L99.

3. ConfigChunk dei moduli lazy (ambientConfig) — stesso deepMerge, L93-L99.
4. ConfigChunk root, nell'ordine dei provider — core-libs/core/src/config/config-tokens.ts:L37-L39 — sempre sopra i default — L10-L12 — e sopra i chunk lazy perché rootConfig è l'ultimo argomento del deepMerge di emitUnifiedConfig — core-libs/core/src/config/services/configuration.service.ts:L93-L99.
5. Config prodotta dai CONFIG_INITIALIZER, fusa in place in RootConfig (e in Config) dopo la costruzione — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L158-L165 — quindi presente anche nel rootConfig usato da emitUnifiedConfig, dove sovrascrive i ConfigChunk root.

---

7. Commenti e note notevoli dal sorgente

- "General rule is, that all config provided in libraries should be provided as default config." — core-libs/core/src/config/config-tokens.ts:L58-L58. È la regola che dà senso all'intera separazione default/root.
- @deprecated since 2211.22 - should not be a public API su locationInitializedFactory — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L32-L39. Esiste un'altra funzione omonima, privata, in core-libs/core/src/routing/routing.module.ts:L66-L71: è quella effettivamente usata per LOCATION_INITIALIZED.
- // Hold on the initial navigation until the Spartacus configuration is stable — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L61-L61.
- @internal ... Not a part of a public API, used by APP_INITIALIZER to initialize all provided CONFIG_INITIALIZERS su initialize — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L129; e // guard for double initialization — L132-L132.
- "Should stay private in 1.x as forRoot() is used internally by ConfigInitializerModule — issue: #5279" su ConfigValidatorModule — core-libs/core/src/config/config-validator/config-validator.module.ts:L39-L44. Il modulo però è riesportato con export * — core-libs/core/src/config/config-validator/index.ts:L7-L8 — e la frase "used internally by ConfigInitializerModule" non corrisponde più al codice: lo importa BaseCoreModule — core-libs/core/src/base-core.module.ts:L32-L32.
- Refuso nel JSDoc di provideConfigValidator: "Use to probide config validation..." — core-libs/core/src/config/config-validator/config-validator.ts:L22-L26.
- "We need to use subscription to propagate changes to the config from the beginning. It will be possible to make it lazy, when we drop this compatibility feature in the future." — core-libs/core/src/config/services/configuration.service.ts:L51-L54.
- @deprecated - this feature will be moved into INJECTION_TOKEN in https://jira.tools.sap/browse/CXSPA-13592 su disableConfigUpdates — core-libs/core/src/features-config/config/features-config.ts:L101-L116.
- "CAUTION: DON'T USE IT IN PRODUCTION! IT HASN'T BEEN REVIEWED FOR SECURITY ISSUES." — core-libs/core/src/config/test-config.module.ts:L69-L73.
- // eslint-disable-next-line @nx/workspace-use-provide-default-config-factory -- deliberately providing a high priority FeatureConfig — core-libs/core/src/config/test-config.module.ts:L99-L102: il commento parla di "FeatureConfig" ma il provider è un ConfigChunk con l'intero oggetto letto dal cookie.
- // TODO 4.0: default to encodeVariable = true — core-libs/core/src/config/utils/string-template.ts:L24-L24, ancora presente nella 2612.

- `// When removing this feature toggle, switch to provideConfigInitializer` — core-libs/core/src/auth/user-auth/user-auth.module.ts:L109-L109.
- Lato Angular, `APP_INITIALIZER` (usato da `ConfigInitializerModule` e `ConfigValidatorModule`) è `@deprecated from v19.0.0, use provideAppInitializer instead` — angular:packages/core/src/application/application_init.ts:L38-L38.

8. Extension point

EXTENSION POINT	TIPO	DEFINIZIONE	EXPORT PUBBLICO
`Config` (augmentation)	classe astratta / interfaccia augmentable	core-libs/core/src/config/config-tokens.ts:L17-L21	core-libs/core/src/config/index.ts:L10-L16; core-libs/core/public_api.ts:L40-L40
`ConfigChunk`, `DefaultConfigChunk`	multi `InjectionToken<Config[]>`	core-libs/core/src/config/config-tokens.ts:L53-L53, L60-L62	core-libs/core/src/config/index.ts:L10-L16
`DefaultConfig`, `RootConfig`	`InjectionToken`	core-libs/core/src/config/config-tokens.ts:L32-L35, L44-L47	idem
`provideConfig`, `provideConfigFactory`, `provideDefaultConfig`, `provideDefaultConfigFactory`	funzioni provider	core-libs/core/src/config/config-providers.ts:L18-L79	core-libs/core/src/config/index.ts:L9-L9
`ConfigModule.withConfig` / `withConfigFactory` / `forRoot`	NgModule	core-libs/core/src/config/config.module.ts:L13-L67	core-libs/core/src/config/index.ts:L18-L18
`CONFIG_INITIALIZER`, `ConfigInitializer`, `provideConfigInitializer`, `provideConfigInitializerFactory`	token / interfaccia / provider	core-libs/core/src/config/config-initializer/config-initializer.ts:L15-L67	core-libs/core/src/config/config-initializer/index.ts:L7-L12 → core-libs/core/src/config/index.ts:L8-L8
`ConfigInitializerService.getStable`	servizio root (metodi `protected` sovrascrivibili)	core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70	core-libs/core/src/config/config-initializer/index.ts:L14-L14
`ConfigValidator`, `provideConfigValidator`, `ConfigValidatorToken`, `validateConfig`	tipo / provider / token / funzione	core-libs/core/src/config/config-validator/config-validator.ts:L11-L48	core-libs/core/src/config/config-validator/index.ts:L7-L7
`ConfigurationService.unifiedConfig$`	Observable	core-libs/core/src/config/services/configuration.service.ts:L21-L35	core-libs/core/src/config/index.ts:L19-L19
`TestConfigModule`, `TEST_CONFIG`, `TEST_CONFIG_COOKIE_NAME`	NgModule / token	core-libs/core/src/config/test-config.module.ts:L31-L106	core-libs/core/src/config/index.ts:L20-L26

EXTENSION POINT	TIPO	DEFINIZIONE	EXPORT PUBBLICO
`LOCATION_INITIALIZED_MULTI`, `provideLocationInitializerFactory`	token / provider	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L36-L51	core-libs/core/src/routing/index.ts:L11-L11
`deepMerge`, `isObject`, `StringTemplate`	util	core-libs/core/src/config/utils/deep-merge.ts:L7-L49, core-libs/core/src/config/utils/string-template.ts:L15-L31	core-libs/core/src/config/index.ts:L27-L28

Ricette pertinenti nella Parte III:

- Override di configurazione (`provideConfig`, ordine dei chunk, config unificata), override via DI: 50 — Ricette: configurazione, DI, interceptor, eventi.
- Endpoint custom e template OCC (`StringTemplate`, `backend.occ.endpoints`): 52 — Ricette: OCC, adapter, converter, endpoint, meta resolver.
- Config di una feature lib lazy (`provideDefaultConfig` nel root module, chunk nei moduli lazy) e traduzioni: 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni.
- Uso di simboli pubblici vs interni (es. `CONFIG_INITIALIZER_FORROOT_GUARD`, `initialize` `@internal`): 54 — API pubblica vs interna.

Schema minimo di un `CONFIG_INITIALIZER` custom, costruito **solo** con simboli esportati (`ConfigInitializer`, `provideConfigInitializer` da core-libs/core/src/config/config-initializer/index.ts:L7-L12). Il codice è illustrativo, non presente nel repo:

```ts
// Esempio illustrativo (non nel repo)
@Injectable({ providedIn: 'root' })
export class MyConfigInitializer implements ConfigInitializer {
  readonly scopes = ['myFeature.endpoint'];
  readonly configFactory = () =>
    lastValueFrom(this.http.get<{ url: string }>('/my-config')).then((r) => ({
      myFeature: { endpoint: r.url },
    }));
  constructor(protected http: HttpClient) {}
}

// nei provider dell'app
provideConfigInitializer(MyConfigInitializer);
```

Vincoli che derivano dal codice: `scopes` non vuoto (altrimenti errore, config-initializer.service.ts:L144-L146); il chunk prodotto vince sui default e sui `ConfigChunk` statici (merge in place dopo la costruzione, L158-L165); un rifiuto della Promise blocca il bootstrap (L172-L174). DA VERIFICARE: l'uso di `HttpClient` dentro una `configFactory` passa per gli interceptor Spartacus già registrati; non è stato verificato quali di essi dipendano a loro volta da config non ancora stabile.

9. Trappole, casi limite, comportamenti controintuitivi

1. **La `Config` è mutabile e viene mutata.** `deepMerge` muta il primo argomento; `ConfigInitializerService` muta `rootConfig` e `config` — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L160-L165 — e `ConfigurationService` muta `config` quando arriva un modulo lazy — core-libs/core/src/config/services/configuration.service.ts:L103-L105. Chi ha letto un valore durante il costruttore può trovarsi con un dato vecchio: per questo esiste `getStable`.
2. **Gli array si sostituiscono, non si concatenano**, e vengono assegnati per riferimento — core-libs/core/src/config/utils/deep-merge.ts:L43-L45; core-libs/core/src/config/utils/deep-merge.spec.ts:L86-L86. Un `provideConfig({ context: { language: ['de'] } })` rimpiazza l'intero array di default. Mutare un array letto dalla `Config` muta anche il chunk originale.
3. **L'ordine di import conta solo dentro lo stesso token.** `ConfigChunk` batte `DefaultConfigChunk` sempre — core-libs/core/src/config/config-tokens.ts:L10-L12; tra due `ConfigChunk` vince l'ultimo fuso — L37-L39. Da qui "Should be imported after other config modules" su `TestConfigModule` — projects/storefrontapp/src/app/private/private.providers.ts:L54-L54.
4. **Un `provideConfig` in un modulo lazy non batte il `provideConfig` dell'app root**: nella config unificata `rootConfig` è applicata per ultima — core-libs/core/src/config/services/configuration.service.ts:L93-L99.
5. **Senza `ConfigInitializerModule.forRoot()` gli initializer vengono ignorati in silenzio**: il guard manca, `isStable` è sempre vero e `getStable` emette subito — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L42-L44 — e nessun `APP_INITIALIZER` chiama `initialize`.
6. **Dipendenze tra initializer**: funzionano solo grazie all'emissione differita di `ongoingScopes$` — config-initializer.service.ts:L62-L66, L154-L170. Una dipendenza circolare (A attende lo scope di B e viceversa) non ha timeout: `DA VERIFICARE` — è un'inferenza dal codice (L156-L174), nessun test letto la copre; il risultato atteso è un bootstrap che non termina.
7. `finishScopes` **con** `splice(indexOf(scope), 1)`: se lo scope non fosse nell'array, `indexOf` = -1 rimuoverebbe l'ultimo elemento — config-initializer.service.ts:L78-L81. Nel flusso normale non accade perché gli scope sono sempre pushati in `initialize` (L154-L154); diventa rilevante solo se si sovrascrive il servizio.
8. **Scope duplicati**: due initializer con lo stesso scope producono solo un warning in dev mode — config-initializer.service.ts:L148-L152; l'array contiene lo scope due volte e ciascun `finishScopes` ne rimuove una occorrenza (L80-L80), quindi lo scope resta "in corso" finché entrambi non hanno finito.
9. **Errore sincrono dentro un metodo async**: un initializer senza scope fa lanciare `initialize`, che essendo `async` restituisce una Promise rifiutata → bootstrap fallito — config-initializer.service.ts:L130-L130, L144-L146.
10. **I validator non proteggono la produzione**: girano solo con `isDevMode()` e loggano soltanto — core-libs/core/src/config/config-validator/config-validator.module.ts:L28-L35; core-libs/core/src/config/config-validator/config-validator.ts:L42-L46.
11. **Non determinismo con i moduli lazy**: la `Config` globale cambia quando arriva un modulo lazy; `CmsComponentsService` copia `cmsComponents` dalla config stabile proprio per questo — core-libs/storefront/cms-structure/services/cms-components.service.ts:L61-L69.
12. `ConfigurationService` **esiste solo se c'è** `ConfigModule` — core-libs/core/src/config/config.module.ts:L15-L19. In un setup minimale (es. test) che non importa `ConfigModule` i chunk dei moduli lazy non arrivano mai nella config globale. L'assenza di altri punti di iniezione è verificata via grep su `core-libs`, `feature-libs`, `integration-libs` e `projects` (spec escluse): `ConfigurationService` è iniettato solo nel costruttore di `ConfigModule` — core-libs/core/src/config/config.module.ts:L17-L17.

13. **Il sito non trovato rompe il bootstrap**: `SiteContextConfigInitializer` lancia un errore se l'URL non matcha alcun base site — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L61-L70 — ma è attivo solo se manca `context.baseSite` statico — core-libs/core/src/site-context/site-context.module.ts:L36-L40.
14. `ConfigModule.forRoot()` **non è un vero forRoot**: identico a `withConfig`, nessun guard di unicità — core-libs/core/src/config/config.module.ts:L28-L34, L60-L66.
15. **Doppio decode e RegExp non escapata nel cookie e2e**: `getCookie` fa `decodeURIComponent` — core-libs/core/src/config/utils/get-cookie.ts:L11-L11 — e `parseConfigJSON` lo rifà — core-libs/core/src/config/test-config.module.ts:L59-L59; il nome del cookie entra nella RegExp senza escape — get-cookie.ts:L8-L8.
16. `StringTemplate.resolve` **non codifica di default** (`TODO 4.0` mai applicato) — core-libs/core/src/config/utils/string-template.ts:L24-L28; chi la usa in codice custom deve passare `true` come fanno i chiamanti di libreria (core-libs/core/src/occ/services/occ-endpoints.service.ts:L122-L122).
    DA VERIFICARE: il valore è usato come stringa di sostituzione di `String.prototype.replace`, quindi sequenze come `$&` verrebbero interpretate; è semantica JavaScript non coperta da test letti.
17. `isObject` **non restituisce** `boolean`: per `null`, `0`, `''` restituisce il valore falsy stesso — core-libs/core/src/config/utils/deep-merge.ts:L7-L9. Irrilevante nei `if`, ma sorprendente se usato in un confronto stretto.

---

10. Come fare debug

- **Vedere la config finale**: iniettare `Config` (o usare `getStable()` per attendere la stabilità, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70) e loggare l'oggetto. Per capire *da dove* arriva un valore, iniettare separatamente `DefaultConfig` e `RootConfig` (entrambi pubblici, core-libs/core/src/config/index.ts:L10-L16) e confrontarli.
- **Elencare i chunk**: `inject(ConfigChunk, { optional: true })` e `inject(DefaultConfigChunk, { optional: true })` restituiscono gli array nell'ordine di merge (stesso pattern di core-libs/core/src/config/config-tokens.ts:L26-L26, L38-L38).
- **Bootstrap bloccato** (pagina bianca, nessuna navigazione): un initializer non si risolve. Sottoscriversi a `getStable()` con log, o sovrascrivere `ConfigInitializerService` estendendolo e loggando in `finishScopes` (metodo `protected`, config-initializer.service.ts:L77-L83) per vedere quale scope resta aperto. Ricordare che `LOCATION_INITIALIZED` attende `getStable()` — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64.
- **Bootstrap fallito con errore**: cercare in console "CONFIG_INITIALIZER should provide scope!" (config-initializer.service.ts:L145-L145) o "Cannot get base site config!" (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L66-L68).
- **Warning in dev mode**: "More than one CONFIG_INITIALIZER is initializing the same config scope." (config-initializer.service.ts:L148-L152) e i messaggi dei validator, emessi con `LoggerService.warn` (core-libs/core/src/config/config-validator/config-validator.ts:L42-L46). In un build di produzione non compaiono (angular:packages/core/src/util/is_dev_mode.ts:L20-L22).
- **Config che cambia dopo un lazy load**: sottoscriversi a `ConfigurationService.unifiedConfig$` — core-libs/core/src/config/services/configuration.service.ts:L30-L30 — che emette a ogni modulo lazy con chunk. Per isolare il problema si può attivare `features.disableConfigUpdates` (deprecato) e verificare se la `Config` globale smette di cambiare — configuration.service.ts:L103-L105.
- **Test e2e**: impostare il cookie `cxConfigE2E` con JSON url-encoded; l'app demo lo legge tramite `TestConfigModule` — projects/storefrontapp/src/app/private/private.providers.ts:L52-L56, core-libs/core/src/config/test-config.module.ts:L44-L54.

- **Spec di riferimento** per comportamenti precisi: `core-libs/core/src/config/utils/deep-merge.spec.ts`, `core-libs/core/src/config/config-initializer/config-initializer.service.spec.ts`, `core-libs/core/src/config/services/configuration.service.spec.ts`.

---

11. Domande di autoverifica

1. **Una libreria fornisce** `provideDefaultConfig({ a: 1 })` **e l'app, importata *prima*, fornisce** `provideConfig({ a: 2 })`**. Quale valore vede** `Config.a`**?** 2. I due token sono fusi separatamente e `RootConfig` è applicato dopo `DefaultConfig` indipendentemente dall'ordine di import — `core-libs/core/src/config/config-tokens.ts:L10-L12`.
2. **Default** `context.language: ['en','de']`**, app** `provideConfig({ context: { language: ['it'] } })`**. Risultato?** `['it']`: gli array sovrascrivono — `core-libs/core/src/config/utils/deep-merge.ts:L43-L45`; `deep-merge.spec.ts:L86-L86`.
3. **Perché iniettare** `SiteContextConfig` **e** `Config` **restituisce lo stesso oggetto?** Perché `SiteContextConfig` è dichiarata `useExisting: Config` — `core-libs/core/src/site-context/config/site-context-config.ts:L10-L19`.
4. **Cosa succede se un** `CONFIG_INITIALIZER` **ha** `scopes: []`**?** `initialize` lancia `Error('CONFIG_INITIALIZER should provide scope!')`, la Promise dell'`APP_INITIALIZER` è rifiutata e il bootstrap fallisce — `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L144-L146`; `angular:packages/core/src/application/application_init.ts:L263-L269`.
5. `getStable('test')` **con** `'testA'` **in corso: emette subito?** Sì, se non ci sono altri scope sovrapposti: `'test'` e `'testA'` non si sovrappongono perché il carattere dopo il prefisso non è `.` — `config-initializer.service.ts:L105-L122`.
6. **Perché** `I18nConfigInitializer` **può chiamare** `getStable('context.language')` **dentro la propria** `configFactory` **senza ricevere subito una config incompleta?** Perché in quel momento `ongoingScopes$` è ancora `undefined` (l'elenco è emesso solo dopo il ciclo, `config-initializer.service.ts:L170-L170`), `isStable` è falso e il filtro `!!ongoingScopes` attende; poi attende la chiusura dello scope `context` — `L58-L70`; `core-libs/core/src/i18n/config/i18n-config-initializer.ts:L25-L25`.
7. **In produzione un** `ConfigValidator` **che restituisce una stringa blocca l'app?** No: i validator girano solo in dev mode e loggano un warning — `core-libs/core/src/config/config-validator/config-validator.module.ts:L28-L35`; `core-libs/core/src/config/config-validator/config-validator.ts:L42-L46`.
8. **Un modulo lazy fornisce** `provideConfig({ x: 'lazy' })`**, l'app root** `provideConfig({ x: 'root' })`**. Dopo il lazy load, quanto vale** `Config.x`**?** `'root'`: in `emitUnifiedConfig` `rootConfig` è l'ultima sorgente — `core-libs/core/src/config/services/configuration.service.ts:L93-L99` — e il risultato è fuso nella `Config` globale — `L103-L105`.
9. **Come fa Spartacus a impedire che il router riconosca una route prima che la config di routing caricata dal backend sia disponibile?** `ConfigInitializerModule` registra un `LOCATION_INITIALIZED_MULTI` che attende `getStable()` — `core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64`; `RoutingModule` lo aggrega in `LOCATION_INITIALIZED` — `core-libs/core/src/routing/routing.module.ts:L66-L71` — e con `initialNavigation: 'enabledBlocking'` Angular attende quella Promise prima della navigazione iniziale — `angular:packages/router/src/provide_router.ts:L441-L455`.
10. **Perché** `ConfigInitializerService` **fonde il chunk asincrono sia in** `rootConfig` **sia in** `config`**?** `config` è l'oggetto letto dall'app; `rootConfig` è riapplicato per ultimo nella config unificata dei moduli lazy (`configuration.service.ts:L93-L99`), quindi il valore asincrono resta prioritario anche dopo un lazy load — `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L158-L165`.

TAVOLA 11.1 **4.1 Costruzione della Config e priorità dei livelli**

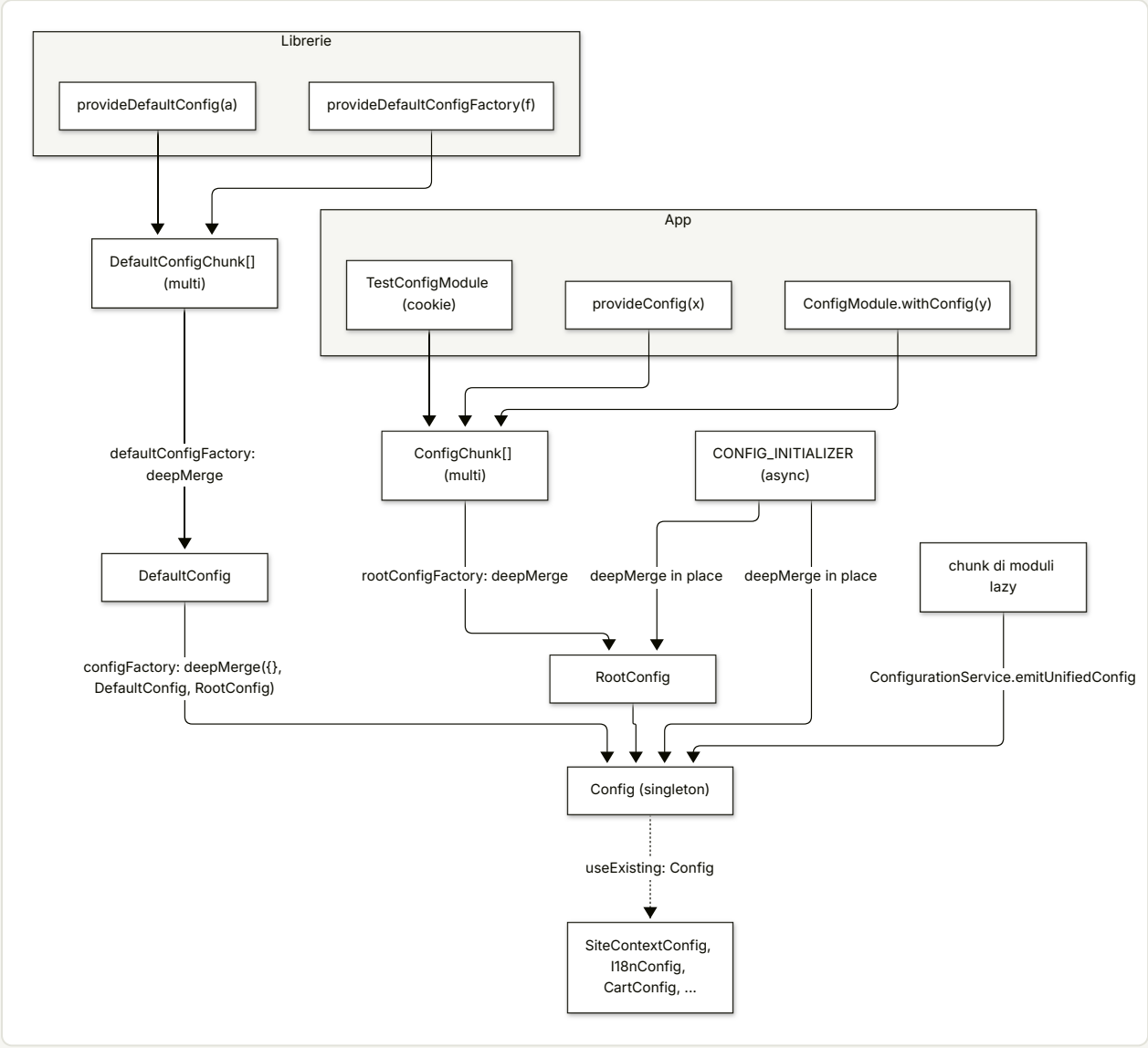
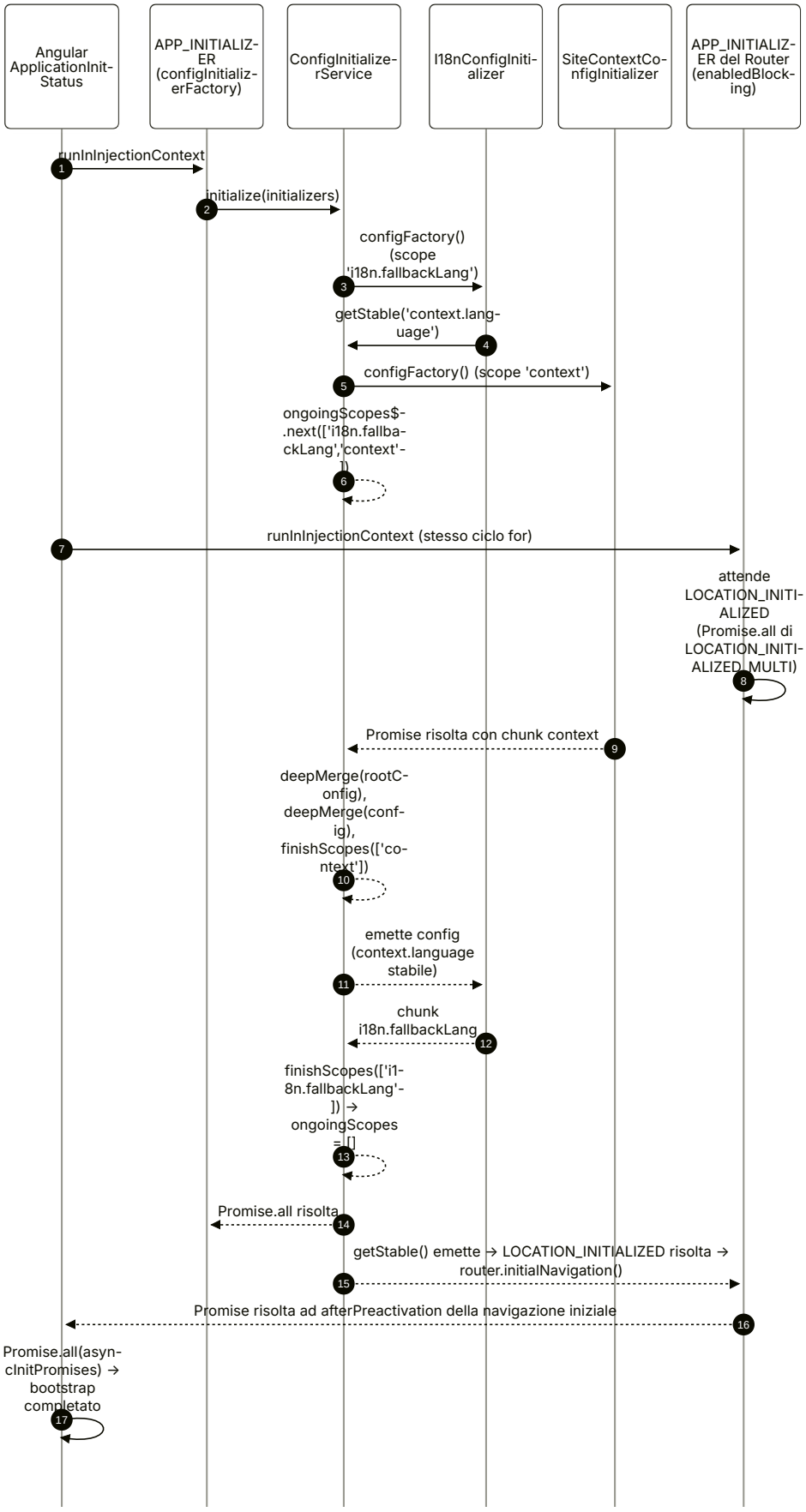


TAVOLA 11.2 **4.2 Bootstrap con config asincrona**



12

Feature module e lazy loading in Spartacus

Il lazy loading guidato dal CMS e non dalle route: `featureModules`, chunk, facade.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 01 — Dependency injection, 03 — Lazy loading in Angular, 11 — Configurazione.

1. A cosa serve e perché esiste

Angular offre un solo meccanismo "nativo" di lazy loading guidato dal router: `loadChildren` / `loadComponent` su una route (vedi capitolo 03). In uno storefront Spartacus però le pagine non sono definite da route statiche dell'applicazione ma dal CMS: la stessa route `cart` mostra i componenti che il backend dichiara nella pagina, e un componente come `MiniCartComponent` può comparire in ogni pagina, nello header. Il codice che serve a renderizzarlo (componenti, store NgRx, adapter OCC) va caricato **quando il CMS chiede quel tipo di componente**, non quando il router entra in una certa route.

Per questo Spartacus implementa un proprio sistema di lazy loading, indipendente dal router, i cui pezzi vivono quasi tutti in `core-libs/core/src/lazy-loading/` e sono esportati dal public API di `@spartacus/core` (core-libs/core/public_api.ts:L24-L24, core-libs/core/src/lazy-loading/index.ts:L7-L12):

- una **configurazione** dichiarativa, `featureModules`, che associa un nome di feature a un `import()` dinamico, a delle dipendenze e ai tipi di componente CMS che la feature implementa (core-libs/core/src/cms/config/cms-config.ts:L109-L134);
- un **motore di caricamento** a due livelli: `FeatureModulesService` (risolve per nome, alias, dipendenze, cache) e `LazyModulesService` (import, creazione dell'`NgModuleRef`, initializer, evento) (core-libs/core/src/lazy-loading/feature-modules.service.ts:L13-L124, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L38-L246);
- un **hook di inizializzazione** per moduli lazy, `MODULE_INITIALIZER`, che fa per un modulo lazy ciò che `APP_INITIALIZER` fa per l'app (core-libs/core/src/lazy-loading/tokens.ts:L9-L20);
- un **injector "unificato"**, `UnifiedInjector`, che permette ai servizi del root di vedere provider (multi provider, config chunk, guard) registrati dentro moduli caricati più tardi (core-libs/core/src/lazy-loading/unified-injector.ts:L20-L94);
- le **proxy facade** (`facadeFactory`), che consentono di iniettare dal root una facade astratta la cui implementazione vive nel modulo lazy, caricandolo alla prima chiamata (core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L11-L24);
- lato storefront, `CmsFeaturesService`, che collega i tipi di componente CMS alle feature e fornisce mapping e `NgModuleRef` al motore di rendering (core-libs/storefront/cms-structure/services/cms-features.service.ts:L27-L206).

Il risultato è il pattern "root module eager + modulo lazy" usato da tutte le feature lib: il **root module** (es. `CartBaseRootModule`) è piccolo, viene importato eager e dichiara *quali* componenti CMS la feature copre (feature-libs/cart/base/root/cart-base-root.module.ts:L77-L81); le facade astratte vivono nello stesso entry point `root` come classi `providedIn: 'root'` con factory `facadeFactory` e non dipendono dall'import del modulo (feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L16); l'**app** fornisce *dove* trovare il codice (`module: () => import(...)`); il codice pesante viene scaricato solo quando serve (feature-libs/cart/base/root/cart-base-root.module.ts:L26-L83, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L55).

> Nota: nel focus dell'outline compariva un ipotetico `provideFeature`. Non esiste nel codice core (grep vuoto, dossier Q1): le feature si dichiarano solo tramite la config `featureModules` con `provideConfig` / `provideDefaultConfig(Factory)`.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/lazy-loading/index.ts	Barrel: riesporta eventi, LazyModulesService, token, UnifiedInjector, FeatureModulesService, facade-factory. NON esporta LazyLoadingModule (:L7-L12).
core-libs/core/src/lazy-loading/tokens.ts	Definisce MODULE_INITIALIZER (:L19-L20).
core-libs/core/src/lazy-loading/lazy-loading.module.ts	LazyLoadingModule.forRoot(): APP_INITIALIZER che esegue i MODULE_INITIALIZER registrati eager (:L18-L53).
core-libs/core/src/lazy-loading/lazy-modules.service.ts	LazyModulesService: import dinamico, creazione NgModuleRef, scelta del parent injector, initializer, ModuleInitializedEvent, stream modules$ (:L38-L246).
core-libs/core/src/lazy-loading/feature-modules.service.ts	FeatureModulesService: risoluzione per nome da CmsConfig.featureModules, alias, dipendenze, cache (:L13-L124).
core-libs/core/src/lazy-loading/unified-injector.ts	UnifiedInjector: stream degli injector root + lazy, get / getMulti (:L20-L94).
core-libs/core/src/lazy-loading/events/index.ts	Barrel eventi (:L7-L7).
core-libs/core/src/lazy-loading/events/module-initialized-event.ts	ModuleInitializedEvent (feature?, moduleRef) (:L16-L32).
core-libs/core/src/lazy-loading/facade-factory/index.ts	Barrel facade factory (:L7-L9).
core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts	FacadeDescriptor e i tipi che vincolano metodi/proprietà (:L10-L57).
core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts	Funzione facadeFactory() (:L20-L24).
core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts	FacadeFactoryService: costruzione del proxy (:L29-L141).
core-libs/core/src/util/combined-injector.ts	(contorno) CombinedInjector, parent di moduli con più dipendenze; non esportato (:L27-L72).
core-libs/core/src/cms/config/cms-config.ts	(contorno) FeatureModuleConfig, CmsConfig.featureModules (:L109-L138).
core-libs/storefront/cms-structure/services/cms-features.service.ts	(contorno) CmsFeaturesService: componentType → feature → mapping / NgModuleRef (:L27-L206).
core-libs/storefront/cms-structure/services/cms-components.service.ts	(contorno) consumo di CmsFeaturesService: determineMappings, getModule (:L55-L180).
feature-libs/cart/base/root/cart-base-root.module.ts	(esempio) root module eager del carrello: featureModules con soli cmsComponents + alias (:L26-L83).
feature-libs/cart/base/root/facade/active-cart.facade.ts	(esempio) facade astratta con provider facadeFactory (:L13-L46).
projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts	(esempio) l'app fornisce i module: () => import(...) (:L23-L55).
projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts	(esempio) modulo lazy effettivamente caricato per cartBase (:L14-L25).

3. Come funziona passo per passo

3.1 La configurazione `featureModules`

`FeatureModuleConfig` ha tre chiavi, tutte opzionali (core-libs/core/src/cms/config/cms-config.ts:L109-L122):

- `module?: () => Promise<any>` — l'import dinamico del modulo lazy;
- `dependencies?: ((() => Promise<any>) | string)[]` — dipendenze, come funzioni di import oppure nomi di altre feature;
- `cmsComponents?: string[]` — i tipi di componente CMS coperti dalla feature.

`CmsConfig.featureModules` è una mappa `featureName → FeatureModuleConfig | string`, dove il valore stringa è un **alias** verso un'altra feature (core-libs/core/src/cms/config/cms-config.ts:L124-L130). `CmsConfig` è un token astratto che punta a `Config` (`useExisting: Config`) ed è fuso in `Config` per module augmentation (core-libs/core/src/cms/config/cms-config.ts:L124-L138).

Il punto chiave è che una feature viene di solito dichiarata **da due parti**: la libreria (root module) fornisce `cmsComponents`, `dependencies` e gli alias; l'app fornisce `module`. I due contributi vanno su token diversi: la lib registra `cmsComponents`/`dependencies`/alias come `DefaultConfigChunk` (`provideDefaultConfigFactory`, core-libs/core/src/config/config-providers.ts:L69-L79), l'app registra `module` come `ConfigChunk` (`provideConfig`, core-libs/core/src/config/config-providers.ts:L18-L27). `DefaultConfig` è il `deepMerge` dei `DefaultConfigChunk` (core-libs/core/src/config/config-tokens.ts:L23-L28), `RootConfig` quello dei `ConfigChunk` (core-libs/core/src/config/config-tokens.ts:L37-L39), e il `Config` globale (a cui punta `CmsConfig`) è `deepMerge({}, DefaultConfig, RootConfig)` (core-libs/core/src/config/config-tokens.ts:L10-L12): per questo la stessa voce `featureModules.cartBase` risulta completa. Esempio carrello:

- `defaultCartComponentsConfig` nel root module dichiara per `cartBase`, `miniCart`, `addToCart` **solo** `cmsComponents` (feature-libs/cart/base/root/cart-base-root.module.ts:L26-L44) e l'alias `cartBaseCore → cartBase` (feature-libs/cart/base/root/cart-base-root.module.ts:L45-L53); è registrata con `provideDefaultConfigFactory` (feature-libs/cart/base/root/cart-base-root.module.ts:L77-L81);
- `CartBaseFeatureModule` dell'app demo importa `CartBaseRootModule` e con `provideConfig` aggiunge i `module: () => import(...)` (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L55). `miniCart` e `addToCart` puntano a entry point secondari diversi, quindi a chunk separati da `cartBase` (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L36-L55).

I nomi delle feature sono costanti esportate dal root entry point: `CART_BASE_FEATURE = 'cartBase'`, `CART_BASE_CORE_FEATURE = 'cartBaseCore'`, `MINI_CART_FEATURE = 'miniCart'`, `ADD_TO_CART_FEATURE = 'addToCart'` (feature-libs/cart/base/root/feature-name.ts:L7-L11). Per la configurazione in generale vedi capitolo 11.

3.2 Bootstrap: `LazyLoadingModule` e `MODULE_INITIALIZER` eager

1. `BaseCoreModule` importa `LazyLoadingModule.forRoot()` insieme agli altri `forRoot()` del core (core-libs/core/src/base-core.module.ts:L26-L46). È l'unico punto d'uso del modulo, che non è nel public API (core-libs/core/src/base-core.module.ts:L18-L18, core-libs/core/src/lazy-loading/index.ts:L7-L12).
2. `forRoot()` registra un `APP_INITIALIZER` multi con factory `moduleInitializersFactory` e `deps: [LazyModulesService, [new Optional(), MODULE_INITIALIZER]]` (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L38-L53).

3. Effetto collaterale importante: poiché la factory dipende da `LazyModulesService`, il servizio viene istanziato durante il bootstrap e il suo costruttore connette subito `modules$` agli eventi (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L47-L47, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L60-L68). Così nessun `ModuleInitializedEvent` successivo va perso.
4. La factory usa `inject(LoggerService)` e restituisce una funzione che esegue i `MODULE_INITIALIZER` del root injector (cioè quelli forniti da moduli importati eager) tramite `runModuleInitializerFunctions` e fa `Promise.all` sulle promise ottenute, loggando e rilanciando in caso di reject (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L18-L36).

È il "fallback" documentato nel JSDoc del token: se il modulo che fornisce `MODULE_INITIALIZER` è caricato eager, le funzioni girano all'init dell'app come un `APP_INITIALIZER` (core-libs/core/src/lazy-loading/tokens.ts:L9-L18). Per l'ordine complessivo degli initializer vedi capitolo 10.

3.3 `FeatureModulesService.resolveFeature(name)`

`FeatureModulesService` è `providedIn: 'root'`, dipende da `CmsConfig` e `LazyModulesService` e tiene una `Map<string, Observable<NgModuleRef>>` di resolver "eseguiti una volta e messi in cache" (core-libs/core/src/lazy-loading/feature-modules.service.ts:L13-L26).

1. **Alias.** `resolveFeature` risolve per prima cosa l'alias: `resolveFeatureAlias` segue con un `while` la catena di valori stringa fino a una config oggetto (core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L43, core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100). Quindi `resolveFeature('cartBaseCore')` diventa `resolveFeature('cartBase')`.
2. **Lazy per costruzione.** Il resto è in un `defer`: nulla accade finché qualcuno non si sottoscrive (core-libs/core/src/lazy-loading/feature-modules.service.ts:L45-L72).
3. **Controllo configurazione.** Se il resolver non è in cache e la feature non è configurata, restituisce `throwError` con `No module defined for Feature Module <name>` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L46-L52). "Configurata" significa: la config (dopo l'alias) ha `module` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35, core-libs/core/src/lazy-loading/feature-modules.service.ts:L80-L86).
4. **Resolver.** Altrimenti crea e registra in cache la pipeline: `resolveDependencies(dependencies)` → `switchMap` → `lazyModules.resolveModuleInstance(module, featureName, deps)` → `shareReplay()` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L54-L71).
5. **Dipendenze.** `resolveDependencies` usa `forkJoin` (in parallelo): una stringa è una feature e richiama ricorsivamente `resolveFeature(dep)`; una funzione è un modulo dipendenza e passa per `resolveDependencyModuleInstance`; senza dipendenze emette `of(undefined)` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L108-L123).

Esempio reale di dipendenza tra feature: il root module del checkout dichiara `dependencies: [CART_BASE_FEATURE]` e l'alias `checkoutCore → checkout` (feature-libs/checkout/base/root/checkout-root.module.ts:L37-L49); lo stesso schema è usato da order (feature-libs/order/root/order-root.module.ts:L71-L76).

3.4 `LazyModulesService`: dal `import()` all'`NgModuleRef`

`LazyModulesService` è `providedIn: 'root'`, implementa `OnDestroy` e dipende (costruttore) da `Compiler`, dal root `Injector` e da `EventService`, e via `inject()` da `LoggerService` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L38-L64, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L42-L42).

Scelta del parent injector in `resolveModuleInstance(moduleFunc, feature?, dependencyModuleRefs = [])` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L94):

- nessuna dipendenza → root injector;
- una dipendenza → l'injector di quella dipendenza (`dependencyModuleRefs[0].injector`);
- più dipendenze → `new CombinedInjector(root, [injector delle dipendenze])`.

Conseguenza: il checkout (unica dipendenza `cartBase`) ha come parent **direttamente** l'injector del modulo `cartBase` e quindi ne vede i provider (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L87-L88).

Pipeline (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L107):

1. `resolveModuleFactory(moduleFunc)`: esegue l'`import()`; se il risultato è già una `NgModuleFactory` la usa, altrimenti chiama `compiler.compileModuleAsync(module)`; emette `[factory, module]` rischedulato su `queueScheduler` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L236). In Angular 21 `compileModuleAsync` è soltanto `Promise.resolve(new NgModuleFactoryR3(moduleType))` e `Compiler` è deprecato (angular:packages/core/src/linker/compiler.ts:L48-L66).
2. `moduleFactory.create(parentInjector)`: l'implementazione Ivy crea `new NgModuleRef(moduleType, parentInjector, [])` (angular:packages/core/src/render3/ng_module_ref.ts:L127-L134), che costruisce un `R3Injector` figlio del parent con scope `'environment'`, risolve gli injector initializer e istanzia la classe modulo (angular:packages/core/src/render3/ng_module_ref.ts:L67-L108). L'`injector` dell'`NgModuleRef` è quindi un `EnvironmentInjector` (angular:packages/core/src/render3/ng_module_ref.ts:L110-L112).
3. `runModuleInitializersForModule(moduleRef)` (in `concatMap`): esegue i `MODULE_INITIALIZER` **del modulo** (vedi 3.5).
4. `events.dispatch(createFrom(ModuleInitializedEvent, { feature, moduleRef }))` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L99-L106). `createFrom` è `Object.assign(new type(), data)` (core-libs/core/src/util/create-from.ts:L15-L17).

Il JSDoc avverte: "New module instance will be created with each call" (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L70-L77). La deduplica per feature è responsabilità di `FeatureModulesService` (3.3).

Moduli dipendenza (dipendenze di tipo funzione): `resolveDependencyModuleInstance` istanzia il modulo **una sola volta per classe**, sempre con il root injector come parent, e lo conserva nella `Map` privata `dependencyModules` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L57-L57, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L110-L129). L'evento emesso per loro non ha `feature` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L130-L138), ed è proprio così che, secondo il JSDoc, si distinguono feature e dipendenze (core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L21-L27).

Distruzione: `ngOnDestroy` chiude la sottoscrizione agli eventi e distrugge i moduli dipendenza, non i moduli feature (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L238-L245).

3.5 `MODULE_INITIALIZER` nei moduli lazy

`MODULE_INITIALIZER` è un `InjectionToken<(() => any)[]>` usato come multi provider (core-libs/core/src/lazy-loading/tokens.ts:L9-L20).

1. `runModuleInitializersForModule` lo legge dall'injector del modulo con `{ self: true }` e default `[]`: solo gli initializer forniti in quel modulo (o nei moduli che importa, che confluiscono nello stesso `R3Injector`), non quelli del root (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157).
2. `runModuleInitializerFunctions` esegue tutte le funzioni sincronamente e in ordine, raccogliendo solo i risultati "promise-like" (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L187-L200). "Promise-like" significa oggetto truthy con `then` funzione (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L213-L215): **un Observable restituito non viene atteso**.

3. Un'eccezione sincrona viene loggata ("init function throwed an error") e rilanciata (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L201-L207).
4. Se ci sono promise si attende `Promise.all`; su reject si logga e l'errore si propaga nello stream, altrimenti si emette `moduleRef`. Senza promise si emette subito `of(moduleRef)` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L158-L174).

Usi reali:

- `CartPersistenceModule` registra un initializer asincrono che, a config `context` stabile, chiama `initSync()` della persistenza del carrello; usa `lastValueFrom` proprio per trasformare l'Observable in promise (feature-libs/cart/base/core/cart-persistence.module.ts:L19-L31, feature-libs/cart/base/core/cart-persistence.module.ts:L58-L66). È importato da `CartBaseCoreModule` (feature-libs/cart/base/core/cart-base-core.module.ts:L22-L40).
- `AsmCoreModule` registra un initializer sincrono (feature-libs/asm/core/asm-core.module.ts:L15-L31).
- `OutletModule.forRoot()` registra gli outlet con `APP_INITIALIZER`, `OutletModule.forChild()` con `MODULE_INITIALIZER` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88); `OrderDetailsModule` registra un outlet via `MODULE_INITIALIZER` (feature-libs/order/components/order-details/order-details.module.ts:L183-L187). Vedi capitolo 17.

3.6 `ModuleInitializedEvent`, `modules$` e `UnifiedInjector`

`ModuleInitializedEvent` estende `CxEvent`, ha `type = 'ModuleInitializedEvent'`, `feature?` e `moduleRef` (core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L32). `EventService.dispatch` lo invia su un `Subject` semplice (non replay) (core-libs/core/src/event/event.service.ts:L105-L127); per gli eventi in generale vedi capitolo 20.

`LazyModulesService.modules$` è un `connectable` degli eventi, mappati a `moduleRef`, con connector `ReplaySubject` illimitato e `resetOnDisconnect: false`, connesso nel costruttore (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L44-L68): chiunque si sottoscriva in seguito riceve **tutti** i moduli già inizializzati.

`UnifiedInjector` (`providedIn: 'root'`) costruisce `injectors$` = root injector (`startWith`) seguito dall'injector di ogni modulo emesso da `modules$` (core-libs/core/src/lazy-loading/unified-injector.ts:L20-L40).

- `get(token, notFoundValue?)` interroga il root injector normalmente (indice 0) e ogni injector lazy con `{ self: true }`; senza `notFoundValue` usa un sentinel `NOT_FOUND_SYMBOL` e filtra i "non trovati" (core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68). Con `notFoundValue` emette esattamente una volta per injector, altrimenti solo quando trova nuove istanze (core-libs/core/src/lazy-loading/unified-injector.ts:L42-L51). L'opzione `self` di Angular limita la ricerca all'injector corrente (angular:packages/core/src/di/interface/injector.ts:L72-L74).
- `getMulti(token)` chiama `get(token, [])`, lancia se trova un valore non-array ("Multi-providers mixed with single providers"), scarta gli array vuoti e accumula con `scan` le istanze di tutti gli injector (core-libs/core/src/lazy-loading/unified-injector.ts:L75-L93).

Consumatori nel core e nello storefront:

CONSUMATORE	USO	RIFERIMENTO
`ConfigurationService`	`get(ConfigChunk, [])` e `get(DefaultConfigChunk, [])` in `zip`, `skip(1)` per saltare il root; i chunk lazy confluiscono in `unifiedConfig$` e, salvo `disableConfigUpdates`, nell'oggetto `Config` globale	core-libs/core/src/config/services/configuration.service.ts:L57-L74, core-libs/core/src/config/services/configuration.service.ts:L92-L105
`ConverterService`	svuota la cache dei converter a ogni nuovo injector, li legge con `getMulti`	core-libs/core/src/util/converter.service.ts:L38-L58

CONSUMATORE	USO	RIFERIMENTO
`HttpErrorInterceptor`	`getMulti(HttpErrorHandler)` (raccoglie anche `BadCartRequestHandler` del core carrello lazy)	core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L24-L28, feature-libs/cart/base/core/cart-base-core.module.ts:L22-L40
`PageMetaService`, `DynamicAttributeService`, `PageLayoutService`	`getMulti` di `PageMetaResolver`, `ComponentDecorator`/`SlotDecorator`, `PAGE_LAYOUT_HANDLER`	core-libs/core/src/cms/facade/page-meta.service.ts:L53-L57, core-libs/core/src/cms/services/dynamic-attribute.service.ts:L24-L30, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L41-L45
`CmsGuardsService`	`getLastValueSync(unifiedInjector.get(guardClass))` per le guard dei componenti CMS	core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48
`CmsRoutesImplService`	avvolge le `canActivate` delle child route CMS per risolvere guard fornite solo in un modulo lazy	core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L126-L136, core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L170-L181

`getLastValueSync` sottoscrive e disiscrive subito restituendo l'ultimo valore emesso in modo sincrono (core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23): grazie al `ReplaySubject` di `modules$` include tutti gli injector già noti.

3.7 Lato storefront: `CmsFeaturesService` e il rendering

`CmsFeaturesService` (`providedIn: 'root'`) mantiene `featureModulesConfig`, `componentFeatureMap` (componentType → feature) e `featureInstances` (cache dei resolver) (core-libs/storefront/cms-structure/services/cms-features.service.ts:L27-L54).

1. **Mappa componenti.** Nel costruttore attende `configInitializer.getStable('featureModules')` e popola `componentFeatureMap` solo per feature **non alias**, con `module` definito e `cmsComponents` non vuoto (core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76). `getStable` emette subito se la config è già stabile, altrimenti attende gli scope richiesti (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70).
2. `hasFeatureFor(type)`: lookup sincrono (core-libs/storefront/cms-structure/services/cms-features.service.ts:L82-L84).
3. `getCmsMapping(type)`: risolve la feature instance e restituisce il mapping del tipo, oppure `of(undefined)` se il tipo non è coperto (core-libs/storefront/cms-structure/services/cms-features.service.ts:L89-L103).
4. `resolveFeatureInstance(name)`: delega a `FeatureModulesService.resolveFeature` e mette in cache il risultato mappato con `shareReplay()` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L134-L152).
5. `createFeatureInstance`: per ogni `cmsComponents` dichiarato nella feature estrae il mapping dalla **config locale** del modulo lazy, o `{}` se assente (core-libs/storefront/cms-structure/services/cms-features.service.ts:L157-L183). La config locale è il `deepMerge` di `DefaultConfigChunk` e poi `ConfigChunk` letti con `{ self: true }` dall'injector del modulo (core-libs/storefront/cms-structure/services/cms-features.service.ts:L188-L205).

6. `getModule(type)`: restituisce il `moduleRef` solo per feature già risolte, con subscribe/unsubscribe sincrono sul resolver in cache (core-libs/storefront/cms-structure/services/cms-features.service.ts:L110-L127).

Il consumatore è `CmsComponentsService` (vedi capitolo 16):

- congela `staticCmsConfig` quando `cmsComponents` è stabile, per non essere influenzato dai moduli lazy che modificano la config globale (core-libs/storefront/cms-structure/services/cms-components.service.ts:L55-L69);
- `determineMappings(types)`: per ogni tipo non ancora risolto, se coperto da una feature accoda un resolver, altrimenti usa la config statica; attende i resolver con `forkJoin` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L79-L115);
- il mapping della feature è un **default** sovrascrivibile dalla config statica dell'app: `deepMerge({}, featureComponentMapping, staticConfig)` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L117-L141);
- `getModule(type)` delega a `CmsFeaturesService.getModule` solo per tipi coperti da feature (core-libs/storefront/cms-structure/services/cms-components.service.ts:L149-L153).

Dove avviene il caricamento durante la navigazione:

1. `CmsPageGuardService.canActivatePage` ottiene i component type della pagina e chiama `determineMappings` **prima** di valutare le guard CMS (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L80). Il route del carrello usa proprio `CmsPageGuard` (feature-libs/cart/base/root/cart-base-root.module.ts:L59-L76).
2. In rendering, `ComponentWrapperDirective.ngOnInit` chiama ancora `determineMappings([flexType])` e lancia il componente se `shouldRender` (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L67-L79); al launcher passa l'element injector di `CmsInjectorService` e il `NgModuleRef` della feature (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L90-L102).
3. `CmsInjectorService.getInjector` crea un `Injector.create` con `CmsComponentData` e i `providers` del mapping, figlio dell'element injector (core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50).
4. `DefaultComponentHandler` passa il `module` come quinto argomento di `createComponent(factory, undefined, injector, undefined, module)` (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L36-L67), cioè come `environmentInjector` (angular:packages/core/src/linker/view_container_ref.ts:L263-L275). Così il componente vede i servizi del modulo lazy.

3.8 Proxy facade: `facadeFactory`

Le facade pubbliche delle feature lib sono classi astratte nel root entry point con un provider root che crea un proxy. Esempio: `ActiveCartFacade` usa `useFactory: () => facadeFactory({ facade: ActiveCartFacade, feature: CART_BASE_CORE_FEATURE, methods: [...], async: true })` (feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46).

1. `facadeFactory(descriptor)` va chiamata in un contesto di injection: è `inject(FacadeFactoryService).create(descriptor)` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24).
2. `FacadeDescriptor<T>` ha `facade`, `feature`, `methods?`, `properties?`, `async?` (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L10-L37). A livello di tipi i metodi devono restituire `void | Observable<any>` e le proprietà essere `Observable`; se la facade ha altri membri `StrictlyAllowedFacade<T>` diventa `never` (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L39-L57).

3. `create` costruisce **un solo** `resolver$` per istanza proxy, istanzia una sottoclasse anonima della facade astratta, assegna a ogni metodo elencato una funzione `call(...)` e a ogni proprietà un `get(...)`, e marca l'istanza con `proxyFacadeInstance = true` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L110-L131). `isProxyFacadeInstance` controlla quel flag (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L133-L140).
4. `getResolver(feature, facadeClass, async)`: se la feature non è configurata restituisce un Observable d'errore `Feature X is not configured properly`; altrimenti `resolveFeature(feature)` → `moduleRef.injector.get(facadeClass)` → opzionale `delay(0)` → `shareReplay()` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57). Nel modulo lazy la stessa classe astratta è ri-fornita con l'implementazione concreta: `{ provide: ActiveCartFacade, useExisting: ActiveCartService }` (feature-libs/cart/base/core/facade/facade-providers.ts:L25-L30).
5. `call` è **eager**: crea un `connectable` con `ReplaySubject` e lo connette subito, quindi il caricamento parte anche se il chiamante non si sottoscrive (tipico dei comandi `void` come `addEntry`) (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L59-L82). Il risultato del metodo reale viene inoltrato se è un Observable, altrimenti il proxy restituisce `EMPTY` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L84-L91).
6. Le proprietà sono **lazy**: `resolver$.pipe(switchMap(service => service[property]))` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L94-L108).
7. `async: true` serve, secondo il commento, a rendere pronto uno store NgRx lazy (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L36). `DA VERIFICARE`: il motivo preciso per cui serve il `delay(0)` rispetto alla registrazione di `StoreModule.forFeature` non è verificato nel codice NgRx (dossier Q6).

Le facade in quanto pattern (command/query, eventi) sono trattate nel capitolo 20; qui interessa solo il loro legame con il lazy loading.

---

4. Diagrammi

4.1 Componenti del sistema

Ogni freccia corrisponde a una dipendenza di costruttore o a una chiamata citata nel paragrafo 3.

■ **TAVOLA 12.1** diagramma a tutta pagina, in fondo al capitolo

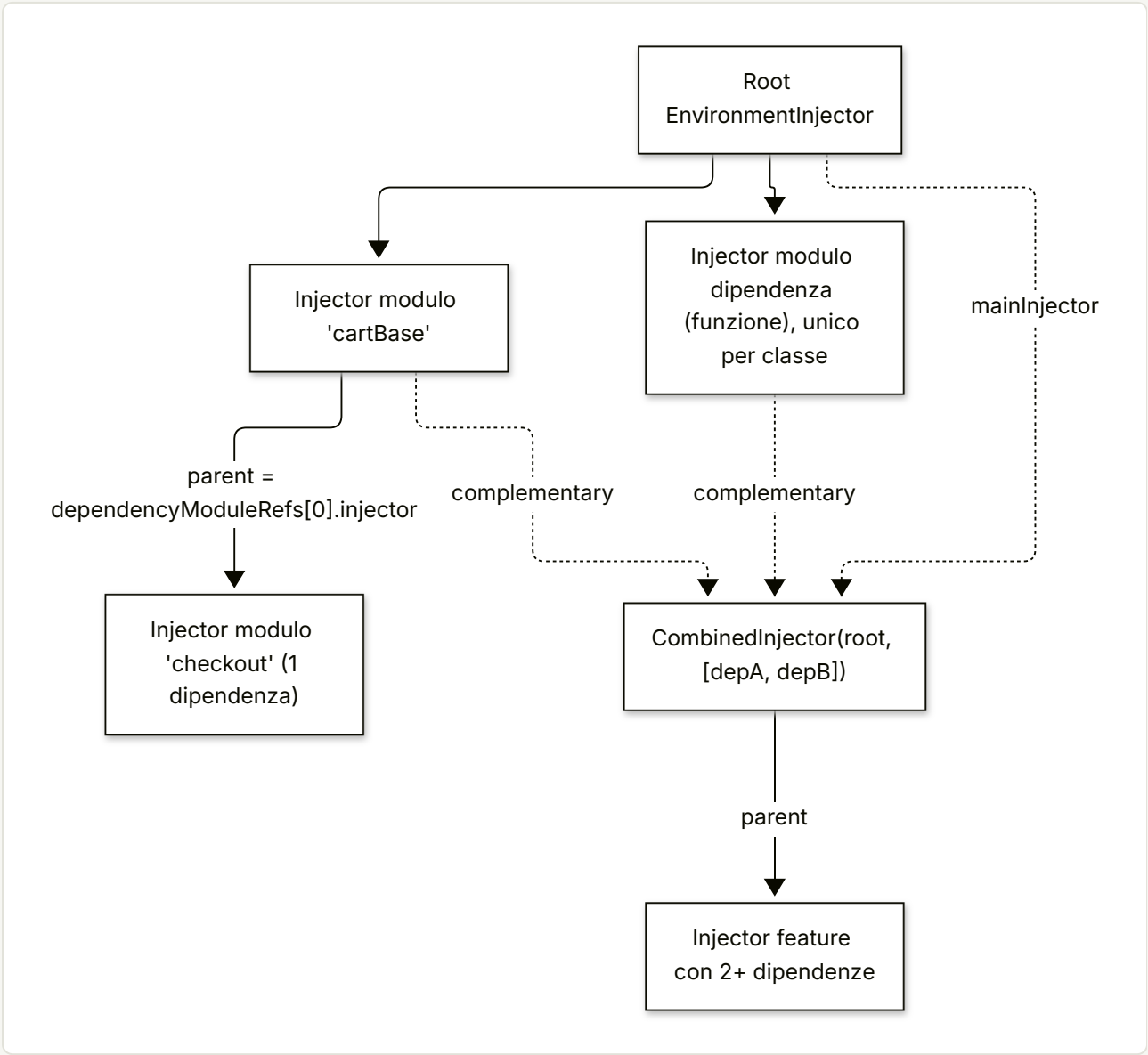
4.2 Navigazione verso la pagina carrello (feature `cartBase`)

■ **TAVOLA 12.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti dei passi: 1 core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L71-L75; 2 core-libs/storefront/cms-structure/services/cms-components.service.ts:L92-L99; 3 core-libs/storefront/cms-structure/services/cms-components.service.ts:L121-L123; 4 core-libs/storefront/cms-structure/services/cms-features.service.ts:L134-L152; 5-7 core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L72, core-libs/core/src/lazy-loading/feature-modules.service.ts:L108-L123; 8 core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L86; 9-10 core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L236; 11-12 angular:packages/core/src/render3/ng_module_ref.ts:L127-L134; 13 core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L175; 14-16 core-libs/

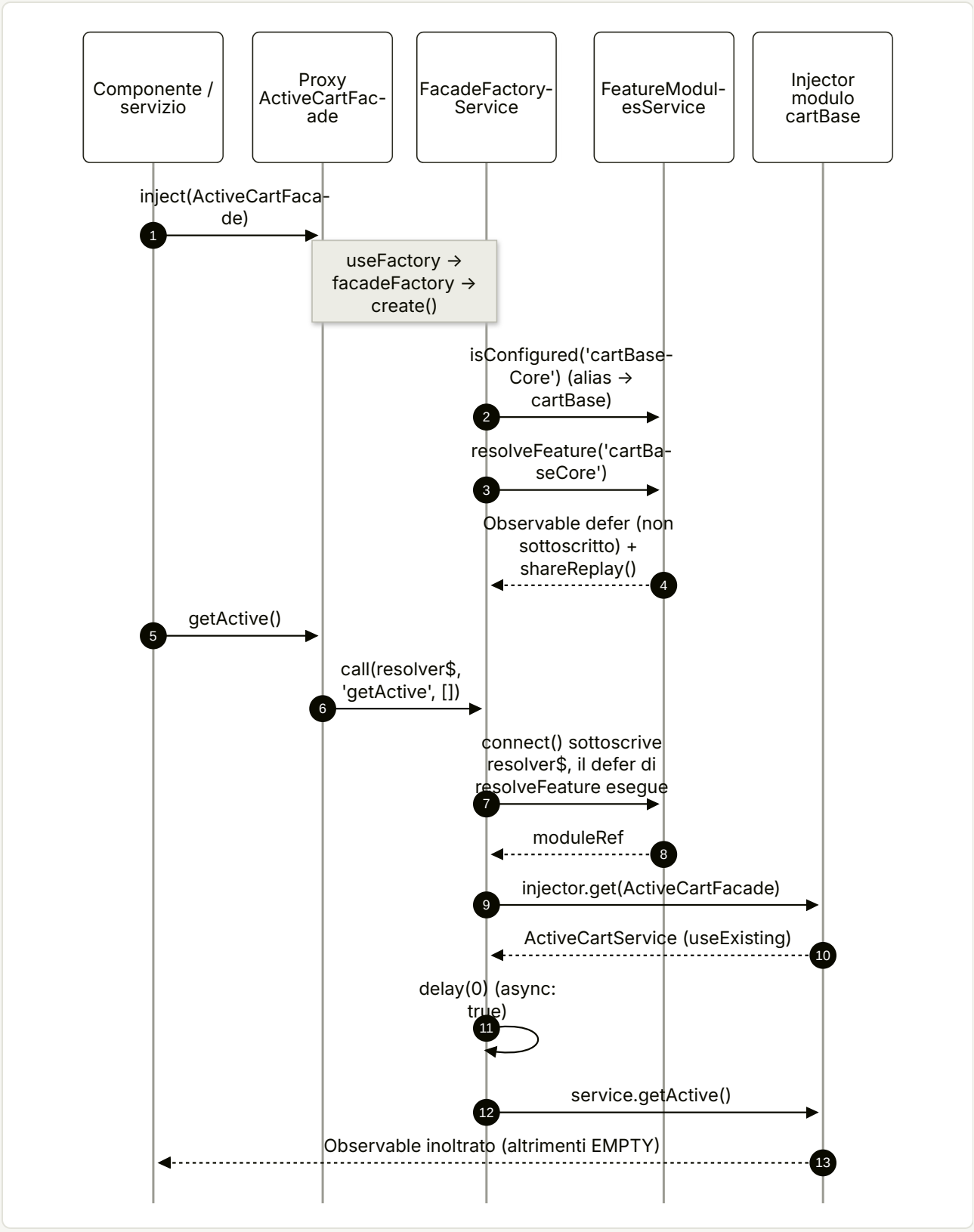
core/src/lazy-loading/lazy-modules.service.ts:L99-L106, core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35; 19 core-libs/storefront/cms-structure/services/cms-features.service.ts:L157-L205; 21 core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L135; 23 core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L78-L80.

4.3 Gerarchia degli injector con dipendenze



Fonti: scelta del parent core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L94; modulo dipendenza creato col root injector core-libs/core/src/lazy-loading/lazy-modules.service.ts:L121-L129; ordine di ricerca del `CombinedInjector` core-libs/core/src/util/combined-injector.ts:L53-L70. Il nodo `MULTI` è illustrativo: nel codice analizzato non c'è una feature reale con due o più dipendenze (dossier Q5 per le dipendenze-funzione).

4.4 Proxy facade alla prima chiamata



Fonti: `feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46`; `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L91`, `core-libs/core/src/lazy-loading/facade-factory/facade-`

factory.service.ts:L110-L131; core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35; feature-libs/cart/base/core/facade/facade-providers.ts:L25-L30.

---

5. Estratti di codice reale

Scelta del parent injector e pipeline di creazione — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L108

```ts
public resolveModuleInstance(
  moduleFunc: () => Promise<any>,
  feature?: string,
  dependencyModuleRefs: NgModuleRef<any>[] = []
): Observable<NgModuleRef<any>> {
  let parentInjector: Injector;

  if (!dependencyModuleRefs.length) {
    parentInjector = this.injector;
  } else if (dependencyModuleRefs.length === 1) {
    parentInjector = dependencyModuleRefs[0].injector;
  } else {
    parentInjector = new CombinedInjector(
      this.injector,
      dependencyModuleRefs.map((moduleRef) => moduleRef.injector)
    );
  }

  return this.resolveModuleFactory(moduleFunc).pipe(
    map(([moduleFactory]) => moduleFactory.create(parentInjector)),
    concatMap((moduleRef) => this.runModuleInitializersForModule(moduleRef)),
    tap((moduleRef) =>
      this.events.dispatch(
        createFrom(ModuleInitializedEvent, {
          feature,
          moduleRef,
        })
      )
    )
  );
}
```

Resolver di feature con cache e dipendenze — core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73

```ts
resolveFeature(featureName: string): Observable<NgModuleRef<any>> {
  featureName = this.resolveFeatureAlias(featureName);
```

```
  return defer(() => {
    if (!this.features.has(featureName)) {
      if (!this.isConfigured(featureName)) {
        return throwError(
          () =>
            new Error('No module defined for Feature Module ' + featureName)
        );
      }

      const featureConfig = this.getFeatureConfig(featureName);

      this.features.set(
        featureName,
        this.resolveDependencies(featureConfig?.dependencies).pipe(
          switchMap((deps) =>
            this.lazyModules.resolveModuleInstance(
              featureConfig?.module as () => Promise<any>,
              featureName,
              deps
            )
          ),
          shareReplay()
        )
      );
    }

    return this.features.get(featureName) ?? EMPTY;
  });
}
```

MODULE_INITIALIZER **letto solo dall'injector del modulo —** core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L175

TS

```
public runModuleInitializersForModule(
  moduleRef: NgModuleRef<any>
): Observable<NgModuleRef<any>> {
  const moduleInits: any[] = moduleRef.injector.get<any[]>(
    MODULE_INITIALIZER,
    [],
    { self: true }
  );
  const asyncInitPromises: Promise<any>[] =
    this.runModuleInitializerFunctions(moduleInits);
  if (asyncInitPromises.length) {
    return from(Promise.all(asyncInitPromises)).pipe(
      tap({
        error: (error) => {
          this.logger.error(
```

```
            'MODULE_INITIALIZER promise was rejected while lazy loading a module.',
            error
          );
        },
      }),
      switchMap(() => of(moduleRef))
    );
  } else {
    return of(moduleRef);
  }
}
```

UnifiedInjector.get / getMulti — core-libs/core/src/lazy-loading/unified-injector.ts:L52-L93 (abbreviato)

```ts
get<T>(
  token: Type<T> | InjectionToken<T> | AbstractType<T>,
  notFoundValue?: T
): Observable<T> {
  return this.injectors$.pipe(
    map((injector, index) =>
      injector.get<T>(
        token,
        notFoundValue ?? NOT_FOUND_SYMBOL,
        // we want to get only Self instances from all injectors except the
        // first one, which is a root injector
        index ? { self: true } : undefined
      )
    ),
    filter((instance) => instance !== NOT_FOUND_SYMBOL)
  );
}
// ...
  return this.get(token, []).pipe(
    filter((instances) => {
      if (!Array.isArray(instances)) {
        throw new Error(
          `Multi-providers mixed with single providers for ${token.toString()}!`
        );
      }
      return instances.length > 0;
    }),
    scan((acc, services) => [...acc, ...services], [])
  );
```

Root module della feature: solo `cmsComponents` **+ alias** — feature-libs/cart/base/root/cart-base-root.module.ts:L26-L57 (abbreviato)

```ts
export function defaultCartComponentsConfig() {
  const config = {
    featureModules: {
      [CART_BASE_FEATURE]: {
        cmsComponents: [
          'CartApplyCouponComponent',
          'CartComponent',
          // ...
        ],
      },
      [MINI_CART_FEATURE]: {
        cmsComponents: ['MiniCartComponent'],
      },
      [ADD_TO_CART_FEATURE]: {
        cmsComponents: ['ProductAddToCartComponent'],
      },
      // By default core is bundled together with components.
      // The cart lib should keep using this default.
      // ...
      [CART_BASE_CORE_FEATURE]: CART_BASE_FEATURE,
    },
  };
  return config;
}
```

L'app fornisce il `module` — projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L45 (abbreviato)

```ts
@NgModule({
  imports: [CartBaseRootModule],
  providers: [
    provideConfig({
      featureModules: {
        [CART_BASE_FEATURE]: {
          module: () =>
            import('./cart-base-wrapper.module').then(
              (m) => m.CartBaseWrapperModule
            ),
        },
      },
    }),
    provideConfig({
      featureModules: {
        [MINI_CART_FEATURE]: {
          module: () =>
            import('@spartacus/cart/base/components/mini-cart').then(
              (m) => m.MiniCartModule
            ),
```

```
      },
    },
  }),
  // ...
```

Config locale della feature (self) — core-libs/storefront/cms-structure/services/cms-features.service.ts:L188-L205

```ts
private resolveFeatureConfiguration(featureInjector: Injector): CmsConfig {
  // get config chunks from feature lib
  const featureConfigChunks = featureInjector.get<any[]>(ConfigChunk, [], {
    self: true,
  });
  // get default config chunks from feature lib
  const featureDefaultConfigChunks = featureInjector.get<any[]>(
    DefaultConfigChunk,
    [],
    { self: true }
  );

  return deepMerge(
    {},
    ...(featureDefaultConfigChunks ?? []),
    ...(featureConfigChunks ?? [])
  ) as CmsConfig;
}
```

Costruzione del proxy facade — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L110-L131

```ts
create<T extends object>({
  facade,
  feature,
  methods,
  properties,
  async,
}: FacadeDescriptor<T>): T {
  const resolver$ = this.getResolver(feature, facade, async);

  const result: any = new (class extends (facade as any) {})();
  (methods ?? []).forEach((method) => {
    result[method] = (...args: any[]) =>
      this.call(resolver$, method as string, args);
  });
  (properties ?? []).forEach((property) => {
    result[property] = this.get(resolver$, property as string);
  });
```

```
    result[PROXY_FACADE_INSTANCE_PROP] = true;

    return result;
  }
```

Initializer asincrono reale — feature-libs/cart/base/core/cart-persistence.module.ts:L19-L31

```ts
export function cartStatePersistenceFactory(
  cartStatePersistenceService: MultiCartStatePersistenceService,
  configInit: ConfigInitializerService
): () => Promise<Config> {
  return () =>
    lastValueFrom(
      configInit.getStable('context').pipe(
        tap(() => {
          cartStatePersistenceService.initSync();
        })
      )
    );
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`featureModules`	`{ [featureName: string]: FeatureModuleConfig | string }`	nessuno nel core; contributi dai root module delle lib via `provideDefaultConfigFactory` (es. cart)	core-libs/core/src/cms/config/cms-config.ts:L129-L129	core-libs/core/src/lazy-loading/feature-modules.service.ts:L80-L100; core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76
`featureModules.<name>.module`	`() => Promise<any>` (classe NgModule o NgModuleFactory)	assente: lo fornisce l'app (es. projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L26-L35)	core-libs/core/src/cms/config/cms-config.ts:L113-L113	core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35, core-libs/core/src/lazy-loading/feature-modules.service.ts:L59-L64; core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L236
`featureModules.<name>.dependencies`	`((() => Promise<any>) | string)[]`	`[]` (default del parametro)	core-libs/core/src/cms/config/cms-config.ts:L117-L117	core-libs/core/src/lazy-loading/feature-modules.service.ts:L108-L123
`featureModules.<name>.cmsComponents`	`string[]`	assente	core-libs/core/src/cms/config/cms-config.ts:L121-L121	core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L72, core-libs/storefront/cms-structure/services/cms-features.service.ts:L177-L181

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`featureModules.<alias>` (stringa)	`string` (nome di un'altra feature)	es. `cartBaseCore: 'cartBase'`	`feature-libs/cart/base/root/cart-base-root.module.ts:L53-L53`	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100` (ignorato da `CmsFeaturesService`, `core-libs/storefront/cms-structure/services/cms-features.service.ts:L66-L66`)
`features.disableConfigUpdates` (feature toggle, `@deprecated`)	`boolean`	disabilitato ("It's disabled by default and recommended to keep it so") ⇒ la config globale viene aggiornata con i chunk lazy	`core-libs/core/src/features-config/config/features-config.ts:L100-L116`	`core-libs/core/src/config/services/configuration.service.ts:L103-L105`
`FacadeDescriptor.async`	`boolean`	`false` (default del parametro di `getResolver`)	`core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L36-L36`	`core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L41-L55`
Token `MODULE_INITIALIZER`	multi `(() => any)[]`	`[]` (lazy, self) / opzionale (eager)	`core-libs/core/src/lazy-loading/tokens.ts:L19-L20`	`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L153-L157`; `core-libs/core/src/lazy-loading/lazy-loading.module.ts:L47-L47`

Sul toggle `disableConfigUpdates`: il JSDoc spiega che dalla 3.0 la config unificata emette una nuova configurazione a ogni istanziazione di modulo lazy, ripubblicandola anche sul token globale `Config`; attivandolo, chi vuole la config dei moduli lazy deve usare `ConfigurationService.unifiedConfig$` (`core-libs/core/src/features-config/config/features-config.ts:L100-L116`). Vedi capitolo 42 e l'appendice A09.

---

7. Commenti e note notevoli dal sorgente

- `core-libs/core/src/lazy-loading/tokens.ts:L9-L18` — "It is, in a way, an APP_INITIALIZER for lazy loaded modules" e la spiegazione del fallback eager (con il typo "Therfore").
- `core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L10-L15` — "This event is thrown for cms driven lazy loaded feature modules amd it's dependencies" (typo "amd"); il JSDoc di `feature` spiega come distinguere moduli feature e dipendenza (`:L21-L27`).
- `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L73-L73` — "New module instance will be created with each call."
- `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L113-L113` — "Module will be instantiated only once, at first request for a this specific module class".
- `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L228-L229` — "using compiler here is for jit compatibility, there is no overhead for aot production builds as it will be stubbed". In Angular 21 lo "stub" è letterale: `compileModuleAsync` è un `Promise.resolve` attorno a una nuova factory (`angular:packages/core/src/linker/compiler.ts:L57-L66`).

- core-libs/core/src/lazy-loading/lazy-modules.service.ts:L203-L203 — messaggio di log "init function throwed an error" (typo).
- core-libs/core/src/lazy-loading/feature-modules.service.ts:L17-L20 — "Each resolver runs only once and caches the result."
- core-libs/core/src/lazy-loading/feature-modules.service.ts:L75-L79 — JSDoc incompleto ("Resolve") di `getFeatureConfig`.
- core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L35 — "Required to make lazy NgRx store feature ready."
- core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L97-L97 — "Property has to be an aobservable" (typo).
- core-libs/core/src/config/services/configuration.service.ts:L51-L53 — la sottoscrizione eager ai chunk lazy è dovuta alla compatibilità con l'aggiornamento della config globale ("It will be possible to make it lazy, when we drop this compatibility feature").
- core-libs/storefront/cms-structure/services/cms-components.service.ts:L64-L66 — "lazy-loaded modules can affect global configuration resulting in non-deterministic state": motivazione di `staticCmsConfig`.
- feature-libs/cart/base/root/cart-base-root.module.ts:L45-L52 — perché core e componenti del carrello restano nello stesso chunk (alias `cartBaseCore → cartBase`): ritardare il caricamento della lib e garantire che sia caricata quando serve.
- API Angular deprecate su cui poggia il meccanismo: `Compiler` (angular:packages/core/src/linker/compiler.ts:L48-L49), `NgModuleFactory` (angular:packages/core/src/linker/ng_module_factory.ts:L62-L75), `APP_INITIALIZER` "deprecated from v19.0.0, use provideAppInitializer instead" (angular:packages/core/src/application/application_init.ts:L36-L38), `createComponent(ComponentFactory, ...)` (angular:packages/core/src/linker/view_container_ref.ts:L263-L275), usato con `ComponentFactoryResolver` (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L76-L85). Vedi appendice A10.
- Nessun `TODO`/`FIXME`/`@deprecated` nei 12 file di `core-libs/core/src/lazy-loading/` (verifica del dossier).

8. Extension point

EXTENSION POINT	TIPO	DEFINIZIONE	EXPORT
`featureModules` (config)	chiave di `CmsConfig`	core-libs/core/src/cms/config/cms-config.ts:L129-L129	`@spartacus/core` (core-libs/core/public_api.ts:L14-L14)
`MODULE_INITIALIZER`	InjectionToken multi	core-libs/core/src/lazy-loading/tokens.ts:L19-L20	`@spartacus/core` (core-libs/core/src/lazy-loading/index.ts:L9-L9)
`facadeFactory` / `FacadeDescriptor`	funzione + interfaccia	core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24; core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L10-L37	`@spartacus/core` (core-libs/core/src/lazy-loading/facade-factory/index.ts:L8-L9)

EXTENSION POINT	TIPO	DEFINIZIONE	EXPORT
`ModuleInitializedEvent`	`CxEvent` ascoltabile via `EventService.get`	`core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L32`	`@spartacus/core` (`core-libs/core/src/lazy-loading/index.ts:L7-L7`)
`UnifiedInjector`	servizio root (`injectors$`, `get`, `getMulti`)	`core-libs/core/src/lazy-loading/unified-injector.ts:L27-L94`	`@spartacus/core` (`core-libs/core/src/lazy-loading/index.ts:L10-L10`)
`FeatureModulesService`	servizio root; metodi `protected` sovrascrivibili `getFeatureConfig`, `resolveFeatureAlias`, `resolveDependencies`	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L80-L123`	`@spartacus/core` (`core-libs/core/src/lazy-loading/index.ts:L11-L11`)
`LazyModulesService`	servizio root; metodi pubblici `resolveModuleInstance`, `resolveDependencyModuleInstance`, `runModuleInitializersForModule`, `runModuleInitializerFunctions`, `modules$`	`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L41-L208`	`@spartacus/core` (`core-libs/core/src/lazy-loading/index.ts:L8-L8`)
`FacadeFactoryService`	servizio root; metodi `protected` `getResolver`, `call`, `get`	`core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L108`	`@spartacus/core` (`core-libs/core/src/lazy-loading/facade-factory/index.ts:L7-L7`)
`CmsFeaturesService`	servizio root storefront	`core-libs/storefront/cms-structure/services/cms-features.service.ts:L33-L206`	`@spartacus/storefront` (`core-libs/storefront/public_api.ts:L23-L23`)
`OutletModule.forChild()`	registrazione outlet in moduli lazy via `MODULE_INITIALIZER`	`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88`	`@spartacus/storefront`
`LazyLoadingModule`	NgModule `forRoot`	`core-libs/core/src/lazy-loading/lazy-loading.module.ts:L38-L53`	**non** esportato: interno, usato solo da `BaseCoreModule`
`CombinedInjector`	classe util	`core-libs/core/src/util/combined-injector.ts:L27-L72`	**non** esportato (`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L30-L30`)

Come si usano in pratica:

- **Creare una feature lib lazy** (root module con `cmsComponents` e facade `facadeFactory`, modulo lazy fornito dall'app): ricetta in Parte III, capitolo 53.
- **Sostituire l'implementazione di una facade lazy** o un servizio fornito nel modulo lazy (DI, `useExisting`/`useClass`): Parte III, capitolo 50.
- **Sostituire un componente CMS coperto da una feature**: basta la config statica `cmsComponents` dell'app, che vince sul mapping della feature (`core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L135`); ricetta in Parte III, capitolo 51.
- **Evitare API interne** (`LazyLoadingModule`, `CombinedInjector`): Parte III, capitolo 54.

Uno schema minimale, che usa solo simboli esportati (`provideDefaultConfigFactory` e `provideConfig` da `@spartacus/core`, `CART_BASE_FEATURE` da `@spartacus/cart/base/root`), è il seguente. **È illustrativo** (non copiato dal repo) e ricalca la struttura di carrello e checkout:

```ts
// illustrativo — non presente nel repository
// root (eager): dichiara i componenti CMS e la dipendenza
provideDefaultConfigFactory(() => ({
  featureModules: {
    myFeature: {
      cmsComponents: ['MyCmsComponent'],
      dependencies: [CART_BASE_FEATURE],
    },
    myFeatureCore: 'myFeature', // alias, come cartBaseCore -> cartBase
  },
}));

// app: fornisce il modulo lazy
provideConfig({
  featureModules: {
    myFeature: { module: () => import('./my-feature.module').then((m) => m.MyFeatureModule) },
  },
});
```

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `cmsComponents` **senza** `module` **non basta.** `CmsFeaturesService` mappa un componente su una feature solo se la feature ha `module` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L68). Se l'app importa solo il root module e dimentica il `provideConfig` con `module`, i componenti cadono sulla config statica e in dev mode compare "No component implementation found for the CMS component type" (core-libs/storefront/cms-structure/services/cms-components.service.ts:L159-L180).
2. **La facade "ricorda" lo stato di configurazione al momento dell'injection.** `getResolver` verifica `isConfigured` in `create`, cioè alla prima injection del proxy (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L117-L117): se in quel momento `module` non è ancora in config, il proxy resta per sempre un Observable d'errore "Feature X is not configured properly".
3. **Errori silenziosi nei comandi** `void`**.** `call` connette subito un `connectable` con connector `ReplaySubject` e restituisce un Observable derivato da esso (`switchMap` verso il risultato o `EMPTY`) (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L75-L91): se il chiamante non si sottoscrive, l'eventuale errore di risoluzione non viene osservato da nessuno. In RxJS 7.8 `connect()` sottoscrive il subject alla sorgente (deps:rxjs-7.8.2/package/src/internal/observable/connectable.ts:L53-L61); `Subject.error` senza observer memorizza solo l'errore (deps:rxjs-7.8.2/package/src/internal/Subject.ts:L73-L84), che verrà riemesso solo a un eventuale subscriber successivo (deps:rxjs-7.8.2/package/src/internal/ReplaySubject.ts:L83-L83, deps:rxjs-7.8.2/package/src/internal/Subject.ts:L137-L144).
4. **Metodi non elencati in** `methods`**.** Il proxy è una sottoclasse anonima di una classe astratta: i metodi non elencati non vengono definiti sull'istanza e chiamarli dà `TypeError` (core-libs/core/src/lazy-loading/

facade-factory/facade-factory.service.ts:L119-L126). Aggiungere un metodo a una facade richiede di aggiornare anche `methods`.

5. **Valori di ritorno non-Observable scartati** dal proxy (`EMPTY`) (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L84-L91). Coerente con il vincolo di tipo `void | Observable<any>` (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L39-L57).
6. **Initializer che restituiscono Observable non vengono attesi** (solo promise-like): usa `lastValueFrom`/`firstValueFrom` come `CartPersistenceModule` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L195-L197, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L213-L215, feature-libs/cart/base/core/cart-persistence.module.ts:L23-L30).
7. `MODULE_INITIALIZER` **e** `self: true`**.** Un initializer fornito nel root non viene rieseguito al caricamento lazy, e uno fornito in un modulo lazy non gira mai al bootstrap (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L153-L157). Lo stesso modulo usato eager o lazy cambia quindi il momento di esecuzione (core-libs/core/src/lazy-loading/tokens.ts:L14-L17).
8. **Moduli dipendenza: istanza unica, initializer ripetuti.** Il `moduleRef` è in cache, ma `runModuleInitializersForModule` e il dispatch di `ModuleInitializedEvent` stanno dopo il `map` e quindi avvengono a ogni risoluzione (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L121-L138): initializer rieseguiti ed emissioni duplicate dello stesso injector in `injectors$`, con possibili istanze duplicate accumulate da `getMulti` via `scan` (core-libs/core/src/lazy-loading/unified-injector.ts:L91-L91). `DA VERIFICARE`: impatto reale; nessun uso di dipendenze-funzione trovato nell'app demo né nei root module (dossier Q5).
9. **Stessa classe NgModule in due feature = due istanze.** `resolveModuleInstance` crea un nuovo modulo a ogni chiamata e la deduplica è per *nome di feature* (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L70-L77, core-libs/core/src/lazy-loading/feature-modules.service.ts:L46-L69): se due feature diverse importano lo stesso modulo (o due nomi non-alias puntano allo stesso `import()`), i servizi vengono istanziati due volte. Per condividere, usa un alias stringa o una dipendenza. `DA VERIFICARE`: la conseguenza concreta su store/effects NgRx duplicati non è stata verificata.
10. **Dipendenze circolari tra feature (A → B → A).** Il resolver di A viene messo in cache prima che la sua pipeline sia sottoscritta; B troverebbe in cache l'Observable di A che attende B (core-libs/core/src/lazy-loading/feature-modules.service.ts:L46-L71, core-libs/core/src/lazy-loading/feature-modules.service.ts:L114-L117). `DA VERIFICARE`: ragionamento statico, presumibile stallo senza errore.
11. **Alias ciclici** (`a: 'b', b: 'a'`) fanno girare all'infinito il `while` di `resolveFeatureAlias` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L96-L98).
12. `cmsComponents` **su un alias è ignorato.** `CmsFeaturesService` salta le voci stringa (core-libs/storefront/cms-structure/services/cms-features.service.ts:L66-L66): i componenti vanno dichiarati sulla feature "reale".
13. `getModule` **è sincrono e opportunistico.** Restituisce `undefined` finché la feature non è risolta (core-libs/storefront/cms-structure/services/cms-features.service.ts:L119-L126); per questo `ComponentWrapperDirective` chiama `determineMappings` prima di lanciare il componente (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L67-L79).
14. **Il mapping CMS della feature viene solo dal modulo lazy.** È letto dalla config locale (`self: true`) dell'injector del modulo; config fornite altrove non contano come default della feature, e la config statica dell'app vince comunque (core-libs/storefront/cms-structure/services/cms-features.service.ts:L188-L205, core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L135).
15. **La config globale cambia dopo i caricamenti lazy.** `ConfigurationService` fonde i chunk dei moduli lazy anche nell'oggetto `Config` (salvo `disableConfigUpdates`) (core-libs/core/src/config/services/configuration.service.ts:L92-L105); `CmsComponentsService` si protegge con uno shallow copy `staticCmsConfig` preso a config stabile (core-libs/storefront/cms-structure/services/cms-components.service.ts:L61-L68). Un servizio che legge `Config` in momenti diversi può vedere valori diversi.

16. `CombinedInjector` **non supporta** `self`: con `self: true` restituisce `notFoundValue` o lancia (core-libs/core/src/util/combined-injector.ts:L44-L51). Non va mai usato come injector di un modulo, solo come parent.
17. **Multi e single provider sullo stesso token** fanno lanciare `getMulti` (core-libs/core/src/lazy-loading/unified-injector.ts:L84-L88).
18. `UnifiedInjector.get` **non lancia mai** per token assenti negli injector lazy, perché usa `NOT_FOUND_SYMBOL` come `notFoundValue` (core-libs/core/src/lazy-loading/unified-injector.ts:L58-L66). Un token assente anche nel root produce semplicemente nessuna emissione.
19. **Nessun modulo feature viene mai distrutto.** `modules$` usa un `ReplaySubject` illimitato e `ngOnDestroy` distrugge solo i moduli dipendenza (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L51-L54, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L238-L245). In SSR, dove l'app è ricreata a ogni richiesta, questo ha impatto solo se il root injector non viene distrutto. DA VERIFICARE: non verificato il ciclo di vita dell'app server in questo capitolo (vedi capitolo 39).
20. **Listener tardivi di** `ModuleInitializedEvent`**.** L'evento passa su un `Subject` semplice; solo `modules$` fa replay. Chi chiama `EventService.get(ModuleInitializedEvent)` tardi perde gli eventi già emessi: per l'elenco completo usa `LazyModulesService.modules$` o `UnifiedInjector.injectors$` (core-libs/core/src/event/event.service.ts:L111-L126, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L55).
21. **Eccezioni sincrone negli initializer eager** escono prima del `.catch` della promise in `moduleInitializersFactory`, ma sono comunque loggate da `runModuleInitializerFunctions` (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L23-L28, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L201-L207).
22. **Initializer lazy che fallisce** blocca la feature: l'errore viene loggato e propagato nello stream, quindi `resolveFeature` termina in errore (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L161-L171).
23. `shareReplay()` **senza** `refCount` in tutte le cache (core-libs/core/src/lazy-loading/feature-modules.service.ts:L66-L66, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L56-L56, core-libs/storefront/cms-structure/services/cms-features.service.ts:L145-L145): il caricamento non viene annullato se tutti i subscriber si disiscrivono (`resetOnRefCountZero: refCount`, default `false`). In caso di errore invece la cache **non** memorizza l'errore: `shareReplay` usa `share` con `resetOnError: true` (deps:rxjs-7.8.2/package/src/internal/operators/shareReplay.ts:L167-L172), che riporta lo stato a "freddo" (deps:rxjs-7.8.2/package/src/internal/operators/share.ts:L167-L171, deps:rxjs-7.8.2/package/src/internal/operators/share.ts:L226-L230): l'Observable resta nella `Map` e il subscriber successivo risottoscrive la pipeline (nuovo tentativo di risoluzione).
24. **SSR e import dinamici delle feature.** `LazyComponentHandler` registra gli `import()` dei componenti lazy come `PendingTasks` (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L38-L70), mentre `LazyModulesService` e `FeatureModulesService` non registrano pending task propri. DA VERIFICARE: in SSR la stabilità durante l'`import()` di una feature dipende presumibilmente dal fatto che la risoluzione avviene dentro `CmsPageGuard`, che il router attende (dossier Q2).

---

10. Come fare debug

- **La feature è configurata?** Inietta `FeatureModulesService` e chiama `isConfigured('cartBase')`: è `true` solo se, dopo l'alias, esiste `module` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35). Se è `false`, manca il `provideConfig` dell'app.

- **Il componente è coperto da una feature?** `CmsFeaturesService.hasFeatureFor('CartComponent')` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L82-L84). `false` con `isConfigured === true` indica `cmsComponents` mancanti o dichiarati su un alias (trappole 1 e 12).
- **Quando viene caricato un modulo?** Sottoscriviti presto a `LazyModulesService.modules$` (replay completo) oppure, se ti interessa il nome, a `EventService.get(ModuleInitializedEvent)` leggendo `feature` (assente per i moduli dipendenza) (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L55, core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L32).
- **Chunk di rete.** Nella tab Network del browser l'`import()` di `module` produce il download di un chunk JS: per il carrello dell'app demo i chunk di `cartBase`, `miniCart` e `addToCart` sono distinti (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L26-L55).
- **Log degli initializer.** Cerca nei log i messaggi "MODULE_INITIALIZER promise was rejected while lazy loading a module.", "MODULE_INITIALIZER promise was rejected during app initialization." e "MODULE_INITIALIZER init function throwed an error." (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L164-L167, core-libs/core/src/lazy-loading/lazy-loading.module.ts:L29-L32, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L202-L205).
- **Errori tipici.** "No module defined for Feature Module X" (da `resolveFeature`, core-libs/core/src/lazy-loading/feature-modules.service.ts:L48-L51), "Feature X is not configured properly" (da un proxy facade, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L44-L46), "Multi-providers mixed with single providers" (core-libs/core/src/lazy-loading/unified-injector.ts:L85-L87), "CombinedInjector should be used as a parent injector" (core-libs/core/src/util/combined-injector.ts:L48-L50).
- **È un proxy?** `FacadeFactoryService.isProxyFacadeInstance(facade)` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L138-L140). Se ottieni il proxy dentro il modulo lazy, l'injector del modulo non ha il provider concreto della facade.
- **Quali istanze vede il root?** `getLastValueSync(unifiedInjector.getMulti(TOKEN))` restituisce le istanze accumulate da tutti gli injector noti (core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93, core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23).
- **Breakpoint utili**: `FeatureModulesService.resolveFeature` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L42), `LazyModulesService.resolveModuleInstance` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L85-L85), `CmsFeaturesService.createFeatureInstance` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L157-L157), `CmsComponentsService.determineMappings` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L79-L79).

---

11. Domande di autoverifica

1. **Chi fornisce di solito la chiave `module` di una feature, e chi `cmsComponents`?** La libreria (root module eager) fornisce `cmsComponents`, `dependencies` e gli alias; l'app fornisce `module` con `provideConfig`. Il contributo della lib è un `DefaultConfigChunk`, quello dell'app un `ConfigChunk`: si fondono nel `Config` globale, che è `deepMerge({}, DefaultConfig, RootConfig)`. — feature-libs/cart/base/root/cart-base-root.module.ts:L26-L53, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L55, core-libs/core/src/config/config-providers.ts:L18-L27, core-libs/core/src/config/config-providers.ts:L69-L79, core-libs/core/src/config/config-tokens.ts:L10-L12.
2. **Cosa succede chiamando `resolveFeature('cartBaseCore')`?** L'alias viene risolto a `cartBase` prima di tutto, quindi si condivide lo stesso resolver in cache di `cartBase`: nessun modulo aggiuntivo. — core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L43, core-libs/core/src/lazy-loading/

feature-modules.service.ts:L95-L100, feature-libs/cart/base/root/cart-base-root.module.ts:L53-L53.

3. **Qual è il parent injector del modulo** `checkout`**?** Avendo una sola dipendenza (`cartBase`), il parent è l'injector del modulo `cartBase`; con due o più dipendenze sarebbe un `CombinedInjector`. — feature-libs/checkout/base/root/checkout-root.module.ts:L40-L43, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L94.
4. **Un** `MODULE_INITIALIZER` **che restituisce un Observable viene atteso prima che la feature sia considerata pronta?** No: vengono raccolti solo valori con `then` funzione; bisogna convertire con `lastValueFrom`/`firstValueFrom`. — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L187-L200, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L213-L215, feature-libs/cart/base/core/cart-persistence.module.ts:L19-L31.
5. **Perché** `HttpErrorInterceptor`**, registrato nel root, vede** `BadCartRequestHandler` **fornito nel core lazy del carrello?** Perché usa `UnifiedInjector.getMulti(HttpErrorHandler)`, che accumula le istanze trovate (con `self: true`) negli injector dei moduli lazy man mano che `modules$` li emette. — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L24-L28, core-libs/core/src/lazy-loading/unified-injector.ts:L52-L93, feature-libs/cart/base/core/cart-base-core.module.ts:L22-L40.
6. **Perché** `UnifiedInjector` **non perde i moduli caricati prima della sua creazione?** Perché `modules$` è un `connectable` con `ReplaySubject`, connesso nel costruttore di `LazyModulesService`, che a sua volta è istanziato al bootstrap come dipendenza dell'`APP_INITIALIZER` di `LazyLoadingModule`. — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L68, core-libs/core/src/lazy-loading/lazy-loading.module.ts:L44-L48.
7. **In che punto della navigazione viene scaricato il chunk di** `cartBase`**?** In `CmsPageGuardService.canActivatePage`, che chiama `determineMappings` sui component type della pagina prima delle guard CMS; `determineMappings` passa per `CmsFeaturesService.getCmsMapping` → `FeatureModulesService.resolveFeature`. — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L80, core-libs/storefront/cms-structure/services/cms-components.service.ts:L79-L115, core-libs/storefront/cms-structure/services/cms-features.service.ts:L89-L103.
8. **Se l'app configura** `cmsComponents.CartComponent` **con un proprio componente, vince lei o la feature?** Vince l'app: il mapping della feature è un default, `deepMerge({}, featureComponentMapping, staticConfig)`. — core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L135.
9. **Chiamare** `activeCartFacade.addEntry(...)` **senza sottoscriversi carica comunque la feature?** Sì: `call` connette subito un `connectable` con `ReplaySubject`, quindi la risoluzione della feature e l'invocazione partono comunque. — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L82.
10. **Come fa un componente CMS lazy a ottenere i servizi del proprio modulo, se il suo element injector è figlio dell'injector dell'elemento host?** Il `NgModuleRef` della feature viene passato come quinto argomento (`environmentInjector`) di `createComponent`. — core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L90-L102, core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L62-L68, angular:packages/core/src/linker/view_container_ref.ts:L263-L275.

TAVOLA 12.1 **4.1 Componenti del sistema**

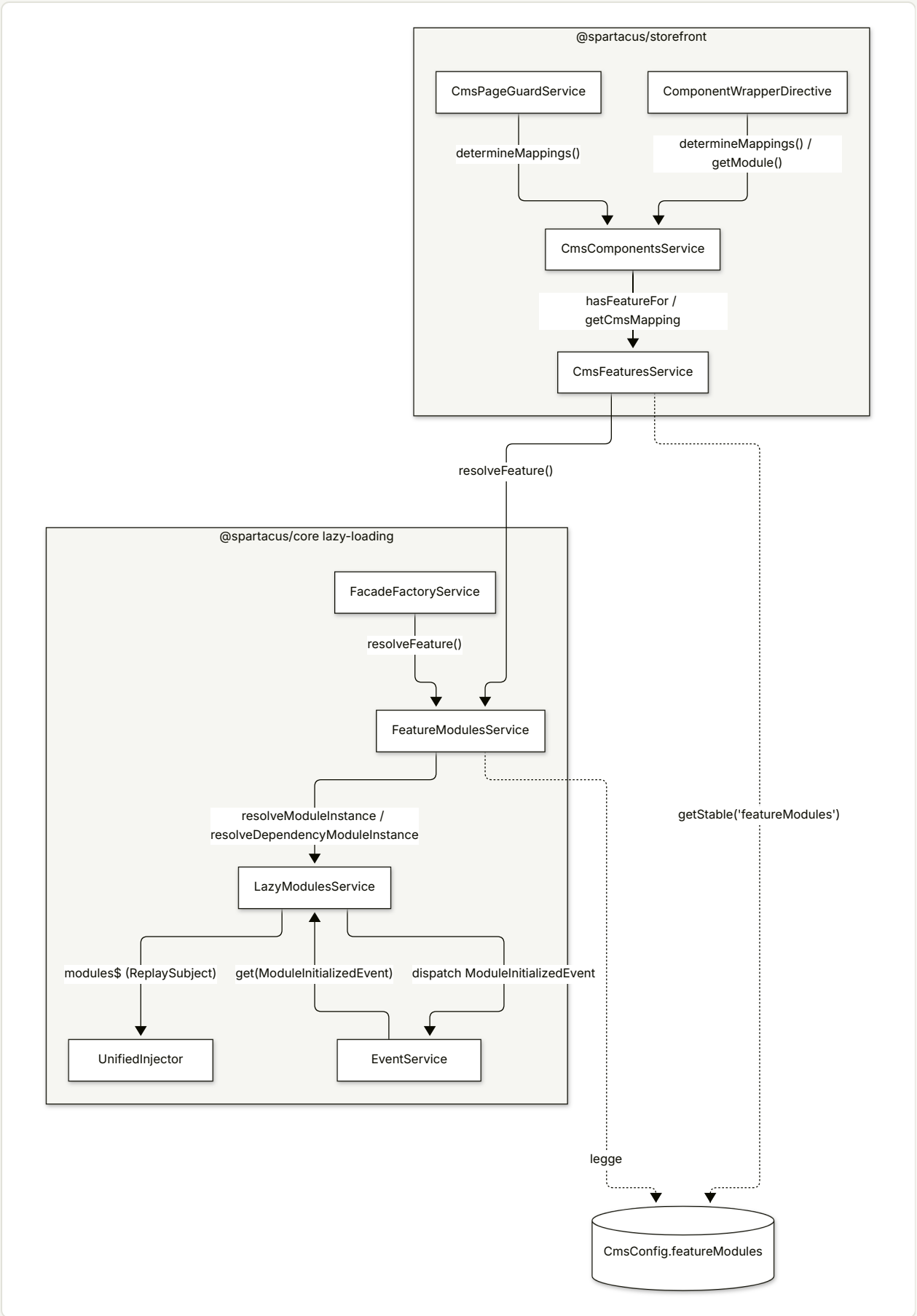
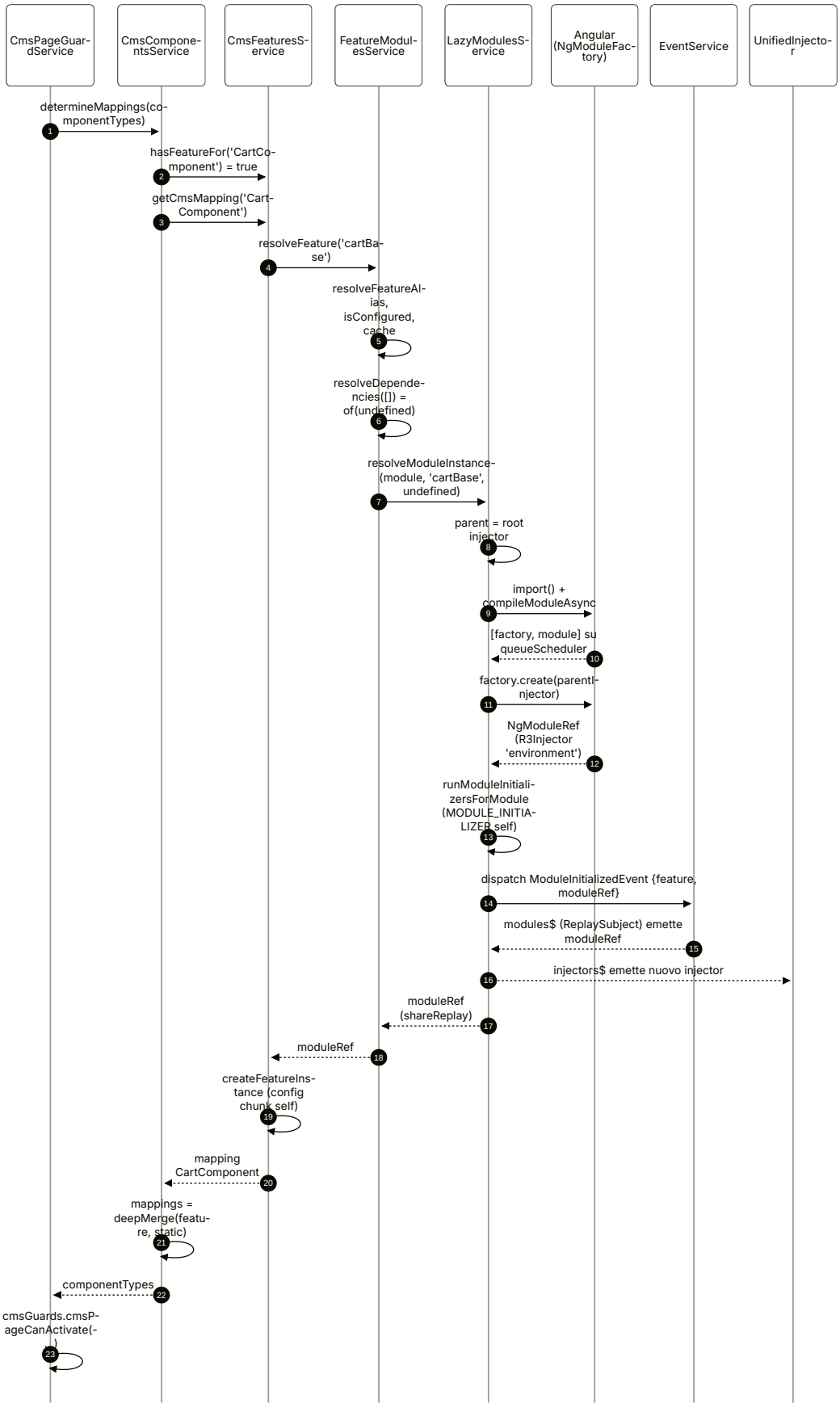


TAVOLA 12.2 **4.2 Navigazione verso la pagina carrello (feature cartBase)**



13

Site context e site theme

Base site, lingua e valuta: da dove arrivano e come si propagano a router, HTTP e UI.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Ogni chiamata OCC verso SAP Commerce Cloud è "situata": appartiene a un **base site** (il segmento `/electronics-spa/` nel path), viene servita in una **lingua** e in una **valuta** (i query param `lang` e `curr`). Lo storefront deve quindi sapere, in ogni momento, qual è la terna attiva `baseSite / language / currency`, da dove arriva (URL, `localStorage`, configurazione, backend) e come propagarla a router, HTTP e UI. È questo il compito del **site context** di `@spartacus/core` (`core-libs/core/src/site-context/`). Accanto a esso vive il **site theme** (`core-libs/core/src/site-theme/`), un quarto parametro di contesto che non arriva al backend ma si traduce in una classe CSS sul componente root.

Gli ID dei parametri sono quattro costanti stringa — `language`, `currency`, `baseSite`, `theme` — core-libs/core/src/site-context/providers/context-ids.ts:L7-L10. Tutto il sottosistema ruota attorno a una sola chiave di configurazione, `context`, un dizionario di array di stringhe con la chiave speciale `urlParameters` — core-libs/core/src/site-context/config/site-context-config.ts:L10-L19. La convenzione fondamentale è: **il default di un parametro è il primo elemento del suo array** — core-libs/core/src/site-context/config/context-config-utils.ts:L28-L34.

Il site context risolve, in sintesi, cinque problemi:

1. **Ottenere la configurazione** `context`: statica (fornita dall'app con `provideConfig`) oppure dinamica, derivata dal base site del backend il cui `urlPatterns` corrisponde all'URL corrente (`SiteContextConfigInitializer`, un `ConfigInitializer` con scope `context`) — core-libs/core/src/site-context/site-context.module.ts:L30-L41, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L24-L27.
2. **Inizializzare i valori attivi** con una precedenza precisa: (federated login →) URL → `localStorage` → default di config — core-libs/core/src/site-context/services/language-initializer.ts:L35-L44, core-libs/core/src/site-context/services/currency-initializer.ts:L33-L41, core-libs/core/src/site-context/services/base-site-initializer.ts:L31-L38.
3. **Codificare il contesto nell'URL** (`/electronics-spa/en/USD/...`) sostituendo l'`UrlSerializer` di Angular con `SiteContextUrlSerializer` e tenendo URL e stato sincronizzati in entrambe le direzioni con `SiteContextRoutesHandler` — core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L18, core-libs/core/src/site-context/services/site-context-routes-handler.ts:L41-L46.
4. **Propagare il contesto alle chiamate OCC**: il base site entra nel path tramite `OccEndpointsService.getBaseUrl()`, lingua e valuta come query param tramite `SiteContextInterceptor` — core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102, core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68.
5. **Notificare i cambi** al resto dell'applicazione: action NgRx `LanguageChange`/`CurrencyChange`, eventi `LanguageSetEvent`/`CurrencySetEvent`, reset del CMS al cambio lingua — core-libs/core/src/site-context/store/effects/languages.effect.ts:L58-L70, core-libs/core/src/site-context/events/site-context-event.builder.ts:L41-L72, core-libs/core/src/cms/store/reducers/index.ts:L78-L87.

Il site theme riusa gli stessi mattoni (store NgRx, persistenza, initializer, interfaccia `SiteContext<T>`), ma con regole proprie: non passa dall'URL, ha un default che può arrivare dal CMS e una lista di temi opzionali (di default due temi ad alto contrasto) — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L36-L50, core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22.

Il capitolo presuppone i meccanismi di configurazione descritti in 11 — Configurazione (`CONFIG_INITIALIZER`, `getStable`, `deepMerge`) e l'ordine di bootstrap di 10 — Bootstrap e ordine di inizializzazione. Il rapporto con il Router di Angular è approfondito in 05 — Il Router di Angular e 14 — Routing.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/site-context/site-context.module.ts	`SiteContextModule.forRoot()` e factory `initSiteContextConfig` (statico vs dinamico)
core-libs/core/src/site-context/config/site-context-config.ts	token astratto `SiteContextConfig` (`useExisting: Config`) + module augmentation di `Config`
core-libs/core/src/site-context/config/context-config-utils.ts	`getContextParameterValues`, `getContextParameterDefault` (default = primo elemento)
core-libs/core/src/site-context/config/default-site-context-config.ts	default: 17 lingue (prima `en`), 20 valute (prima `USD`), nessun `baseSite`
core-libs/core/src/site-context/config/base-site-config-validator.ts	validator che segnala `baseSite` mancante
core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts	`SiteContextConfigInitializer`: config `context` dal base site che matcha l'URL
core-libs/core/src/site-context/providers/context-ids.ts	costanti `LANGUAGE_CONTEXT_ID`, `CURRENCY_CONTEXT_ID`, `BASE_SITE_CONTEXT_ID`, `THEME_CONTEXT_ID`
core-libs/core/src/site-context/providers/context-service-map.ts	`ContextServiceMap` (nome contesto → classe facade) e `serviceMapFactory`
core-libs/core/src/site-context/providers/context-service-providers.ts	facade + `APP_INITIALIZER` che attende `getStable('context')`
core-libs/core/src/site-context/providers/context-initializer-providers.ts	tre `APP_INITIALIZER` (language, currency, baseSite)
core-libs/core/src/site-context/providers/site-context-params-providers.ts	`SiteContextParamsService`, `SiteContextUrlSerializer`, override di `UrlSerializer`
core-libs/core/src/site-context/facade/site-context.interface.ts	interfaccia `SiteContext<T>`
core-libs/core/src/site-context/facade/base-site.service.ts	`BaseSiteService`
core-libs/core/src/site-context/facade/language.service.ts	`LanguageService`
core-libs/core/src/site-context/facade/currency.service.ts	`CurrencyService` (filtro `showOnlyActiveCurrencies`)
core-libs/core/src/site-context/services/site-context-params.service.ts	accesso per nome a parametri, valori e servizi di contesto
core-libs/core/src/site-context/services/site-context-url-serializer.ts	`SiteContextUrlSerializer`: toglie/aggiunge i prefissi di contesto dall'URL
core-libs/core/src/site-context/services/site-context-routes-handler.ts	`SiteContextRoutesHandler` (PRIVATE API): sincronizzazione URL ↔ stato
core-libs/core/src/site-context/services/language-initializer.ts	precedenza lingua: federated login → URL → storage → default

PATH	RUOLO
core-libs/core/src/site-context/services/currency-initializer.ts	precedenza valuta: URL → storage → default
core-libs/core/src/site-context/services/base-site-initializer.ts	precedenza base site: URL → default (nessuno storage)
core-libs/core/src/site-context/services/language-state-persistence.service.ts	sync lingua con localStorage (chiave language)
core-libs/core/src/site-context/services/currency-state-persistence.service.ts	sync valuta con localStorage (chiave currency)
core-libs/core/src/site-context/store/state.ts	feature siteContext: languages, currencies, baseSite
core-libs/core/src/site-context/store/site-context-store.module.ts	StoreModule.forFeature, effects, transfer SSR dello stato
core-libs/core/src/site-context/store/reducers/*.ts	reducer di base site, lingue, valute
core-libs/core/src/site-context/store/effects/*.ts	caricamento liste + emissione delle action *Change
core-libs/core/src/site-context/store/actions/*.ts, store/selectors/*.ts	namespace SiteContextActions, SiteContextSelectors
core-libs/core/src/site-context/events/*.ts	LanguageSetEvent, CurrencySetEvent, SiteContextEventBuilder, SiteContextEventModule
core-libs/core/src/site-context/connectors/*.ts	SiteAdapter, SiteConnector, token normalizer
core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts	binding SiteAdapter → OccSiteAdapter, interceptor, normalizer
core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts	implementazione OCC (basesites, languages, currencies, …)
core-libs/core/src/occ/adapters/site-context/converters/base-site-normalizer.ts	stores[0] → baseStore
core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts	SiteContextInterceptor: query param lang/curr
core-libs/core/src/occ/services/occ-endpoints.service.ts	base site attivo nel path OCC
core-libs/core/src/site-theme/site-theme.module.ts	SiteThemeModule.forRoot()
core-libs/core/src/site-theme/config/site-theme-config.ts, core-libs/core/src/site-theme/config/default-site-theme-config.ts	SiteThemeConfig, temi opzionali di default
core-libs/core/src/site-theme/facade/site-theme.service.ts	SiteThemeService: default, lista temi, validazione
core-libs/core/src/site-theme/services/site-theme-initializer.ts	init tema e precedenza statico > opzionale > CMS
core-libs/core/src/site-theme/services/site-theme-persistence.service.ts	sync tema con localStorage (chiave siteTheme)
core-libs/core/src/site-theme/store/*	feature siteTheme, action, reducer, effect, selector
core-libs/storefront/layout/theme/theme.module.ts, core-libs/storefront/layout/theme/theme.service.ts	applicazione della classe CSS del tema sul root component

3. Come funziona passo per passo

3.1 Registrazione: `SiteContextModule.forRoot()` e `SiteThemeModule.forRoot()`

`BaseCoreModule` importa `ConfigInitializerModule.forRoot()` e, più in basso, `SiteContextModule.forRoot()` con un commento esplicito: deve stare dopo `RouterModule.forRoot` perché sovrascrive `UrlSerializer`; `SiteThemeModule.forRoot()` viene importato ancora dopo — core-libs/core/src/base-core.module.ts:L31-L43. Il motivo è che il router registra `UrlSerializer` con `useClass: DefaultUrlSerializer` in `ROUTER_PROVIDERS` — angular:packages/router/src/router_module.ts:L69-L72 — e vince l'ultimo provider registrato.

`SiteContextModule` importa `StateModule`, `SiteContextStoreModule` e `SiteContextEventModule` — core-libs/core/src/site-context/site-context.module.ts:L43-L46. Il suo `forRoot()` registra — core-libs/core/src/site-context/site-context.module.ts:L47-L65:

- la default config (`provideDefaultConfigFactory(defaultSiteContextConfigFactory)`);
- `contextServiceMapProvider` (la mappa nome → facade);
- `contextServiceProviders`: `BaseSiteService`, `LanguageService`, `CurrencyService` e un `APP_INITIALIZER` che restituisce `lastValueFrom(configInit.getStable('context'))` — core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L33;
- `siteContextParamsProviders`: `SiteContextParamsService`, `SiteContextUrlSerializer` e `{ provide: UrlSerializer, useExisting: SiteContextUrlSerializer }` — core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L18;
- il validator `baseSiteConfigValidator`;
- un `CONFIG_INITIALIZER` con factory `initSiteContextConfig`;
- `contextInitializerProviders`: tre `APP_INITIALIZER` (language, currency, baseSite, in quest'ordine) — core-libs/core/src/site-context/providers/context-initializer-providers.ts:L36-L55.

Le tre facade sono `@Injectable()` senza `providedIn` — core-libs/core/src/site-context/facade/base-site.service.ts:L21-L22 — quindi esistono solo dove è importato `SiteContextModule.forRoot()`.

`SiteThemeModule.forRoot()` fornisce `SiteThemeService`, `defaultSiteThemeConfig` e l'`APP_INITIALIZER` del tema; importa `StateModule` e `SiteThemeStoreModule` — core-libs/core/src/site-theme/site-theme.module.ts:L15-L29.

3.2 Configurazione `context`: statica o dinamica

La factory del `CONFIG_INITIALIZER` decide se caricare la config dal backend: lo fa **solo se non esiste una config statica per** `context.baseSite`, altrimenti restituisce `null` — core-libs/core/src/site-context/site-context.module.ts:L30-L41. `ConfigInitializerService.initialize` salta gli initializer `null` con un `continue` — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L140-L143.

Caso statico (il più comune, e quello della demo app): l'app fornisce `context.urlParameters` e `context.baseSite`. La demo B2C configura `urlParameters: ['baseSite', 'language', 'currency']` e una lista di base site il cui primo elemento è `electronics-spa` (oppure `electronics-epdvisualization-spa` se `environment.epdVisualization` è attivo) — projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L11-L31, projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L21-L23, projects/storefrontapp/src/environments/environment.ts:L32-L32. Lingue e valute, se non fornite dall'app, restano quelle del default: 17 lingue con `en` in testa e 20 valute con `USD` in testa — core-libs/core/src/site-context/config/default-site-context-config.ts:L13-L58.

Caso dinamico: `SiteContextConfigInitializer` è un `ConfigInitializer` in root con `scopes = ['context']` e `configFactory = () => lastValueFrom(this.resolveConfig())` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L24-L27. `resolveConfig()`:

1. carica tutti i base site con `BaseSiteService.getAll()` (che dispatcha `LoadBaseSites` se lo store è vuoto) — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L46-L48, core-libs/core/src/site-context/facade/base-site.service.ts:L41-L51;
2. legge l'URL corrente da `WindowRef.location.href` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L37-L39. In SSR `WindowRef.location` restituisce `href: SERVER_REQUEST_URL` e lancia un errore se il token manca — core-libs/core/src/window/window-ref.ts:L70-L89;
3. se il federated login è attivo chiama `detectContext()` e, sul login domain, usa l'`origin` del federated login al posto dell'URL — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L49-L59;
4. sceglie il primo base site con un `urlPatterns` che matcha l'URL (regex Java convertite con `JavaRegExpConverter.toJsRegExp`); se nessuno matcha lancia `Error: Cannot get base site config! ...` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L61-L74, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L100-L106;
5. traduce il base site in config con `getConfig()` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L77-L98:
   - `urlParameters` da `urlEncodingAttributes`, mappando l'attributo OCC `storefront` sul parametro Spartacus `baseSite` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L108-L119;
   - `baseSite: [uid]`;
   - `language` e `currency` dal `baseStore`, con il default spostato in testa (`getIsoCodes` + `moveToFirst`, su una copia dell'array); la lingua di default privilegia `BaseSite.defaultLanguage` su `baseStore.defaultLanguage` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L82-L85, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L124-L151;
   - `theme: [source.theme]` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L90-L93.

Il `baseStore` esiste perché `BaseSiteNormalizer` lo valorizza con il primo elemento di `stores` (e cancella `stores`) — core-libs/core/src/occ/adapters/site-context/converters/base-site-normalizer.ts:L18-L28; il normalizer è applicato da `OccSiteAdapter.loadBaseSites()`, che chiama l'endpoint `baseSites` (default `basesites?fields=FULL`) **senza** segmento base site — core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L109-L120, core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L9-L25.

Il risultato viene fuso con `deepMerge` sia in `rootConfig` sia nella `config` globale, poi lo scope `context` viene chiuso — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L158-L169. Attenzione: `deepMerge` non fonde gli array (un array, o anche `undefined`, sostituisce il valore di destinazione) — core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/config/utils/deep-merge.ts:L36-L45. Le liste del backend rimpiazzano quindi quelle di default.

Il validator `baseSiteConfigValidator` restituisce un messaggio se manca il default di `baseSite` — core-libs/core/src/site-context/config/base-site-config-validator.ts:L11-L15; come tutti i validator gira solo in dev mode e dopo `getStable()` — core-libs/core/src/config/config-validator/config-validator.module.ts:L23-L37.

3.3 Attesa della config e blocco della navigazione iniziale

Due meccanismi trattengono l'applicazione finché la config `context` non è pronta:

- l'`APP_INITIALIZER` `initializeContext` restituisce una Promise su `getStable('context')` — core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L21. `getStable(...scopes)` emette subito se tutto è stabile, altrimenti attende che gli scope richiesti non siano più in corso — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70;
- `ConfigInitializerModule.forRoot()` registra un location initializer che blocca la navigazione iniziale fino a `getStable()` — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L63.

Al contrario, i tre `APP_INITIALIZER` di language/currency/baseSite chiamano `initialize()` ma **non restituiscono** la Promise — core-libs/core/src/site-context/providers/context-initializer-providers.ts:L12-L34. Angular attende solo gli initializer che restituiscono Promise o Subscribable — angular:packages/core/src/application/application_init.ts:L244-L255 — quindi l'inizializzazione dei valori attivi procede in modo asincrono rispetto al bootstrap. Nota a margine: `APP_INITIALIZER` è deprecato da Angular v19 in favore di `provideAppInitializer` — angular:packages/core/src/application/application_init.ts:L38-L38.

3.4 Facade: `SiteContext<T>`, `BaseSiteService`, `LanguageService`, `CurrencyService`

Tutte le facade implementano `SiteContext<T>`: `getAll()`, `getActive()`, `setActive(isocode)` — core-libs/core/src/site-context/facade/site-context.interface.ts:L9-L13. Le regole comuni:

- `getAll()` **è lazy**: se lo store non ha le entità dispatcha la `Load*` corrispondente e filtra i `null` — core-libs/core/src/site-context/facade/base-site.service.ts:L41-L51, core-libs/core/src/site-context/facade/language.service.ts:L34-L44. `CurrencyService.getAll()` filtra inoltre le valute con `active` quando il toggle `showOnlyActiveCurrencies` è attivo — core-libs/core/src/site-context/facade/currency.service.ts:L37-L52.
- `getActive()` filtra i valori "non ancora impostati": `BaseSiteService` usa `Boolean(active)` perché lo stato iniziale è `''` — core-libs/core/src/site-context/facade/base-site.service.ts:L31-L36, core-libs/core/src/site-context/store/reducers/base-site.reducer.ts:L11-L15.
- `setActive()` dispatcha `SetActive*` solo se il valore è diverso dall'attuale; lingua e valuta verificano inoltre `isValid` (appartenenza a `context.language` / `context.currency`) — core-libs/core/src/site-context/facade/language.service.ts:L59-L69, core-libs/core/src/site-context/facade/language.service.ts:L88-L95, core-libs/core/src/site-context/facade/currency.service.ts:L67-L77, core-libs/core/src/site-context/facade/currency.service.ts:L96-L103. `BaseSiteService.setActive` invece **non** chiama `isValid`, che pure esiste come metodo `protected` — core-libs/core/src/site-context/facade/base-site.service.ts:L72-L82, core-libs/core/src/site-context/facade/base-site.service.ts:L101-L108.
- `isInitialized()` verifica in modo sincrono se `getActive()` ha già emesso (subscribe + unsubscribe immediato) — core-libs/core/src/site-context/facade/base-site.service.ts:L87-L94. È il predicato su cui si basa tutta la precedenza degli initializer.
- `BaseSiteService.get(siteUid?)` cerca il sito nella lista; senza uid usa il base site attivo — core-libs/core/src/site-context/facade/base-site.service.ts:L56-L70.

`ContextServiceMap` è una classe astratta indicizzata per nome; la factory mappa solo `language`, `currency` e `baseSite` — **non** `theme` — core-libs/core/src/site-context/providers/context-service-map.ts:L18-L33.

3.5 `SiteContextParamsService`: il contesto "per nome"

`SiteContextParamsService` fa da ponte generico fra nomi di parametro e facade, ed è usato da serializer e routes handler:

- `getContextParameters()` → tutte le chiavi di `context` tranne `urlParameters` — core-libs/core/src/site-context/services/site-context-params.service.ts:L29-L36;
- `getUrlEncodingParameters()` → `context.urlParameters` o `[]` — core-libs/core/src/site-context/services/site-context-params.service.ts:L38-L40;
- `getParamValues()` / `getParamDefaultValue()` → le utility di `context-config-utils` — core-libs/core/src/site-context/services/site-context-params.service.ts:L42-L48;
- `getSiteContextService(param)` → risolve la classe dal `ContextServiceMap` tramite `Injector`; se l'injection fallisce logga un warning in dev mode e restituisce `undefined` — core-libs/core/src/site-context/services/site-context-params.service.ts:L50-L63;
- `getValue(param)` → valore attivo letto sincronicamente, con fallback al default di config — core-libs/core/src/site-context/services/site-context-params.service.ts:L65-L77;
- `setValue(param, value)` → `service.setActive(value)` — core-libs/core/src/site-context/services/site-context-params.service.ts:L79-L84;
- `getValues(params)` → `combineLatest` dei valori attivi, emette solo quando tutti sono truthy — core-libs/core/src/site-context/services/site-context-params.service.ts:L93-L107.

3.6 Il contesto nell'URL: `SiteContextUrlSerializer`

`SiteContextUrlSerializer` estende `DefaultUrlSerializer` — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L30-L35. Il `Router` lo usa automaticamente perché inietta `UrlSerializer` e lo invoca in `parseUrl`/`serializeUrl` — angular:packages/router/src/router.ts:L105-L105, angular:packages/router/src/router.ts:L588-L595; anche `router.url` è `serializeUrl(currentUrlTree)` — angular:packages/router/src/router.ts:L351-L353.

Parse (`parse(url)`): se `urlParameters` non è vuoto estrae i prefissi di contesto, fa il parse del resto con `super.parse` e salva i parametri nella proprietà custom `siteContext` del `UrlTree` (tipo `UrlTreeWithSiteContext`); altrimenti delega del tutto a `super.parse` — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L62-L71, core-libs/core/src/site-context/services/site-context-url-serializer.ts:L14-L24. L'estrazione vera e propria (`urlExtractContextParameters`) separa path e query/fragment con `URL_SPLIT`, scarta il segmento vuoto iniziale — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L83-L88 — e poi scorre in parallelo i nomi di `urlParameters` e i segmenti: un segmento viene consumato solo se è tra i valori ammessi del parametro corrente, mentre il parametro avanza comunque — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L91-L108.

Esempio con la config della demo (`urlParameters: ['baseSite','language','currency']`): l'URL `/electronics-spa/de/EUR/product/123` produce `params = { baseSite: 'electronics-spa', language: 'de', currency: 'EUR' }` e il router vede `product/123`. Un URL `/de/product/123` produce `{ language: 'de' }`: `de` non è un base site, quindi il parametro `baseSite` viene saltato e `de` confrontato con `language`.

Serialize (`serialize(tree)`): serializza con `super.serialize` e antepone, per ogni nome in `urlParameters`, il valore salvato nel tree oppure il valore **attivo** corrente (`getValue`, che a sua volta ripiega sul default di config) — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L126-L131, core-libs/core/src/site-context/services/site-context-url-serializer.ts:L146-L159. Ogni `routerLink` generato dall'app ottiene così il prefisso completo senza che i componenti debbano conoscerlo.

3.7 Sincronizzazione bidirezionale: `SiteContextRoutesHandler`

`SiteContextRoutesHandler` è in root, marcato `// PRIVATE API` e non riesportato dal barrel dei servizi — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L21-L25, core-libs/core/src/site-

context/services/index.ts:L7-L13. `init()` è deprecato dalla 2211.32 e delega a `initOnce()` — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L48-L56. `initOnce()` è idempotente: esegue `initInternal()` una sola volta e restituisce sempre lo stesso `ReplaySubject(1)` — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L62-L84. Questo permette ai tre initializer di "chiamarlo" tutti senza ripetere il lavoro.

`initInternal()` — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L89-L105:

1. risolve `Router` e `Location` dall'`Injector` (lazy, per non creare dipendenze circolari al bootstrap);
2. se ci sono `urlParameters`, legge sincronicamente il path corrente (`location.path(true)`) e applica i valori (`setContextParamsFromRoute` → `urlExtractContextParameters` → `SiteContextParamsService.setValue`) — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L163-L168;
3. attiva le due subscription `subscribeChanges` e `subscribeRouting`;
4. in ogni caso emette e completa `initialized$`.

URL → stato (`subscribeRouting`): sugli eventi `NavigationStart/End/Error/Cancel` imposta `isNavigating = event instanceof NavigationStart` e, a ogni `NavigationStart`, applica i parametri estratti dall'URL di destinazione — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L137-L156.

Stato → URL (`subscribeChanges`): per ogni parametro URL con un servizio, sottoscrive `getActive()`; se il valore cambia **fuori** da una navigazione e c'era già un valore precedente memorizzato, ricostruisce l'URL corrente con `router.parseUrl(router.url)` + `router.serializeUrl(...)` e lo sostituisce con `location.replaceState`, senza navigare — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L111-L131. Il flag `isNavigating` esiste proprio per evitare il ciclo infinito fra le due subscription — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L41-L46.

> DA VERIFICARE — Perché `replaceState` produca il nuovo prefisso, il `currentUrlTree` del router non deve portare con sé la proprietà custom `siteContext` con i vecchi valori (altrimenti `serialize` userebbe quelli invece del valore attivo). Il codice Angular assegna `currentUrlTree` dall'URL finale della navigazione (angular:packages/router/src/statemanager/state_manager.ts:L108-L108), ma non è stato verificato a runtime se quell'oggetto è lo stesso restituito da `SiteContextUrlSerializer.parse`.

3.8 Inizializzazione dei valori attivi e persistenza

Ogni initializer è una catena `switchMap` che parte da `getStable('context')`; ogni passo applica un valore solo se il precedente non l'ha già fatto (tranne URL e federated login, che chiamano `setActive` direttamente):

- **Lingua** — core-libs/core/src/site-context/services/language-initializer.ts:L35-L44:
  1. `setFromFederatedLoginContext()` → `setActive(federatedLoginService.language)` se abilitato e presente — core-libs/core/src/site-context/services/language-initializer.ts:L73-L79;
  2. `siteContextRoutesHandler.initOnce()` → valori dall'URL;
  3. `languageStatePersistenceService.initSync()` → `localStorage`;
  4. `setFallbackValue()` → primo elemento di `context.language` se non ancora inizializzata — core-libs/core/src/site-context/services/language-initializer.ts:L51-L71.
- **Valuta** — stessa catena senza federated login — core-libs/core/src/site-context/services/currency-initializer.ts:L33-L41, fallback core-libs/core/src/site-context/services/currency-initializer.ts:L60-L68.
- **Base site** — config stabile → URL → fallback, **senza** persistenza su storage — core-libs/core/src/site-context/services/base-site-initializer.ts:L31-L38, core-libs/core/src/site-context/services/base-site-initializer.ts:L57-L65.

Poiché `setContextParamsFromRoute` passa per `setValue` → `setActive` senza controllare `isInitialized()` (core-libs/core/src/site-context/services/site-context-routes-handler.ts:L163-L168, core-libs/core/src/site-context/services/site-context-params.service.ts:L79-L84, core-libs/core/src/site-context/facade/language.service.ts:L59-L69), un valore valido nell'URL **sovrascrive** quello impostato dal federated login.

La persistenza (`LanguageStatePersistenceService`, `CurrencyStatePersistenceService`) usa `StatePersistenceService.syncWithStorage` con chiave `language` / `currency`, `state$ = service.getActive()` e un `onRead` che applica il valore dallo storage solo se non già inizializzato, poi completa `initialized$` — core-libs/core/src/site-context/services/language-state-persistence.service.ts:L22-L47, core-libs/core/src/site-context/services/currency-state-persistence.service.ts:L24-L42. `syncWithStorage` usa per default `localStorage` e `context$ = of('')`; la lettura avviene prima della scrittura e ogni emissione di `state$` viene persistita — core-libs/core/src/state/services/state-persistence.service.ts:L44-L92. La chiave effettiva è `spartacus⍰<context>⍰<key>`, cioè ad esempio `spartacus⍰⍰language` — core-libs/core/src/state/services/state-persistence.service.ts:L124-L131. In SSR `WindowRef.localStorage` è `undefined`, quindi `onRead` riceve `undefined` e non si scrive nulla — core-libs/core/src/window/window-ref.ts:L41-L44, core-libs/core/src/window/window-ref.ts:L58-L60, core-libs/core/src/state/services/state-persistence.service.ts:L68-L75.

3.9 Store NgRx, effects ed eventi

La feature si chiama `siteContext` e contiene `languages`, `currencies`, `baseSite` — core-libs/core/src/site-context/store/state.ts:L8-L18. `BaseSiteState` ha `entities`, `activeSite` e `details` (il `BaseSite` attivo) — core-libs/core/src/site-context/store/state.ts:L37-L41. `SiteContextStoreModule` registra feature ed effects e una default config che trasferisce lo slice `siteContext` dal server al browser via `TRANSFER_STATE` — core-libs/core/src/site-context/store/site-context-store.module.ts:L20-L43. La chiave `state.ssrTransfer.keys` è letta dal meta-reducer `getTransferStateReducer` (registrato su `META_REDUCERS`): sul server scrive a ogni action lo slice in `TransferState` sotto `CX_KEY`; sul browser, all'action `INIT` e solo se l'utente non è loggato, lo fonde nello stato con `deepMerge` — core-libs/core/src/state/reducers/transfer-state.reducer.ts:L18-L40, core-libs/core/src/state/reducers/transfer-state.reducer.ts:L51-L60, core-libs/core/src/state/reducers/transfer-state.reducer.ts:L74-L92, core-libs/core/src/state/reducers/index.ts:L27-L43. I dettagli sono in 19 — State management e 39 — SSR ottimizzato.

Reducer:

- `SET_ACTIVE_BASE_SITE` aggiorna `activeSite` e ricalcola `details` dalle entità se presenti — core-libs/core/src/site-context/store/reducers/base-site.reducer.ts:L29-L41;
- `LOAD_BASE_SITES_SUCCESS` indicizza per `uid` e popola `details` del sito attivo — core-libs/core/src/site-context/store/reducers/base-site.reducer.ts:L43-L65;
- lingue e valute partono da `entities: null`, attivo `null`; `LOAD_*_SUCCESS` indicizza per `isocode`, `SET_ACTIVE_*` imposta l'isocode — core-libs/core/src/site-context/store/reducers/languages.reducer.ts:L11-L51, core-libs/core/src/site-context/store/reducers/currencies.reducer.ts:L11-L52.

Effects:

- `loadLanguages$`, `loadCurrencies$`, `loadBaseSite$`, `loadBaseSites$` usano `exhaustMap` sul `SiteConnector` e normalizzano gli errori con `tryNormalizeHttpError` — core-libs/core/src/site-context/store/effects/languages.effect.ts:L34-L56, core-libs/core/src/site-context/store/effects/currencies.effect.ts:L29-L51, core-libs/core/src/site-context/store/effects/base-site.effect.ts:L20-L68;
- `activateLanguage$` / `activateCurrency$` osservano il selector dell'attivo con `bufferCount(2, 1)` e dispatchano `LanguageChange` / `CurrencyChange` con `{ previous, current }` solo se esiste un valore precedente

— core-libs/core/src/site-context/store/effects/languages.effect.ts:L58-L70, core-libs/core/src/site-context/store/effects/currencies.effect.ts:L29-L65;

- `reloadPageOnLanguageChange$` (non-dispatching) fa `location.reload()` su `LANGUAGE_CHANGE` se `reloadOnLanguageChange === true` e la piattaforma è browser — core-libs/core/src/site-context/store/effects/languages.effect.ts:L72-L88;
- non esiste un effect "activate" per il base site — core-libs/core/src/site-context/store/effects/base-site.effect.ts:L16-L74.

Eventi: `SiteContextEventBuilder` mappa `SET_ACTIVE_LANGUAGE` → `LanguageSetEvent` e `SET_ACTIVE_CURRENCY` → `CurrencySetEvent` registrandoli su `EventService` — core-libs/core/src/site-context/events/site-context-event.builder.ts:L41-L72; `SiteContextEventModule` esiste solo per istanziare eagerly il builder — core-libs/core/src/site-context/events/site-context-event.module.ts:L10-L15. Per il sistema eventi in generale vedi 20 — Facade, command & query, eventi.

Consumatori del cambio lingua fuori dal site context: il meta-reducer CMS azzera l'intero stato CMS su `LANGUAGE_CHANGE` — core-libs/core/src/cms/store/reducers/index.ts:L78-L87 — e `PageEffects` ricarica la pagina CMS corrente — core-libs/core/src/cms/store/effects/page.effect.ts:L36-L42 (vedi 15 — CMS core).

3.10 Propagazione alle chiamate OCC

`SiteContextOccModule` (importato da `BaseOccModule`) lega `SiteAdapter` a `OccSiteAdapter`, registra `SiteContextInterceptor` su `HTTP_INTERCEPTORS` e `BaseSiteNormalizer` su `BASE_SITE_NORMALIZER` — core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts:L18-L38, core-libs/core/src/occ/base-occ.module.ts:L18-L18. `SiteConnector` delega 1:1 all'adapter — core-libs/core/src/site-context/connectors/site.connector.ts:L19-L31.

- **Base site nel path**: `OccEndpointsService` inietta `BaseSiteService` come `@Optional()`, memorizza l'attivo e in sua assenza usa il default di `context.baseSite` — core-libs/core/src/occ/services/occ-endpoints.service.ts:L34-L54; `getBaseUrl()` compone `baseUrl + prefix + baseSite` — core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102. `urlPathJoin` scarta le parti falsy — core-libs/core/src/occ/utils/occ-url-util.ts:L12-L26.
- **Lingua e valuta in query**: `SiteContextInterceptor` parte dai default di config e si aggiorna sottoscrivendo `getActive()` di lingua e valuta — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L25-L52; aggiunge `lang` e `curr` solo alle richieste il cui URL contiene `occEndpoints.getBaseUrl()` — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68.

L'architettura adapter/connector/normalizer è trattata in 21 — Livello OCC; gli interceptor in 06 — HttpClient e interceptor.

3.11 Site theme

Configurazione. `SiteThemeConfig` (token con `useExisting: Config`) espone `siteTheme.optionalThemes: SiteTheme[]` — core-libs/core/src/site-theme/config/site-theme-config.ts:L11-L23; un `SiteTheme` è `{ i18nNameKey, className }` — core-libs/core/src/model/misc.model.ts:L140-L143. Il default contiene due temi ad alto contrasto, `cx-theme-high-contrast-dark` e `cx-theme-high-contrast-light` — core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22. Il tema **di default** non sta in `siteTheme` ma in `context.theme`: può venire dal CMS (via `SiteContextConfigInitializer`), mentre i temi opzionali si configurano solo in `siteTheme.optionalThemes` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L90-L93.

Facade. `SiteThemeService` (`@Injectable()` senza root) implementa `SiteContext<SiteTheme>` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L27-L32:

- `getDefault()` → `{ className: getDefaultClassName(), i18nNameKey: 'siteThemeSwitcher.themes.default' }`, con fallback `''` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L37-L44;
- `getDefaultClassName()` → `context.theme[0]` se presente; altrimenti, con il toggle `applyBaseSiteThemeFromCms`, il `theme` del base site attivo letto in modo sincrono (`getLastValueSync`); altrimenti `''` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L51-L71;
- la lista dei temi è `[default, ...optionalThemes]` e `getAll()` restituisce `of(themes)` solo per conformità all'interfaccia — core-libs/core/src/site-theme/facade/site-theme.service.ts:L84-L94;
- `setActive(className)` dispatcha `SetActiveSiteTheme` se diverso e valido — core-libs/core/src/site-theme/facade/site-theme.service.ts:L109-L119; con il toggle CMS ogni stringa non null/undefined è valida (anche `''`), senza toggle il valore deve essere un `className` della lista — core-libs/core/src/site-theme/facade/site-theme.service.ts:L130-L137.

Store. Feature `siteTheme` con un solo sotto-stato `themes.activeSiteTheme` — core-libs/core/src/site-theme/store/state.ts:L9-L25; action `SET_ACTIVE_SITE_THEME` e `SITE_THEME_CHANGE` — core-libs/core/src/site-theme/store/actions/site-themes.action.ts:L9-L25; il reducer gestisce solo `SET_ACTIVE_SITE_THEME` — core-libs/core/src/site-theme/store/reducers/site-themes.reducer.ts:L10-L28; `activateSiteTheme$` emette `SiteThemeChange` con la tecnica `bufferCount(2, 1)` — core-libs/core/src/site-theme/store/effects/site-themes.effect.ts:L21-L31; lo slice è trasferito da SSR via `TRANSFER_STATE` — core-libs/core/src/site-theme/store/site-theme-store.module.ts:L20-L42.

Inizializzazione. L'`APP_INITIALIZER` del tema usa `inject()` nella factory e non restituisce nulla da attendere — core-libs/core/src/site-theme/providers/site-theme-initializer-providers.ts:L10-L23. `SiteThemeInitializer.initialize()` — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L36-L50:

1. config stabile → `SiteThemePersistenceService.initSync()` (chiave `siteTheme`, `onRead` applica solo valori truthy e solo se non inizializzato) — core-libs/core/src/site-theme/services/site-theme-persistence.service.ts:L26-L43, core-libs/core/src/site-theme/providers/site-theme-id.ts:L7-L7;
2. fallback: se non inizializzato imposta `getDefault().className`; se invece la persistenza ha ripristinato un tema e il toggle CMS è attivo, un tema statico (`context.theme[0]`) vince su un tema persistito diverso, salvo che quest'ultimo sia un tema opzionale scelto dall'utente — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L67-L91;
3. con il toggle CMS, una seconda subscription reagisce ai cambi di base site (`baseSiteService.get()`, filtrando siti senza tema e temi uguali) applicando la precedenza **statico > opzionale scelto dall'utente > CMS** — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L97-L127, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L141-L147.

Il tema non passa da `SiteContextRoutesHandler`, quindi non dipende dall'URL.

Applicazione nel DOM. `ThemeModule` (importato da `LayoutModule`, pubblico in `@spartacus/storefront`) registra un `APP_BOOTSTRAP_LISTENER` che passa il componente root a `ThemeService.init` — core-libs/storefront/layout/theme/theme.module.ts:L10-L25, core-libs/storefront/layout/layout.module.ts:L15-L21, core-libs/storefront/layout/theme/index.ts:L7-L8. `ThemeService` sottoscrive `SiteThemeService.getActive()` e, in `setTheme`, rimuove la classe precedente e aggiunge quella nuova sull'elemento host del root component tramite `Renderer2` — core-libs/storefront/layout/theme/theme.service.ts:L35-L61. Le classi CSS dei temi sono materia di 38 — Stili e theming.

3.12 UI: selettori di contesto e di tema

Il componente selettore di contesto (lingua/valuta) risolve il servizio tramite `ContextServiceMap` + `Injector` e chiama `setActive` — core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L72-L78, core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L120-L125. Lo switcher dei temi delega a `SiteThemeService.getAll/getActive/setActive` — core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.component.service.ts:L13-L24. I componenti CMS sono descritti in 45 — Componenti CMS e shared components.

---

4. Diagrammi

4.1 Risoluzione della config `context` al bootstrap

■ **TAVOLA 13.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/base-core.module.ts:L38-L38, core-libs/core/src/site-context/site-context.module.ts:L30-L41, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L46-L98, core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L109-L120, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L158-L169, core-libs/core/src/site-context/providers/context-service-providers.ts:L15-L21, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L63.

4.2 Precedenza nell'inizializzazione della lingua

■ **TAVOLA 13.2** diagramma a tutta pagina, in fondo al capitolo

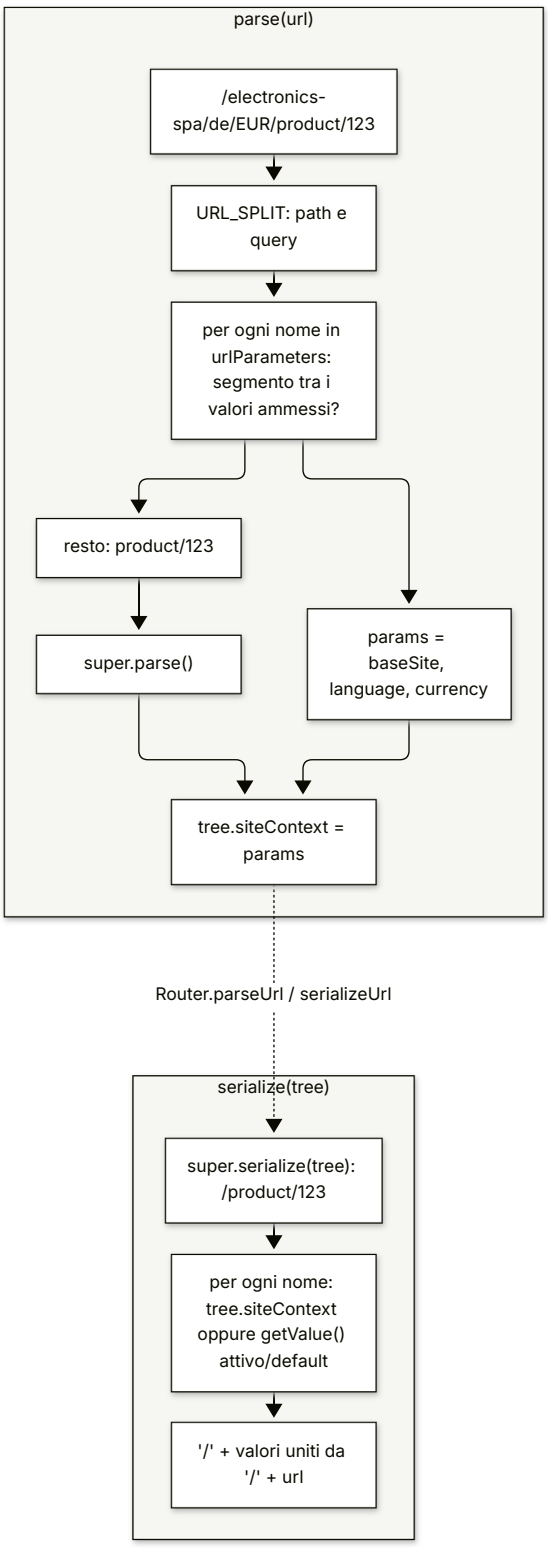
Riferimenti: core-libs/core/src/site-context/providers/context-initializer-providers.ts:L20-L26, core-libs/core/src/site-context/services/language-initializer.ts:L35-L79, core-libs/core/src/site-context/services/site-context-routes-handler.ts:L78-L105, core-libs/core/src/site-context/services/language-state-persistence.service.ts:L29-L47.

4.3 Cambio lingua dalla UI

■ **TAVOLA 13.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L72-L78, core-libs/core/src/site-context/facade/language.service.ts:L59-L69, core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L45-L52, core-libs/core/src/site-context/services/site-context-routes-handler.ts:L111-L131, core-libs/core/src/site-context/store/effects/languages.effect.ts:L58-L88, core-libs/core/src/cms/store/reducers/index.ts:L78-L87, core-libs/core/src/cms/store/effects/page.effect.ts:L36-L42.

4.4 Parse e serialize dell'URL con prefissi di contesto



Riferimenti: core-libs/core/src/site-context/services/site-context-url-serializer.ts:L62-L159, core-libs/core/src/site-context/services/site-context-params.service.ts:L65-L77, angular:packages/router/src/router.ts:L588-L595.

4.5 Precedenza del tema

■ **TAVOLA 13.4** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/site-theme/services/site-theme-initializer.ts:L36-L147, core-libs/core/src/site-theme/services/site-theme-persistence.service.ts:L26-L43, core-libs/storefront/layout/theme/theme.service.ts:L35-L61.

---

5. Estratti di codice reale

5.1 Statico vs dinamico

core-libs/core/src/site-context/site-context.module.ts:L30-L41

```ts
export function initSiteContextConfig(
  configInitializer: SiteContextConfigInitializer,
  config: SiteContextConfig
): ConfigInitializer | null {
  /**
   * Load config for `context` from backend only when there is no static config for `context.baseSite`
   */
  if (!config.context?.[BASE_SITE_CONTEXT_ID]) {
    return configInitializer;
  }
  return null;
}
```

5.2 Dal base site OCC alla config `context`

core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L77-L98

```ts
  protected getConfig(source: BaseSite): SiteContextConfig {
    const result = {
      context: {
        urlParameters: this.getUrlParams(source.urlEncodingAttributes),
        [BASE_SITE_CONTEXT_ID]: [source.uid],
        [LANGUAGE_CONTEXT_ID]: this.getIsoCodes(
          source.baseStore?.languages,
          source.defaultLanguage || source.baseStore?.defaultLanguage
        ),
        [CURRENCY_CONTEXT_ID]: this.getIsoCodes(
          source.baseStore?.currencies,
          source.baseStore?.defaultCurrency
        ),
        // Note: The default Site Theme can be driven by the CMS, but additional possible
        // selectable themes (like a11y high contrast themes) can be configured only
```

```
      // in the separate Spartacus configuration `config.siteTheme.optionalThemes`.
      [THEME_CONTEXT_ID]: [source.theme],
    },
  } as SiteContextConfig;

  return result;
}
```

5.3 Estrazione dei prefissi dall'URL

core-libs/core/src/site-context/services/site-context-url-serializer.ts:L91-L108

```ts
  let paramId = 0;
  let segmentId = 0;
  while (
    paramId < this.urlEncodingParameters.length &&
    segmentId < segments.length
  ) {
    const paramName = this.urlEncodingParameters[paramId];
    const paramValues = this.siteContextParams.getParamValues(paramName);

    if (paramValues.includes(segments[segmentId])) {
      params[paramName] = segments[segmentId];
      segmentId++;
    }
    paramId++;
  }

  url = segments.slice(segmentId).join('/') + queryPart;
  return { url, params };
```

5.4 Stato → URL senza navigazione

core-libs/core/src/site-context/services/site-context-routes-handler.ts:L111-L131

```ts
private subscribeChanges(params: string[]) {
  params.forEach((param) => {
    const service = this.siteContextParams.getSiteContextService(param);
    if (service) {
      this.subscription.add(
        service.getActive().subscribe((value) => {
          if (
            !this.isNavigating &&
            this.contextValues[param] &&
            this.contextValues[param] !== value
          ) {
            const parsed = this.router.parseUrl(this.router.url);
```

```
            const serialized = this.router.serializeUrl(parsed);
            this.location.replaceState(serialized);
          }
          this.contextValues[param] = value;
        })
      );
    }
  });
}
```

5.5 Catena di inizializzazione della valuta

core-libs/core/src/site-context/services/currency-initializer.ts:L33-L41

```ts
initialize(): Promise<unknown> {
  return firstValueFrom(
    this.configInit.getStable('context').pipe(
      switchMap(() => this.siteContextRoutesHandler.initOnce()),
      switchMap(() => this.currencyStatePersistenceService.initSync()),
      switchMap(() => this.setFallbackValue())
    )
  );
}
```

5.6 Interceptor `lang`/`curr`

core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68

```ts
intercept(
  request: HttpRequest<any>,
  next: HttpHandler
): Observable<HttpEvent<any>> {
  if (request.url.includes(this.occEndpoints.getBaseUrl())) {
    request = request.clone({
      setParams: {
        lang: this.activeLang ?? '',
        curr: this.activeCurr ?? '',
      },
    });
  }

  return next.handle(request);
}
```

5.7 Action `*Change` solo sui cambi reali

core-libs/core/src/site-context/store/effects/languages.effect.ts:L58-L70

```ts
activateLanguage$: Observable<SiteContextActions.LanguageChange> =
  createEffect(() =>
    this.state.select(getActiveLanguage).pipe(
      bufferCount(2, 1),

      // avoid dispatching `change` action when we're just setting the initial value:
      filter(([previous]) => !!previous),
      map(
        ([previous, current]) =>
          new SiteContextActions.LanguageChange({ previous, current })
      )
    )
  );
```

5.8 Default del tema

core-libs/core/src/site-theme/facade/site-theme.service.ts:L51-L63

```ts
protected getDefaultClassName(): string {
  const staticDefault = getContextParameterDefault(
    this.config,
    THEME_CONTEXT_ID
  );
  if (staticDefault) {
    return staticDefault;
  }
  if (this.featureToggles.applyBaseSiteThemeFromCms) {
    return this.readActiveBaseSiteTheme() ?? this.FALLBACK_DEFAULT_THEME_ID;
  }
  return this.FALLBACK_DEFAULT_THEME_ID;
}
```

5.9 Classe CSS sul root component

core-libs/storefront/layout/theme/theme.service.ts:L52-L61

```ts
setTheme(theme: string | undefined): void {
  const element = this.rootComponent.location.nativeElement;
  if (this.existingTheme) {
    this.renderer.removeClass(element, this.existingTheme);
  }
  if (theme) {
    this.renderer.addClass(element, theme);
    this.existingTheme = theme;
  }
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
context.baseSite	string[]	nessuno (il validator avvisa in dev mode)	core-libs/core/src/site-context/config/site-context-config.ts:L15-L18	core-libs/core/src/site-context/site-context.module.ts:L37-L37, core-libs/core/src/site-context/services/base-site-initializer.ts:L57-L65, core-libs/core/src/occ/services/occ-endpoints.service.ts:L37-L42, core-libs/core/src/site-context/config/base-site-config-validator.ts:L11-L15
context.language	string[]	17 codici, primo `en`	core-libs/core/src/site-context/config/default-site-context-config.ts:L16-L34	core-libs/core/src/site-context/facade/language.service.ts:L88-L95, core-libs/core/src/site-context/services/language-initializer.ts:L63-L71, core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L36-L39
context.currency	string[]	20 codici, primo `USD`	core-libs/core/src/site-context/config/default-site-context-config.ts:L35-L56	core-libs/core/src/site-context/facade/currency.service.ts:L96-L103, core-libs/core/src/site-context/services/currency-initializer.ts:L60-L68, core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L40-L43
context.theme	string[]	nessuno (fallback `''`)	dinamico: core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L93-L93	core-libs/core/src/site-theme/facade/site-theme.service.ts:L51-L58, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L78-L81, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L130-L135
context.urlParameters	string[]	`[]`	core-libs/core/src/site-context/config/site-context-config.ts:L16-L16	core-libs/core/src/site-context/services/site-context-params.service.ts:L38-L40, core-libs/core/src/site-context/services/site-context-url-serializer.ts:L40-L42, core-libs/core/src/site-context/services/site-context-routes-handler.ts:L92-L92
context.<altro>	string[]	—	core-libs/core/src/site-context/config/site-context-config.ts:L17-L17	core-libs/core/src/site-context/services/site-context-params.service.ts:L29-L36
siteTheme.optionalThemes	SiteTheme[]	2 temi high-contrast	core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22	core-libs/core/src/site-theme/facade/site-theme.service.ts:L86-L89, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L145-L146
state.ssrTransfer.keys.siteContext	StateTransferType	TRANSFER_STATE	core-libs/core/src/site-context/store/site-context-store.module.ts:L20-L30	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L24-L36
state.ssrTransfer.keys.siteTheme	StateTransferType	TRANSFER_STATE	core-libs/core/src/site-theme/store/site-theme-store.module.ts:L20-L29	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L24-L36

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.endpoints.baseSites	string	'basesites?fields=FULL'	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L21-L21	core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L100-L100, core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L114-L114
backend.occ.endpoints.languages / currencies	string	'languages' / 'currencies'	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L13-L14	core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L43-L43, core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L52-L52
feature toggle showOnlyActiveCurrencies	boolean	true	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L306-L311, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1051-L1051	core-libs/core/src/site-context/facade/currency.service.ts:L47-L49
feature toggle applyBaseSiteThemeFromCms	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L655-L670, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1092-L1092	core-libs/core/src/site-theme/facade/site-theme.service.ts:L59-L59, core-libs/core/src/site-theme/facade/site-theme.service.ts:L131-L131, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L47-L47, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L77-L77
feature toggle reloadOnLanguageChange	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L700-L710, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1096-L1096	core-libs/core/src/site-context/store/effects/languages.effect.ts:L78-L78
chiave storage lingua	string	spartacus⁃⁃⁃language	core-libs/core/src/site-context/services/language-state-persistence.service.ts:L31-L31, core-libs/core/src/state/services/state-persistence.service.ts:L124-L131	idem
chiave storage valuta	string	spartacus⁃⁃⁃currency	core-libs/core/src/site-context/services/currency-state-persistence.service.ts:L26-L26	idem
chiave storage tema	string	spartacus⁃⁃⁃siteTheme	core-libs/core/src/site-theme/providers/site-theme-id.ts:L7-L7, core-libs/core/src/site-theme/services/site-theme-persistence.service.ts:L28-L28	idem

La demo app attiva showOnlyActiveCurrencies, applyBaseSiteThemeFromCms e reloadOnLanguageChange — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L340-L340, projects/

storefrontapp/src/app/spartacus/spartacus-features.module.ts:L379-L382. Per il funzionamento generale dei feature toggle vedi 42 — Feature toggle e deprecazioni; per le chiavi di config in generale l'Appendice A02.

---

7. Commenti e note notevoli dal sorgente

- `// should be imported after RouterModule.forRoot, because it overwrites UrlSerializer` — core-libs/core/src/base-core.module.ts:L38-L38. L'ordine degli import è parte del contratto.
- `Load config for \context` from backend only when there is no static config for `context.baseSite`–core-libs/core/src/site-context/site-context.module.ts:L34-L36`. Una sola chiave decide fra due modalità di funzionamento.
- `// functions below should not be exposed in public API:` — core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L12; e `// PRIVATE API` su `SiteContextRoutesHandler` — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L21-L21.
- `@deprecated since 2211.32. Use \initOnce()` instead for safe idempotent initialization.–core-libs/core/src/site-context/services/site-context-routes-handler.ts:L51-L52`.
- JSDoc su `isNavigating`: serve a evitare il loop infinito fra `subscribeChanges` e `subscribeRouting` — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L41-L45. Curiosamente il commento dice che `subscribeChanges` "triggers a navigation", mentre il codice fa `location.replaceState` (nessuna navigazione Angular) — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L122-L124.
- `It maps the string "storefront" (used in OCC) to the "baseSite" (used in Spartacus)` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L111-L111.
- `There is no OCC API to load one site based on Uid.` — core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L86-L87: `loadBaseSite` carica tutti i siti e filtra.
- `// We take the first store as the base store.` — core-libs/core/src/occ/adapters/site-context/converters/base-site-normalizer.ts:L23-L23.
- `// avoid dispatching \change` action when we're just setting the initial value:–core-libs/core/src/site-context/store/effects/languages.effect.ts:L63-L63, core-libs/core/src/site-context/store/effects/currencies.effect.ts:L58-L58`.
- `// if we want to reuse SITE_CONTEXT_FEATURE const in config, we have to use factory instead of plain object` — core-libs/core/src/site-context/store/site-context-store.module.ts:L21-L21.
- `CAUTION: This property should be accessed only when those configs are stable` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L84-L84.
- `// We could return a simple value, but we return Observable only to conform the class interface \SiteContext<T>`–core-libs/core/src/site-theme/facade/site-theme.service.ts:L92-L92`.
- `// Empty string is valid – it means "default / no theme".` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L132-L133.
- Precedenza del tema (statico > opzionale utente > CMS) documentata nel JSDoc — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L97-L100, e ripetuta nel JSDoc del toggle — core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L655-L670.
- Il JSDoc di `CurrencyService.setActive` dice "Sets the active language." (copia-incolla) — core-libs/core/src/site-context/facade/currency.service.ts:L64-L66.
- `// TODO(#11133): Consider throwing in SSR` su `WindowRef.nativeWindow` — core-libs/core/src/window/window-ref.ts:L42-L42.

- `// Do not change order of subscription! Read should happen before write on context change.` — core-libs/core/src/state/services/state-persistence.service.ts:L63-L63.

8. Extension point

API pubbliche (riesportate da core-libs/core/public_api.ts:L31-L32 tramite core-libs/core/src/site-context/index.ts:L7-L16 e core-libs/core/src/site-theme/index.ts:L7-L13):

EXTENSION POINT	DOVE	COME SI USA
Config `context` (`SiteContextConfig`)	core-libs/core/src/site-context/config/site-context-config.ts:L10-L19	`provideConfig({ context: { baseSite, language, currency, urlParameters, theme } })`, come la demo app — projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L11-L31
Config `siteTheme` (`SiteThemeConfig`, `defaultSiteThemeConfig`)	core-libs/core/src/site-theme/config/site-theme-config.ts:L11-L23, core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22	sovrascrivere `optionalThemes` (array: sostituisce, non fonde)
`SiteContextConfigInitializer`	core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L46-L98	estendere i metodi `protected resolveConfig`/`getConfig` e ri-fornire la classe
`SiteAdapter` / `SiteConnector`	core-libs/core/src/site-context/connectors/site.adapter.ts:L17-L57, core-libs/core/src/site-context/connectors/site.connector.ts:L19-L31	fornire un adapter alternativo a `OccSiteAdapter`
Token normalizer (`BASE_SITE_NORMALIZER`, `LANGUAGE_NORMALIZER`, …)	core-libs/core/src/site-context/connectors/converters.ts:L12-L38	aggiungere converter `multi`
Facade `BaseSiteService`, `LanguageService`, `CurrencyService`, `SiteThemeService`	core-libs/core/src/site-context/facade/base-site.service.ts:L21-L22, core-libs/core/src/site-theme/facade/site-theme.service.ts:L27-L32	sostituzione via DI (non sono `providedIn: 'root'`)
`ContextServiceMap`	core-libs/core/src/site-context/providers/context-service-map.ts:L18-L33	mappare un nome di contesto su una classe `SiteContext<T>`
Initializer e persistence service (`setFallbackValue`, `setDefaultFromConfig`, `onRead` protected)	core-libs/core/src/site-context/services/language-initializer.ts:L51-L79, core-libs/core/src/site-context/services/language-state-persistence.service.ts:L38-L47	personalizzare la precedenza
`SiteContextUrlSerializer`, `SiteContextParamsService`, tipi `UrlTreeWithSiteContext`	core-libs/core/src/site-context/services/site-context-url-serializer.ts:L14-L160, core-libs/core/src/site-context/services/site-context-params.service.ts:L19-L108	estendere i metodi `protected` del serializer

EXTENSION POINT	DOVE	COME SI USA
Eventi `LanguageSetEvent`, `CurrencySetEvent`	core-libs/core/src/site-context/events/site-context.events.ts:L12-L29	`EventService.get(LanguageSetEvent)`
Action/selector `SiteContextActions`, `SiteContextSelectors`, `SiteThemeActions`, `SiteThemeSelectors`	core-libs/core/src/site-context/store/actions/index.ts:L7-L8, core-libs/core/src/site-context/store/selectors/index.ts:L7-L8, core-libs/core/src/site-theme/store/actions/index.ts:L7-L8, core-libs/core/src/site-theme/store/selectors/index.ts:L7-L8	`ofType(SiteContextActions.LANGUAGE_CHANGE)` per reagire ai cambi reali
`OccSiteAdapter`, `SiteContextInterceptor`, `BaseSiteNormalizer`, `SiteContextOccModule`	core-libs/core/src/occ/adapters/site-context/index.ts:L7-L10	estendere/sostituire via DI
Feature toggle	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L306-L311, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L655-L670, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L700-L710	`provideFeatureToggles`/config `features` (vedi capitolo 42)
`ThemeService` (storefront)	core-libs/storefront/layout/theme/theme.service.ts:L52-L61, export core-libs/storefront/layout/theme/index.ts:L7-L8	estendere `setTheme` per applicare il tema altrove

Non pubblici (non contarci in un upgrade): `SiteContextRoutesHandler`, `contextInitializerProviders`, `contextServiceProviders`, `siteContextParamsProviders`, `defaultSiteContextConfigFactory`, `baseSiteConfigValidator`, `SiteThemeInitializer`, `SiteThemePersistenceService`, `SITE_THEME_ID`, reducer ed effects — core-libs/core/src/site-context/providers/index.ts:L7-L8, core-libs/core/src/site-context/config/index.ts:L7-L9, core-libs/core/src/site-theme/index.ts:L7-L13, core-libs/core/src/site-context/services/index.ts:L7-L13.

> DA VERIFICARE — Estendere `ContextServiceMap` con un contesto aggiuntivo (per farlo comparire nell'URL o nel selettore) è possibile sul piano dei tipi (`[context: string]: Type<SiteContext<any>>`), ma nel repository non c'è alcun esempio che lo faccia: il comportamento end-to-end (validità dei valori, ordine degli initializer) non è verificato.

Ricette pertinenti in Parte III:

- configurare base site, lingue, valute e prefissi URL; sostituire una facade o un initializer via DI: 50 — Ricette: configurazione, DI, interceptor, eventi;
- reagire al cambio lingua/valuta con eventi o action: 50 — Ricette: configurazione, DI, interceptor, eventi;
- un adapter o un normalizer custom per i base site: 52 — Ricette: OCC, adapter, converter, endpoint, meta resolver;
- aggiungere un tema opzionale e le sue classi CSS, traduzioni del selettore: 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni;
- selettori di contesto e switcher del tema nel layout: 51 — Ricette: componenti CMS, outlet, route e layout;

- cosa è API pubblica e cosa no (`SiteContextRoutesHandler`, provider interni): 54 — API pubblica vs interna, upgrade, anti-pattern, checklist.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `context.baseSite` **statico spegne tutto il caricamento dinamico.** Non solo il base site: anche `urlParameters`, lingue, valute e tema dal backend non vengono letti; restano i default (`en`, `USD`, nessun `urlParameters`) se l'app non li fornisce — `core-libs/core/src/site-context/site-context.module.ts:L37-L40`, `core-libs/core/src/site-context/config/default-site-context-config.ts:L13-L58`. Il JSDoc del toggle `applyBaseSiteThemeFromCms` lo conferma per il tema — `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L655-L670`.
2. **Il messaggio del validator cita una chiave sbagliata**: `context.parameters.baseSite`, mentre la chiave reale è `context.baseSite` — `core-libs/core/src/site-context/config/base-site-config-validator.ts:L13-L13`.
3. **Gli array non si fondono.** Una lista di lingue fornita dall'app sostituisce interamente quella di default; idem quelle del backend — `core-libs/core/src/config/utils/deep-merge.ts:L7-L9`, `core-libs/core/src/config/utils/deep-merge.ts:L36-L45`.
4. `undefined` **sovrascrive.** In modalità dinamica `getIsoCodes` restituisce `undefined` se manca la lista o il default; `deepMerge` assegna anche `undefined`, quindi il default di lingua/valuta verrebbe cancellato — `core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L124-L135`, `core-libs/core/src/config/utils/deep-merge.ts:L36-L45`. `DA VERIFICARE` a runtime: dipende dai dati restituiti da OCC.
5. **Il bootstrap non aspetta i valori attivi.** I `APP_INITIALIZER` di language/currency/baseSite e del tema non restituiscono la Promise: si attende solo `getStable('context')` e il location initializer — `core-libs/core/src/site-context/providers/context-initializer-providers.ts:L12-L34`, `core-libs/core/src/site-theme/providers/site-theme-initializer-providers.ts:L10-L23`, `angular:packages/core/src/application/application_init.ts:L244-L255`. Codice che legge sincronicamente il valore attivo molto presto può trovare il default.
6. `BaseSiteService.setActive` **non valida.** Lingua e valuta scartano valori non ammessi, e così il tema finché `applyBaseSiteThemeFromCms` è spento (con il toggle attivo accetta qualunque stringa non null/undefined — `core-libs/core/src/site-theme/facade/site-theme.service.ts:L130-L137`); il base site non valida mai (`isValid` esiste ma non è chiamato) — `core-libs/core/src/site-context/facade/base-site.service.ts:L72-L82`, `core-libs/core/src/site-context/facade/base-site.service.ts:L101-L108`.
7. **Lingue/valute fuori lista vengono ignorate in silenzio**, anche se il backend le espone — `core-libs/core/src/site-context/facade/language.service.ts:L59-L69`, `core-libs/core/src/site-context/facade/currency.service.ts:L67-L77`.
8. **L'ordine di** `urlParameters` **conta.** Il parse avanza sui nomi anche quando il segmento non matcha; valori omonimi fra parametri diversi possono finire sul parametro sbagliato — `core-libs/core/src/site-context/services/site-context-url-serializer.ts:L91-L105`.
9. `serialize` **aggiunge sempre tutti i prefissi**, usando il valore attivo (o il default) quando il tree non li ha: un URL senza prefisso diventa un URL con prefisso completo — `core-libs/core/src/site-context/services/site-context-url-serializer.ts:L146-L159`, `core-libs/core/src/site-context/services/site-context-params.service.ts:L65-L77`.
10. **Cambio di contesto =** `replaceState`**, non navigazione**: nessuna nuova voce di history, nessun guard/resolver eseguito — `core-libs/core/src/site-context/services/site-context-routes-handler.`

ts:L117-L125. E il primo valore non provoca riscrittura (serve un `contextValues[param]` precedente) — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L117-L126.

11. **L'URL vince sul federated login** per la lingua: entrambi chiamano `setActive` senza guardare `isInitialized()`, e l'URL viene dopo — core-libs/core/src/site-context/services/language-initializer.ts:L35-L44, core-libs/core/src/site-context/services/site-context-routes-handler.ts:L163-L168.
12. `theme` **non è nel** `ContextServiceMap`: `setValue('theme', …)` è no-op e il tema non può stare nell'URL; inoltre `SiteContextParamsService.getValues([... 'theme'])` non emette mai, perché quel ramo produce `''` e il filtro richiede tutti valori truthy — core-libs/core/src/site-context/providers/context-service-map.ts:L26-L33, core-libs/core/src/site-context/services/site-context-params.service.ts:L93-L107.
13. **Eventi vs action.** `LanguageSetEvent`/`CurrencySetEvent` scattano su ogni `SET_ACTIVE_*`, anche l'iniziale; `LANGUAGE_CHANGE`/`CURRENCY_CHANGE` saltano il primo valore. Per reagire solo ai cambi reali usare le action `*Change` — core-libs/core/src/site-context/events/site-context-event.builder.ts:L41-L72, core-libs/core/src/site-context/store/effects/languages.effect.ts:L58-L70.
14. **Nessuna action di cambio per il base site.** `BASE_SITE_CHANGE`/`BaseSiteChange` esiste ma nessun file non-spec delle librerie la dispatcha, e non c'è un effect "activate" — core-libs/core/src/site-context/store/actions/base-site.action.ts:L11-L20, core-libs/core/src/site-context/store/effects/base-site.effect.ts:L16-L74. Anche `LoadBaseSite` (singolo) non è dispatchata dalle librerie.
15. `loadBaseSite` **non normalizza**: a differenza di `loadBaseSites`, non applica `BASE_SITE_NORMALIZER`, quindi `baseStore` non viene valorizzato per quella via — core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L86-L107, core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L109-L120.
16. `details` **può essere** `undefined`: `SET_ACTIVE_BASE_SITE` con entità caricate ma uid assente produce `details = entities[uid]` — core-libs/core/src/site-context/store/reducers/base-site.reducer.ts:L29-L41, mentre il tipo dichiara `BaseSite` — core-libs/core/src/site-context/store/state.ts:L37-L41.
17. `lang`**/**`curr` **e l'URL base.** L'interceptor confronta con `getBaseUrl()` comprensivo del base site: le chiamate costruite con `baseSite: false` (es. `basesites`) non ricevono i parametri quando un base site è attivo; se invece non c'è ancora alcun base site (né statico), `urlPathJoin` scarta il segmento vuoto e `getBaseUrl()` diventa prefisso anche di `basesites` — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L58-L58, core-libs/core/src/occ/services/occ-endpoints.service.ts:L37-L42, core-libs/core/src/occ/utils/occ-url-util.ts:L12-L26, core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L114-L114.
18. **Prima dello store l'interceptor usa i default di config** (`context.language[0]`, `context.currency[0]`), cioè `en`/`USD` con base site statico e config di default — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L36-L43.
19. **SSR.** `WindowRef.location` lancia se manca `SERVER_REQUEST_URL` (rilevante per il caricamento dinamico); lo storage non esiste; gli slice `siteContext` e `siteTheme` passano al browser via TransferState, ma il browser li applica all'`INIT` solo se l'utente non è loggato — core-libs/core/src/window/window-ref.ts:L70-L89, core-libs/core/src/state/reducers/transfer-state.reducer.ts:L74-L92. Vedi 07 — SSR, hydration e TransferState.
20. `reloadPage()` **usa il global** `location` e non `WindowRef`; è protetto da `isPlatformBrowser` — core-libs/core/src/site-context/store/effects/languages.effect.ts:L72-L88.
21. **Il tema "Default" può avere** `className ''`. Con `applyBaseSiteThemeFromCms ''` è valido ma non viene ripristinato dallo storage (`onRead` richiede un valore truthy); `ThemeService.setTheme('')` rimuove la classe precedente senza aggiungerne — core-libs/core/src/site-theme/facade/site-theme.service.ts:L130-L137, core-libs/core/src/site-theme/services/site-theme-persistence.service.ts:L26-L43, core-libs/storefront/layout/theme/theme.service.ts:L52-L61.

22. **Default tema letto in modo sincrono**: con il toggle CMS `getDefault()` usa `getLastValueSync` sul base site; se non è ancora risolto il default è `''` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L51-L71. La seconda subscription di `SiteThemeInitializer` corregge poi il tema quando il base site arriva — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L102-L127.
23. **Collisione di nomi** `siteTheme`: chiave di config, nome della feature NgRx e chiave di persistenza — core-libs/core/src/site-theme/config/site-theme-config.ts:L16-L16, core-libs/core/src/site-theme/store/state.ts:L9-L9, core-libs/core/src/site-theme/providers/site-theme-id.ts:L7-L7.
24. **Ordine degli import**: se `SiteContextModule.forRoot()` finisse prima di `RouterModule.forRoot`, il `DefaultUrlSerializer` del router vincerebbe e i prefissi non verrebbero gestiti — core-libs/core/src/base-core.module.ts:L38-L38, angular:packages/router/src/router_module.ts:L69-L72.
25. **Subscription mai chiuse** nei costruttori di `SiteContextInterceptor` e `OccEndpointsService` (servizi root con durata dell'app) — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L45-L52, core-libs/core/src/occ/services/occ-endpoints.service.ts:L50-L54.

> DA VERIFICARE — Ordine effettivo fra l'`APP_INITIALIZER` di `ConfigInitializerModule` e quelli di `SiteContextModule` quando la config è statica: se `initialize()` degli initializer di contesto gira prima che `ongoingScopes$` sia valorizzato, `getStable('context')` semplicemente attende (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70); l'ordine reale non è stato osservato a runtime.

---

10. Come fare debug

- **Quale modalità è attiva?** Se `context.baseSite` è presente nella config statica, `SiteContextConfigInitializer` non gira: un breakpoint in `initSiteContextConfig` (core-libs/core/src/site-context/site-context.module.ts:L30-L41) lo conferma subito. In modalità dinamica, un breakpoint in `getConfig()` (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L77-L98) mostra la config derivata dal base site.
- **Errore "Cannot get base site config!"**: confrontare l'URL codificato nel messaggio con gli `urlPatterns` della risposta di `basesites?fields=FULL` (tab Network) — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L61-L74. In SSR controllare che `SERVER_REQUEST_URL` sia fornito — core-libs/core/src/window/window-ref.ts:L70-L89.
- **Config finale**: iniettare `ConfigInitializerService` e loggare `getStable('context')`, oppure ispezionare `SiteContextConfig` (che è `useExisting: Config`) — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70.
- **Store**: con Redux DevTools cercare le action `[Site-context] Set Active Language`, `[Site-context] Language Change`, `[Site-context] Load BaseSites Success` e lo slice `siteContext` (`activeLanguage`, `activeCurrency`, `baseSite.activeSite`) — core-libs/core/src/site-context/store/actions/base-site.action.ts:L11-L20, core-libs/core/src/site-context/store/state.ts:L8-L41. Per il tema: `[Site-theme] Set Active Site Theme` e `siteTheme.themes.activeSiteTheme` — core-libs/core/src/site-theme/store/actions/site-themes.action.ts:L9-L25.
- **Un valore "non cambia"**: verificare che sia nella lista `context.language`/`context.currency` (altrimenti `setActive` lo scarta, core-libs/core/src/site-context/facade/language.service.ts:L88-L95) e che non sia già l'attivo.
- **Precedenza inattesa**: controllare il `localStorage` alle chiavi `spartacus⚿⚿language`, `spartacus⚿⚿currency`, `spartacus⚿⚿siteTheme` — core-libs/core/src/state/services/state-persistence.service.ts:L124-L131. Cancellarle riporta al default (salvo prefisso URL).

- **URL**: in console `router.parseUrl('/electronics-spa/de/EUR/cart')` restituisce un `UrlTree` con proprietà `siteContext`; `router.serializeUrl(tree)` mostra i prefissi riaggiunti — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L62-L71, core-libs/core/src/site-context/services/site-context-url-serializer.ts:L126-L131. Se i prefissi non compaiono, verificare che `UrlSerializer` sia risolto in `SiteContextUrlSerializer` (ordine import, core-libs/core/src/base-core.module.ts:L38-L38).
- **Chiamate OCC**: nel tab Network verificare il segmento base site nel path e i query param `lang`/`curr`; se mancano, l'URL della richiesta non contiene `getBaseUrl()` — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68.
- **Warning** `Couldn't find site context service for '...'` in dev mode: il nome in `urlParameters` è mappato in `ContextServiceMap` ma `injector.get` della classe fallisce (es. una mappa personalizzata che punta a una classe non fornita). Senza `SiteContextModule.forRoot()` il warning non compare affatto: è quel modulo a fornire `ContextServiceMap` e `SiteContextParamsService` — core-libs/core/src/site-context/services/site-context-params.service.ts:L50-L63, core-libs/core/src/site-context/site-context.module.ts:L47-L65, core-libs/core/src/site-context/providers/site-context-params-providers.ts:L14-L18.
- **Tema**: ispezionare le classi CSS dell'elemento host del root component — nell'app demo è `app-root`, perché `main.ts` fa `bootstrapApplication(AppComponent, appConfig)` (projects/storefrontapp/src/main.ts:L17-L17, projects/storefrontapp/src/app/app.component.ts:L15-L15) — e la catena `SiteThemeService.getActive()` → `ThemeService.setTheme` — core-libs/storefront/layout/theme/theme.service.ts:L35-L61.

---

11. Domande di autoverifica

1. **Che cosa succede se configuro staticamente** `context.baseSite` **ma non** `context.language`**?** Il `CONFIG_INITIALIZER` restituisce `null` e il backend non viene interrogato per la config; la lingua di default resta il primo elemento del default di Spartacus, `en`. — core-libs/core/src/site-context/site-context.module.ts:L30-L41, core-libs/core/src/site-context/config/default-site-context-config.ts:L13-L58.
2. **Perché** `SiteContextModule.forRoot()` **deve essere importato dopo** `RouterModule.forRoot`**?** Perché registra `{ provide: UrlSerializer, useExisting: SiteContextUrlSerializer }`, che deve prevalere sul `DefaultUrlSerializer` registrato dal router in `ROUTER_PROVIDERS`. — core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L18, angular:packages/router/src/router_module.ts:L69-L72, core-libs/core/src/base-core.module.ts:L38-L38.
3. **Qual è l'ordine di precedenza per la valuta attiva all'avvio?** Config stabile → valore dall'URL (`initOnce`) → `localStorage` (solo se non già inizializzata) → primo elemento di `context.currency`. — core-libs/core/src/site-context/services/currency-initializer.ts:L33-L41, core-libs/core/src/site-context/services/currency-state-persistence.service.ts:L24-L42.
4. **Il base site attivo viene salvato nel** `localStorage`**?** No: `BaseSiteInitializer` passa solo da URL e default di config, senza persistence service. — core-libs/core/src/site-context/services/base-site-initializer.ts:L31-L38.
5. **Come fa Spartacus ad aggiungere** `/electronics-spa/en/USD` **a tutti i link senza che i componenti lo sappiano?** `SiteContextUrlSerializer.serialize` antepone a ogni URL serializzato dal router i valori dei parametri in `urlParameters` (dal tree o attivi). — core-libs/core/src/site-context/services/site-context-url-serializer.ts:L126-L159, angular:packages/router/src/router.ts:L588-L595.
6. **Se l'utente cambia lingua dal selettore, viene eseguita una navigazione Angular?** No: `SiteContextRoutesHandler.subscribeChanges` riscrive l'URL con `location.replaceState`; il ricaricamento dei contenuti avviene tramite `LANGUAGE_CHANGE` (reset CMS + `PageEffects`) ed eventualmente `location.`

`reload()` se `reloadOnLanguageChange` è attivo. — core-libs/core/src/site-context/services/site-context-routes-handler.ts:L111-L131, core-libs/core/src/cms/store/reducers/index.ts:L78-L87, core-libs/core/src/site-context/store/effects/languages.effect.ts:L72-L88.

7. **Voglio eseguire codice solo quando la lingua cambia davvero, non all'impostazione iniziale. Uso** `LanguageSetEvent` **o** `LanguageChange`**?** `LanguageChange`: è emessa da `activateLanguage$` solo se esiste un valore precedente, mentre `LanguageSetEvent` scatta su ogni `SET_ACTIVE_LANGUAGE`. — core-libs/core/src/site-context/store/effects/languages.effect.ts:L58-L70, core-libs/core/src/site-context/events/site-context-event.builder.ts:L41-L72.
8. **Da dove arrivano i query param** `lang` **e** `curr` **delle chiamate OCC?** Da `SiteContextInterceptor`, che parte dai default di config, si aggiorna su `getActive()` di lingua e valuta e li aggiunge solo agli URL che contengono `getBaseUrl()`. — core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L25-L68.
9. **Con** `applyBaseSiteThemeFromCms` **attivo, un tema scelto dall'utente tra quelli ad alto contrasto viene sovrascritto dal tema CMS?** No: la precedenza è tema statico (`context.theme`) > tema opzionale scelto dall'utente > tema CMS. — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L97-L127, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L141-L147.
10. **Perché** `theme` **non può comparire tra gli** `urlParameters` **con effetto?** Perché `ContextServiceMap` mappa solo `language`, `currency` e `baseSite`: per `theme` `getSiteContextService` non trova un servizio e `setValue` non fa nulla; inoltre `SiteThemeInitializer` non passa da `SiteContextRoutesHandler`. — core-libs/core/src/site-context/providers/context-service-map.ts:L26-L33, core-libs/core/src/site-context/services/site-context-params.service.ts:L79-L84, core-libs/core/src/site-theme/services/site-theme-initializer.ts:L36-L50.

TAVOLA 13.1 **4.1 Risoluzione della config context al bootstrap**

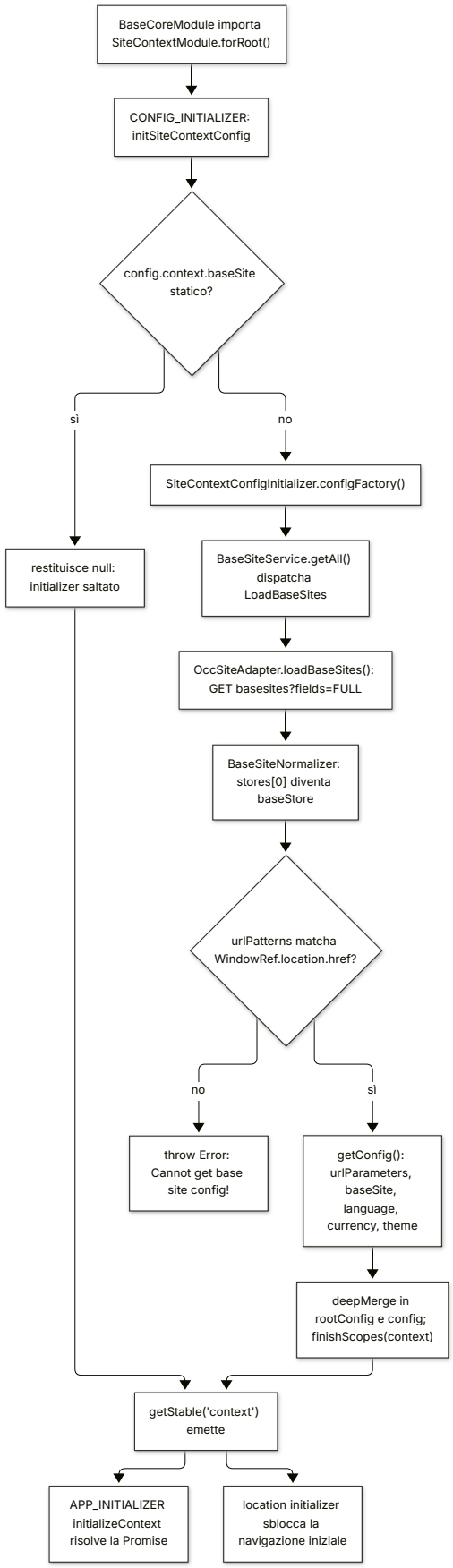


TAVOLA 13.2 **4.2 Precedenza nell'inizializzazione della lingua**

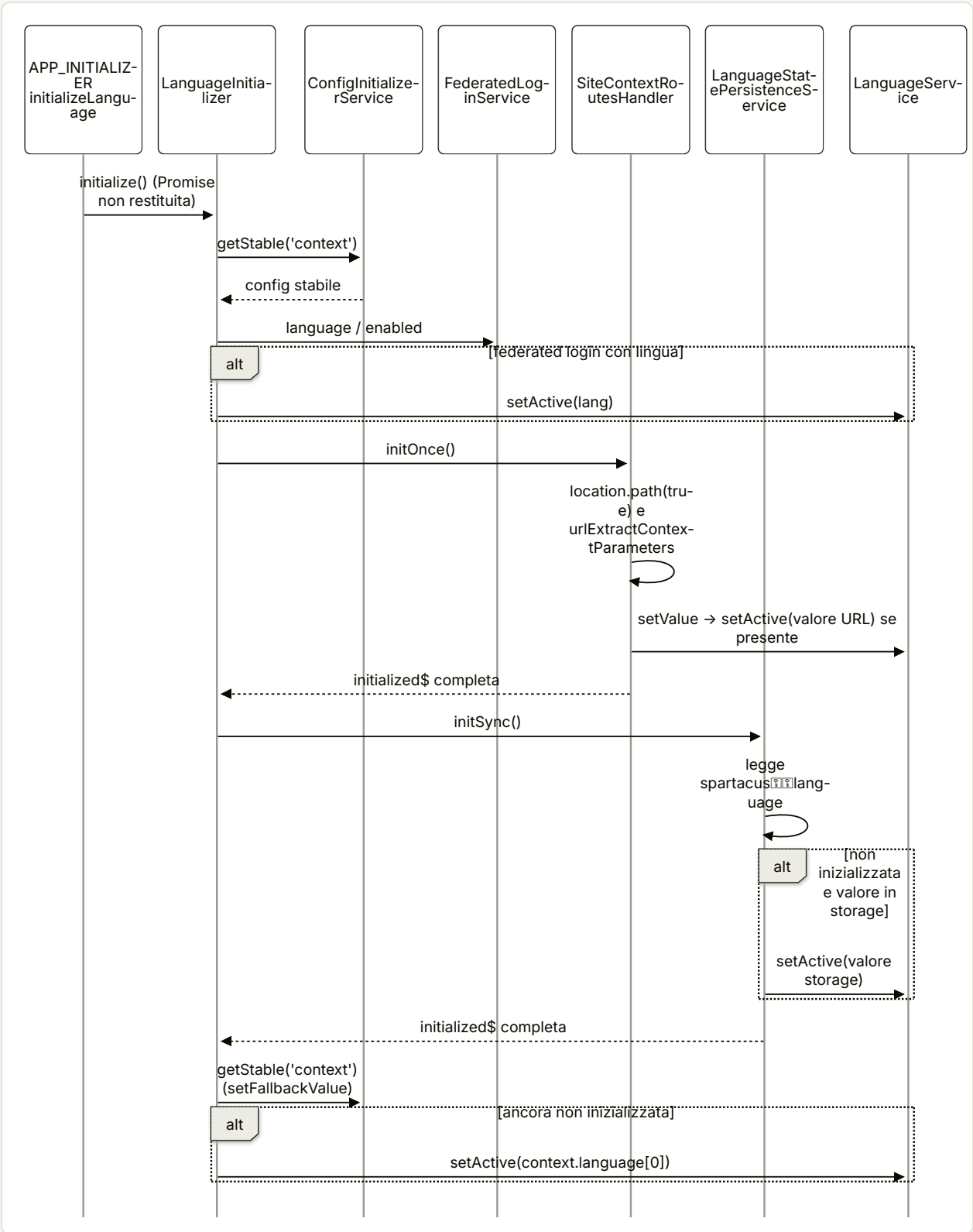


TAVOLA 13.3 **4.3 Cambio lingua dalla UI**

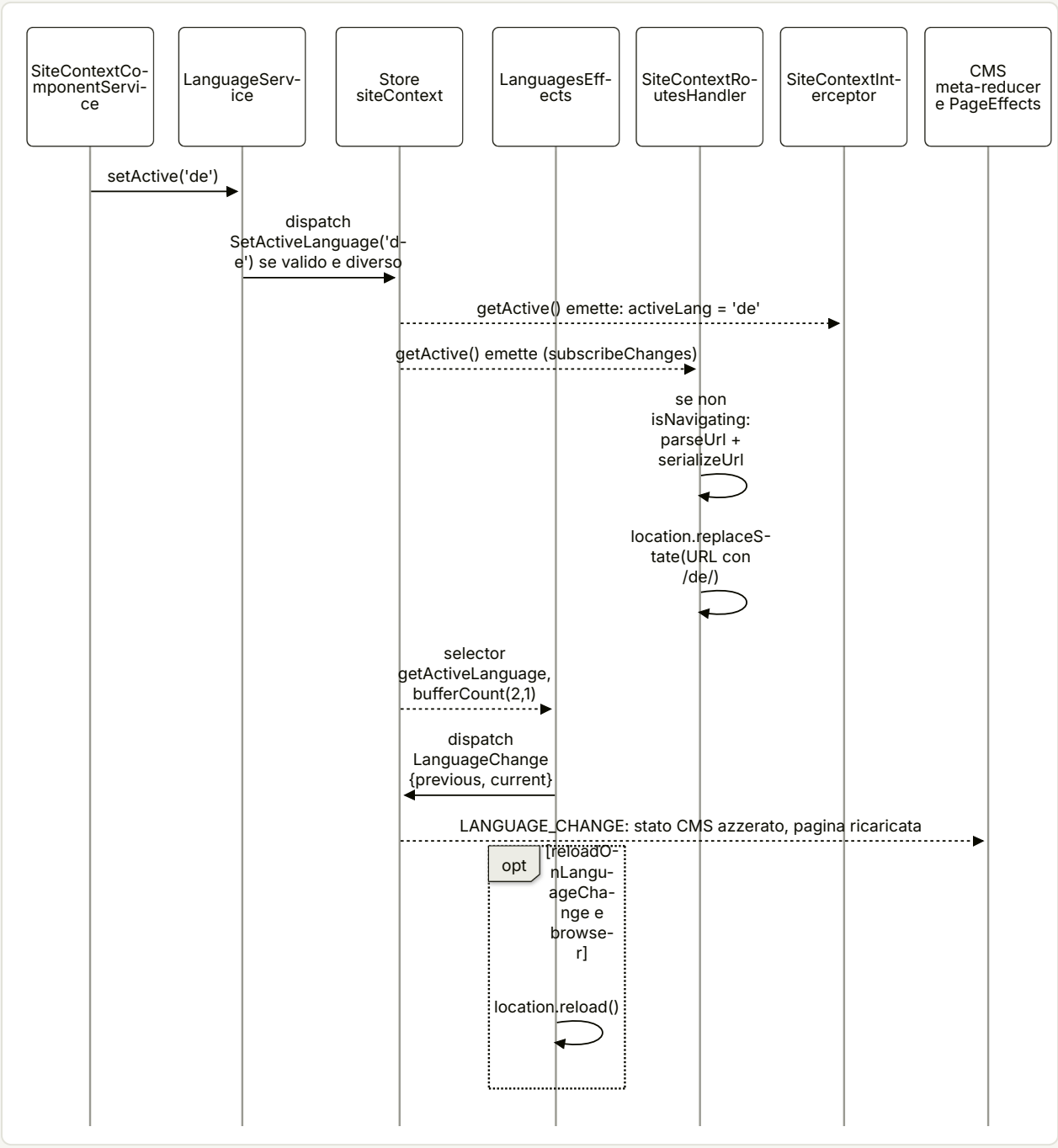
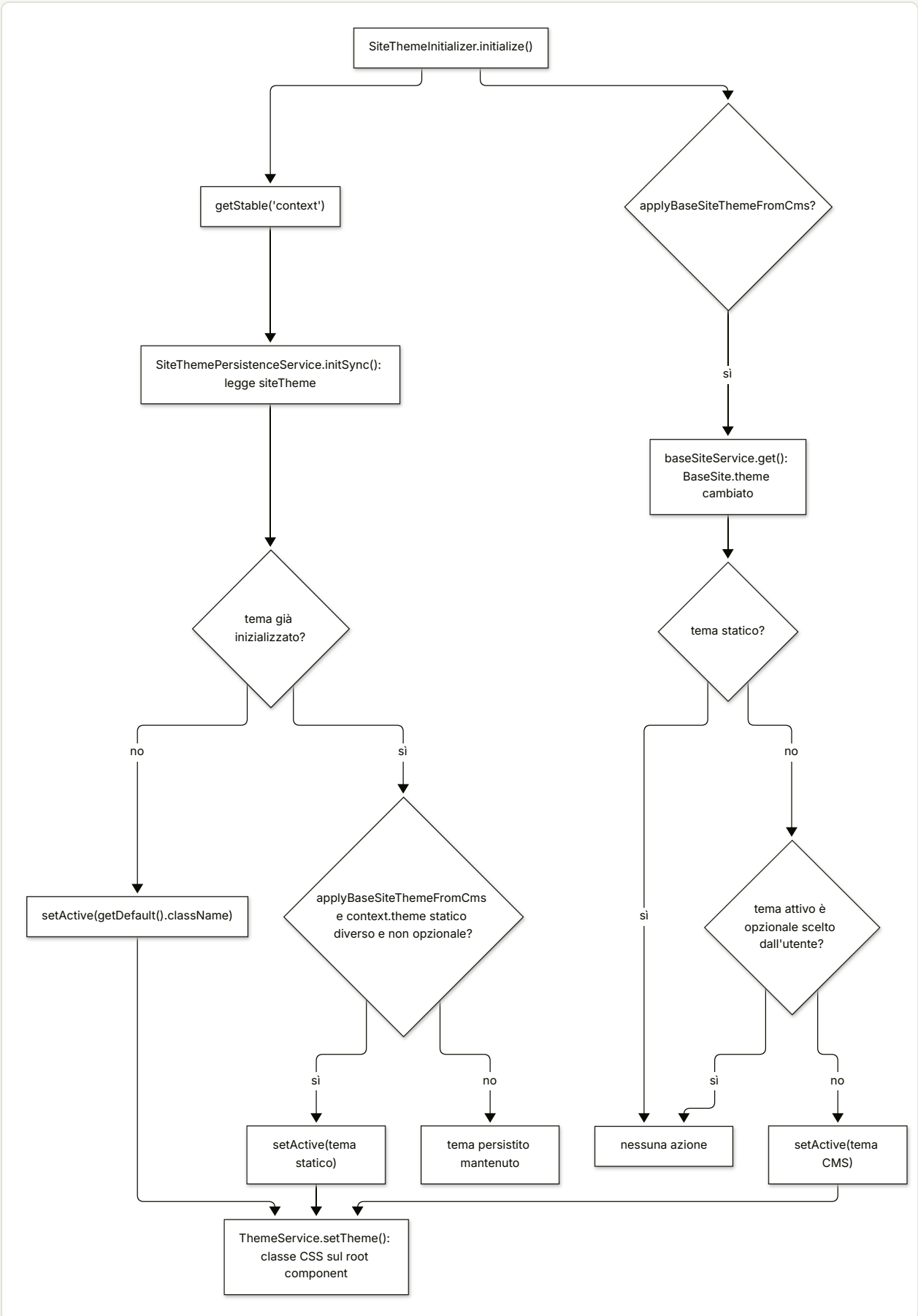


TAVOLA 13.4 **4.5 Precedenza del tema**



14

Routing: route semantiche, URL, page guard, contesto

Route semantiche configurabili e route wildcard CMS protetta da `CmsPageGuard`. Protected routes.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Questo capitolo spiega come Spartacus si appoggia al Router di Angular senza scrivere quasi nessuna route "vera". Le route sono **semantiche**: si chiamano `product`, `cart`, `login` e i loro path reali vengono dalla configurazione `routing.routes`. All'avvio un servizio riscrive la configurazione del Router. Durante la navigazione uno store NgRx tiene lo stato "corrente" e quello "in arrivo". Un unico guard, `CmsPageGuard`, decide se e quale pagina CMS mostrare. Il router di Angular in sé (matching, guard, eventi) è descritto nel cap. 05. L'ordine di bootstrap è nel cap. 10, il caricamento delle pagine CMS nel cap. 15, il loro rendering nel cap. 16. `AuthGuard` e il flusso OAuth sono nel cap. 22.

1. A cosa serve e perché esiste

Uno storefront e-commerce ha due esigenze in conflitto:

1. **Gli URL appartengono al business.** `/product/123/nome`, `/p/123`, `/Brands/Canon/c/brand_10` cambiano da progetto a progetto, e spesso da lingua a lingua. Il codice dei componenti non deve contenerli.
2. **Le pagine appartengono al CMS.** Quasi tutto il contenuto (layout, slot, componenti) arriva dal backend SAP Commerce. Una route Angular statica per ogni pagina non è possibile.

Spartacus risolve la prima esigenza con le **route semantiche**. Il modello è `RoutingConfig.routing.routes`: un dizionario `nomeRoute → { paths, matchers, paramsMapping, ... }` aperto anche a pagine custom (core-libs/core/src/routing/configurable-routes/routes-config.ts:L11-L13, core-libs/core/src/routing/configurable-routes/routes-config.ts:L16-L57). Le route Angular dichiarano solo il proprio nome in `data.cxRoute`. All'avvio `ConfigurableRoutesService` ne riscrive `path`/`matcher` a partire dalla config (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L48-L52, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L109). Per generare link si usano i comandi `{ cxRoute, params }`, tradotti da `SemanticPathService` e dalla pipe `cxUrl` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L41-L68, core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts:L11-L18).

La seconda esigenza è risolta da una route wildcard `**` aggiunta in coda alla config del Router e protetta da `CmsPageGuard` (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25). Il guard carica la pagina CMS che corrisponde al **PageContext** della navigazione in corso. Se la pagina non esiste mostra quella `notFound` (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84). Il `PageContext` (`id` + `PageType`) è calcolato da un serializer custom del router-state NgRx (core-libs/core/src/routing/store/reducers/router.reducer.ts:L112-L217).

Intorno a questo nucleo ci sono:

- **Protected routes e secure portal.** Con `routing.protected = true` lo storefront è chiuso agli anonimi, tranne le route con `protected: false` (core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L30-L45, core-libs/core/src/routing/protected-routes/protected-routes.guard.ts:L23-L33).
- **External routes.** Gli URL che non rientrano nei glob `routing.internal` causano un full page reload verso un altro sistema (core-libs/core/src/routing/external-routes/external-routes.service.ts:L46-L70, core-libs/core/src/routing/external-routes/external-routes.guard.ts:L22-L41).
- **Context.** I componenti possono chiedere "il contesto" (per esempio le order entries del carrello attivo) tramite `ContextService`, che lo ricava dal `data.cxContext` delle route attive (core-libs/storefront/context/context.service.ts:L16-L37, core-libs/storefront/context/routing-context.service.ts:L31-L63).

- **OnNavigateService.** Riporta scroll e focus in cima alla pagina dopo ogni navigazione (core-libs/storefront/router/on-navigate.service.ts:L80-L119).

Dove viene attivato

- AppRoutingModule registra il Router con RouterModule.forRoot([], { anchorScrolling: 'enabled', initialNavigation: 'enabledBlocking' }) (core-libs/storefront/router/app-routing.module.ts:L13-L37).
- BaseStorefrontModule importa RoutingModule.forRoot() di storefront (core-libs/storefront/base-storefront.module.ts:L41-L41). Questo importa il RoutingModule.forRoot() di core e CmsRouteModule, e registra la config di default e i guard globali (core-libs/storefront/cms-structure/routing/routing.module.ts:L18-L40).
- Il RoutingModule di core collega lo store: feature 'router', effects, StoreRouterConnectingModule (core-libs/core/src/routing/routing.module.ts:L73-L82). Fornisce anche serializer, initializer delle route e LOCATION_INITIALIZED (core-libs/core/src/routing/routing.module.ts:L84-L108).
- ExternalRoutesModule.forRoot() è opt-in. L'app demo lo importa esplicitamente (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L246-L246).

---

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/routing/configurable-routes/config/routing-config.ts	RouteLoadStrategy, RoutingConfig, RoutingConfigDefinition (routes, protected, loadStrategy) e augmentation di Config
core-libs/core/src/routing/configurable-routes/routes-config.ts	RoutesConfig, RouteConfig (paths, matchers, paramsMapping, disabled, protected, authFlow), ParamsMapping
core-libs/core/src/routing/configurable-routes/routing-config.service.ts	RoutingConfigService: lookup per nome, load strategy, mappa inversa path → nome
core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts	ConfigurableRoutesService: all'avvio riscrive path/matcher delle route con data.cxRoute
core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts	SecurePortalConfigInitializer: ricava routing.protected da BaseSite.requiresAuthentication
core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts	SemanticPathService: {cxRoute, params} → segmenti URL
core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts	Pipe cxUrl (UrlPipe)
core-libs/core/src/routing/configurable-routes/url-translation/product-url.pipe.ts	Pipe cxProductUrl (ProductURLPipe)
core-libs/core/src/routing/configurable-routes/url-translation/url.module.ts	UrlModule: importa e riesporta le due pipe standalone
core-libs/core/src/routing/configurable-routes/url-translation/url-command.ts	Tipi UrlCommandRoute, UrlCommand, UrlCommands
core-libs/core/src/routing/configurable-routes/url-translation/url-parsing.service.ts	UrlParsingService: segmenti primari, matchPath

PATH	RUOLO
core-libs/core/src/routing/configurable-routes/url-translation/path-utils.ts	isParam, getParamName, ensureLeadingSlash, removeLeadingSlash (non esportati)
core-libs/core/src/routing/configurable-routes/url-translation/testing/mock-url.pipe.ts	MockUrlPipe e URL_TESTING_ALLOWLISTED_PARAMS per i test
core-libs/core/src/routing/configurable-routes/url-translation/testing/url-testing.module.ts	UrlTestingModule.allowlistParams()
core-libs/core/src/routing/services/url-matcher.service.ts	UrlMatcherService: matcher falsy / da path / combinato / opposto / glob
core-libs/core/src/routing/services/activated-routes.service.ts	ActivatedRoutesService.routes$: catena root → foglia a ogni NavigationEnd
core-libs/core/src/routing/url-matcher/default-url-matcher.ts	DEFAULT_URL_MATCHER (factory di matcher dai paths)
core-libs/core/src/routing/url-matcher/url-matcher-factory.ts	Tipo UrlMatcherFactory
core-libs/core/src/routing/facade/routing.service.ts	RoutingService: go, goByUrl, getUrl, back, selettori del router-state
core-libs/core/src/routing/facade/routing-params.service.ts	RoutingParamsService: params fusi di tutte le route attive
core-libs/core/src/routing/store/routing-state.ts	ROUTING_FEATURE, RouterState (con nextState), ActivatedRouterStateSnapshot
core-libs/core/src/routing/store/reducers/router.reducer.ts	initialState, reducer, reducerToken, CustomSerializer (calcolo di PageContext)
core-libs/core/src/routing/store/actions/router.action.ts	ChangeNextPageContext
core-libs/core/src/routing/store/selectors/routing.selector.ts	getRouterState, getPageContext, getNextPageContext, isNavigating, getSemanticRoute
core-libs/core/src/routing/store/effects/router.effect.ts	RouterEffects.clearCmsRoutes$
core-libs/core/src/routing/models/page-context.model.ts	PageContext, HOME_PAGE_CONTEXT, SMART_EDIT_CONTEXT
core-libs/core/src/routing/models/cms-route.ts	CmsRouteData, CmsRoute, CmsActivatedRouteSnapshot
core-libs/core/src/routing/routing.module.ts	RoutingModule core: store, serializer, APP/CONFIG/LOCATION initializer
core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts	LOCATION_INITIALIZED_MULTI, provideLocationInitializer(Factory)
core-libs/core/src/routing/protected-routes/protected-routes.service.ts	ProtectedRoutesService: quali URL sono protetti
core-libs/core/src/routing/protected-routes/protected-routes.guard.ts	ProtectedRoutesGuard: delega ad AuthGuard
core-libs/core/src/routing/external-routes/external-routes-config.ts	ExternalRoutesConfig (routing.internal)
core-libs/core/src/routing/external-routes/external-routes.service.ts	ExternalRoutesService: aggiunge in testa la route "esterna"

PATH	RUOLO
core-libs/core/src/routing/external-routes/external-routes.guard.ts	ExternalRoutesGuard: full page reload, ritorna sempre false
core-libs/core/src/routing/external-routes/external-routes.module.ts	ExternalRoutesModule.forRoot()
core-libs/storefront/cms-structure/routing/routing.module.ts	RoutingModule storefront: core + CmsRouteModule + default config + BEFORE_CMS_PAGE_GUARD
core-libs/storefront/cms-structure/routing/default-routing-config.ts	defaultRoutesConfigFactory: route di default (home, login, product, ...)
core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts	addCmsRoute: route ** con CmsPageGuard
core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts	getSuffixUrlMatcher (es. **/p/:productCode)
core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts	PRODUCT_DETAILS_URL_MATCHER
core-libs/storefront/cms-structure/guards/cms-page.guard.ts	CmsPageGuard
core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts	CmsPageGuardService: canActivatePage, canActivateNotFoundPage
core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts	BEFORE_CMS_PAGE_GUARD
core-libs/storefront/cms-structure/guards/before-cms-page-guard.service.ts	BeforeCmsPageGuardService
core-libs/storefront/cms-structure/services/guards-composer.ts	GuardsComposer: esecuzione sequenziale dei guard
core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts	CmsRoutesImplService: child routes generate dal CMS
core-libs/storefront/context/context.service.ts	ContextService
core-libs/storefront/context/routing-context.service.ts	RoutingContextService (data.cxContext)
core-libs/storefront/router/app-routing.module.ts	AppRoutingModule (RouterModule.forRoot)
core-libs/storefront/router/on-navigate.service.ts	OnNavigateService (reset scroll/focus)
core-libs/storefront/router/config/on-navigate-config.ts	OnNavigateConfig
core-libs/storefront/router/config/default-on-navigate-config.ts	defaultOnNavigateConfig

3. Come funziona passo per passo

3.1 Il modello di configurazione

`RoutingConfig` è una classe astratta fornita in `root` con `useExisting: Config`: legge dall'oggetto di configurazione globale (core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L22-L28). Il meccanismo è descritto nel cap. 11. `RoutingConfigDefinition` ha tre chiavi: `routes`, `protected`, `loadStrategy` (core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L30-L45). `ExternalRoutesConfig` aggiunge `internal` via augmentation (core-libs/core/src/routing/external-routes/external-routes-config.ts:L10-L24).

Per ogni route, `RouteConfig` contiene:

- `paths`: alias usati **sia** per generare i link **sia** per il matching, a meno che sia dato `matchers` (core-libs/core/src/routing/configurable-routes/routes-config.ts:L16-L22);
- `matchers`: `UrlMatcher` di Angular o `InjectionToken<UrlMatcherFactory>`. Se assenti si usano i `paths` (core-libs/core/src/routing/configurable-routes/routes-config.ts:L24-L31);
- `paramsMapping`, `disabled` (spegne il matcher ma lascia generare i link), `protected` (ha effetto solo `false`), `authFlow` (core-libs/core/src/routing/configurable-routes/routes-config.ts:L33-L57).

`RoutingConfigService` è il punto di lettura:

- `getRouteConfig(name)` legge `routing.routes[name]` e, solo in dev mode, avvisa se manca (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L26-L40);
- `getLoadStrategy()` ritorna `ALWAYS` di default (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L45-L47);
- `getRouteName(path)` usa una mappa inversa costruita la prima volta che serve (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L63-L68). Se un path è configurato per due route, l'ultima vince e l'errore compare solo in dev mode (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L77-L92).

La config di default arriva da `defaultRoutesConfigFactory`, registrata con `provideDefaultConfigFactory` (core-libs/storefront/cms-structure/routing/routing.module.ts:L26-L26). La factory legge tre feature toggle (core-libs/storefront/cms-structure/routing/default-routing-config.ts:L10-L15) e definisce:

- `home: ['']`, `notFound: ['not-found']`, `login` (`'sign-in'` o `'login'` a seconda di `authorizationCodeFlowByDefault`) e le altre route del flusso di login, tutte con `protected: false, authFlow: true` (core-libs/storefront/cms-structure/routing/default-routing-config.ts:L20-L59);
- le route PLP/PDP: `search`, `category` (`paramsMapping { categoryCode: 'code' }`), `brand`, `product` con due path e `paramsMapping { productCode: 'code' }` (core-libs/storefront/cms-structure/routing/default-routing-config.ts:L61-L73);
- `termsAndConditions`, `coupons`, `couponClaim`, `myInterests`, `notificationPreference` (core-libs/storefront/cms-structure/routing/default-routing-config.ts:L75-L86);
- in base ai toggle, `loginForm` e `oAuthCallback` (core-libs/storefront/cms-structure/routing/default-routing-config.ts:L90-L112).

Le altre route (carrello, checkout, my-account…) sono aggiunte dalle feature lib con il loro `provideDefaultConfig`. Il dettaglio è nei rispettivi capitoli.

3.2 Bootstrap: dalla config alle route Angular

1. **Route "vuote" con nome.** Le feature dichiarano route con `path: null` (con `// @ts-ignore`), un `canActivate: [CmsPageGuard]`, `component: PageLayoutComponent` e `data.cxRoute`. Esempi: `product` (core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L16-L35) e `cart` (feature-libs/cart/base/root/cart-base-root.module.ts:L64-L74).
2. **Router vuoto e bloccante.** `AppRoutingModule` registra `routes: []` con `initialNavigation: 'enabledBlocking'` (core-libs/storefront/router/app-routing.module.ts:L15-L18).
3. **Secure portal.** Se `routing.protected` non è definito nella config statica, un `CONFIG_INITIALIZER` (scope `'routing'`) lo ricava dal BaseSite (core-libs/core/src/routing/routing.module.ts:L44-L52, core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L16-L19). `resolveConfig()` prende la prima emissione (`take(1)`) del BaseSite attivo (`baseSiteService.get()` senza uid, core-libs/core/src/site-context/facade/base-site.service.ts:L63-L69), lancia un errore se è indefinito e mappa `requiresAuthentication → routing.protected` (core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L31-L50).
4. **Riscrittura delle route.** Un APP_INITIALIZER chiama `ConfigurableRoutesService.init()` (core-libs/core/src/routing/routing.module.ts:L93-L95, core-libs/core/src/routing/routing.module.ts:L37-L42). `init()` è idempotente (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L31-L43). `configure()` prende il `Router` dall'`Injector`, perché iniettarlo nel costruttore creerebbe una dipendenza ciclica con APP_INITIALIZER. Poi chiama `router.resetConfig(this.configureRoutes(router.config))` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L48-L52). La ricorsione scende anche nei `children` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L59-L68).
5. **Priorità in** `configureRoute` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L83-L106):
   1. `disabled` → matcher sempre falso (`getFalsy`);
   2. `matchers` → matcher combinato. Le funzioni sono usate così come sono, i token sono risolti con `injector.get(token)(route)` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L118-L143);
   3. esattamente un `path` → si usa il `path` nativo di Angular e si cancella `matcher`;
   4. altrimenti (0 o ≥ 2 path) → `urlMatcherService.getFromPaths(paths || [])`. Le route senza `data.cxRoute` passano invariate (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L108-L108, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L149-L151). In dev mode `validateRouteConfig` avvisa se config o `paths` sono `undefined`. `null` vale come spegnimento esplicito (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L153-L179).
6. **Wildcard CMS.** L'APP_INITIALIZER di `CmsRouteModule` (core-libs/storefront/cms-structure/routing/cms-route/cms-route.module.ts:L10-L20) fa `router.config.push({ path: '**', canActivate: [CmsPageGuard], component: PageLayoutComponent })` (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25).
7. **Route esterna (opzionale).** Se esistono pattern `routing.internal`, `ExternalRoutesService.addRoutes()` aggiunge **in testa** una route con matcher "opposto" al glob (core-libs/core/src/routing/external-routes/external-routes.service.ts:L35-L70).
8. **Navigazione iniziale.** Con `enabledBlocking`, Angular attende `LOCATION_INITIALIZED` prima di chiamare `router.initialNavigation()` (angular:packages/router/src/provide_router.ts:L441-L470). Spartacus sostituisce `LOCATION_INITIALIZED` con il `Promise.all` di tutti i `LOCATION_INITIALIZED_MULTI` (core-libs/core/src/routing/routing.module.ts:L54-L71). Tra questi: la stabilità della config (core-libs/core/src/

config/config-initializer/config-initializer.module.ts:L60-L64) e due initializer dell'auth (core-libs/core/src/auth/user-auth/user-auth.module.ts:L119-L120).

> DA VERIFICARE: l'ordine relativo degli APP_INITIALIZER dei passi 4, 6 e 7 dipende dall'ordine di import dei moduli. L'esito descritto (route esterna in testa, wildcard in coda) non dipende dall'ordine. `addCmsRoute` fa `push` e `addRoutes` fa prepend. Se però `addCmsRoute` girasse prima di `configure()`, la wildcard passerebbe da `configureRoute` e uscirebbe invariata, perché non ha `cxRoute`.

3.3 Il matching: `UrlMatcherService`

- `getFromPaths(paths)` crea un matcher per path e li combina. In dev mode aggiunge `_paths` per il debug (core-libs/core/src/routing/services/url-matcher.service.ts:L33-L40).
- `getCombined(matchers)` restituisce il primo risultato non nullo, nell'ordine dell'array (core-libs/core/src/routing/services/url-matcher.service.ts:L45-L58).
- `getFromPath(path)` è una copia adattata del `defaultUrlMatcher` di Angular. Prende il path dall'argomento invece che da `route.path` e gestisce anche `''` (core-libs/core/src/routing/services/url-matcher.service.ts:L65-L88). Rispetta `pathMatch: 'full'`, raccoglie `:param` in `posParams` e consuma `parts.length` segmenti (core-libs/core/src/routing/services/url-matcher.service.ts:L90-L121). Con path vuoto matcha senza consumare nulla, tranne quando `pathMatch: 'full'` e ci sono segmenti residui o il gruppo ha figli (core-libs/core/src/routing/services/url-matcher.service.ts:L128-L142).
- `getOpposite(m)` consuma tutto quando `m` **non** matcha, e restituisce `null` quando matcha (core-libs/core/src/routing/services/url-matcher.service.ts:L162-L176).
- `getFromGlob(patterns)` ricostruisce `/seg1/seg2` e lo passa a `GlobService.getValidator` (core-libs/core/src/routing/services/url-matcher.service.ts:L178-L197). Il validator richiede che almeno un pattern positivo matchi e nessun pattern negativo (core-libs/core/src/util/glob.service.ts:L17-L32).

Per comporre matcher senza perdere i `paths` configurati esiste `DEFAULT_URL_MATCHER`. È una factory che legge `route.data['cxRoute']` e restituisce `getFromPaths(paths)` (core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L13-L24, core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L35-L45). La PDP la usa: `PRODUCT_DETAILS_URL_MATCHER` combina il matcher di default con un suffix matcher `p/:productCode` (core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L16-L29). È anche il valore di default di `routing.routes.product.matchers` (core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L27-L35). `getSuffixUrlMatcher` cerca l'**ultima** occorrenza del marker, mette il segmento successivo in `paramName` e i precedenti in `param0..n` (core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L18-L48).

3.4 Generare URL: `SemanticPathService`, `cxUrl`, `RoutingService`

1. `{ cxRoute: 'product', params: product } | cxUrl` chiama `UrlPipe.transform`, che delega a `SemanticPathService.transform` (core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts:L11-L18). `cxProductUrl` è una scorciatoia per `{ cxRoute: 'product', params: product }` (core-libs/core/src/routing/configurable-routes/url-translation/product-url.pipe.ts:L10-L16).
2. `transform` normalizza i comandi in array. I comandi non-route passano invariati. Ogni comando `{cxRoute}` (truthy con `cxRoute` truthy, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L70-L72) è sostituito dai segmenti generati. Se **uno** non è risolvibile, il risultato è `ROOT_URL = ['/']` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L16-L16, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L41-L61).
3. `generateUrlPart` ritorna `null` se la route non ha config o `paths`, oppure se nessun path è soddisfacibile (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L78-

L111).

4. `findPathWithFillableParams` sceglie il **primo** path i cui parametri, dopo il `paramsMapping`, sono tutti `!== undefined` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L135-L154). Con i default della PDP, un prodotto con `name` produce `product/:productCode/:name`, senza `name` produce `product/:productCode`.
5. `provideParamsValues` sostituisce ogni `:param` con `params[mapping[param] || param]` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L117-L133, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L163-L171).
6. Se il primo comando era una route, il risultato diventa assoluto con `unshift('/')` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L63-L76).

`RoutingService` riusa lo stesso traduttore:

- `go(commands, extras)` → `router.navigate(transform(commands), extras)` (core-libs/core/src/routing/facade/routing.service.ts:L91-L94, core-libs/core/src/routing/facade/routing.service.ts:L167-L169);
- `goByUrl` → `router.navigateByUrl`, senza traduzione (core-libs/core/src/routing/facade/routing.service.ts:L131-L133);
- `getUrl` → `serializeUrl(createUrlTree(...))` con `/` iniziale garantito; `getFullUrl` vi antepone `winRef.location.origin` (core-libs/core/src/routing/facade/routing.service.ts:L101-L121);
- `back()` usa `location.back()` se `document.referrer` contiene (`includes`, test di sottostringa, non confronto di origin) l'origin corrente, altrimenti va a `['/']` (core-libs/core/src/routing/facade/routing.service.ts:L138-L157).

`SemanticPathService.get(name)` è la variante "solo stringa": `'/' + paths[0]` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L26-L31). La usano per esempio `AuthGuard` per `login` (core-libs/core/src/auth/user-auth/guards/auth.guard.ts:L30-L42) e il guard per `notFound` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L107-L121).

3.5 Lo stato del router in NgRx

Il `RoutingModule` di core registra la feature `'router'` e `StoreRouterConnectingModule.forRoot({ routerState: RouterState.Minimal, stateKey: 'router' })` (core-libs/core/src/routing/routing.module.ts:L73-L82). Nello stesso `forRoot()` sostituisce il serializer: `RouterStateSerializer → CustomSerializer` (core-libs/core/src/routing/routing.module.ts:L84-L92).

Chi vince tra i due serializer? `provideRouterStore` registra `RouterStateSerializer` con il serializer minimale di NgRx (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L351-L370). In Angular però i `providers` di un `ModuleWithProviders` vengono elaborati **dopo** quelli dei moduli che importa (angular:packages/core/src/di/provider_collection.ts:L300-L321). `R3Injector.processProvider` sovrascrive il record per token (angular:packages/core/src/di/r3_injector.ts:L473-L522). Vince quindi `CustomSerializer`.

Quando parte `ROUTER_NAVIGATION`? Spartacus non imposta `navigationActionTiming`, quindi vale il default `NavigationActionTiming.PreActivation` (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L102-L109). Il servizio di connessione di NgRx (`StoreRouterConnectingService.setUpRouterEventsListener`) mappa gli eventi del Router così (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L217-L256):

EVENTO ANGULAR	AZIONE NGRX	EFFETTO NEL REDUCER SPARTACUS
`NavigationStart`	ROUTER_REQUEST	nessuno (non gestita)

EVENTO ANGULAR	AZIONE NGRX	EFFETTO NEL REDUCER SPARTACUS
RoutesRecognized	ROUTER_NAVIGATION	nextState = snapshot target serializzato
NavigationCancel	ROUTER_CANCEL	nextState = undefined
NavigationError	ROUTER_ERROR	nextState = undefined
NavigationEnd	ROUTER_NAVIGATED	state = snapshot attivo, context preservato, nextState = undefined

Nel Router di Angular RoutesRecognized è emesso **prima** di GuardsCheckStart e dell'operatore checkGuards (angular:packages/router/src/navigation_transition.ts:L564-L572, angular:packages/router/src/navigation_transition.ts:L630-L649). Il JSDoc del modulo NgRx lo conferma: "before any guards or resolvers run" (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L375-L376). Per questo CmsPageGuard trova nextPageContext già pronto. ROUTER_NAVIGATION serializza lo snapshot target lastRoutesRecognized.state, mentre ROUTER_NAVIGATED serializza router.routerState.snapshot (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L260-L282). Con Minimal, l'event nel payload si riduce a { id, url, urlAfterRedirects } (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L283-L305).

Lo stato Spartacus estende quello di NgRx con nextState (core-libs/core/src/routing/store/routing-state.ts:L11-L16). Ogni snapshot ha url, queryParams, params, context, cmsRequired, semanticRoute (core-libs/core/src/routing/store/routing-state.ts:L18-L25). La feature è annidata due volte (router.router) (core-libs/core/src/routing/store/routing-state.ts:L27-L29, core-libs/core/src/routing/store/reducers/router.reducer.ts:L41-L45). Il reducer:

- ROUTER_NAVIGATION → nextState + navigationId, state invariato (core-libs/core/src/routing/store/reducers/router.reducer.ts:L52-L58);
- ROUTER_ERROR/ROUTER_CANCEL → nextState: undefined (core-libs/core/src/routing/store/reducers/router.reducer.ts:L60-L66);
- CHANGE_NEXT_PAGE_CONTEXT → sostituisce nextState.context se esiste un nextState (core-libs/core/src/routing/store/reducers/router.reducer.ts:L68-L75);
- ROUTER_NAVIGATED → state = snapshot, ma con context preso da nextState.context se presente. Serve a conservare un contesto cambiato durante la navigazione, come il not-found (core-libs/core/src/routing/store/reducers/router.reducer.ts:L77-L89).

I selettori espongono getPageContext (default { id: '' }), getNextPageContext e isNavigating = !!nextPageContext (core-libs/core/src/routing/store/selectors/routing.selector.ts:L15-L46). RoutingService li incapsula (core-libs/core/src/routing/facade/routing.service.ts:L49-L81). Il modello generale dello store è nel cap. 19.

3.6 Il calcolo del PageContext (CustomSerializer)

CustomSerializer.serialize scende lungo firstChild (core-libs/core/src/routing/store/reducers/router.reducer.ts:L112-L133):

1. accumula urlString, prende l'ultimo data.cxRoute (→ semanticRoute) e l'ultimo data.cxCmsRouteContext (→ context);
2. imposta cmsRequired = true se c'è un contesto CMS, oppure se una route ha in canActivate una funzione con guardName === 'CmsPageGuard' (core-libs/core/src/routing/store/reducers/router.reducer.ts:L135-L149). La proprietà statica è definita su CmsPageGuard (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L26-L27);

3. se manca `cxRoute`, cerca la route semantica con `getRouteName(urlString.substr(1))`. Funziona solo per URL statici (core-libs/core/src/routing/store/reducers/router.reducer.ts:L152-L155, core-libs/core/src/routing/store/reducers/router.reducer.ts:L222-L237);
4. calcola il contesto con `getPageContext` e restituisce `url`, `queryParams` della root, `params` della foglia, ecc. (core-libs/core/src/routing/store/reducers/router.reducer.ts:L156-L166).

`getPageContext`:

- `cx-preview` come primo segmento + `cmsTicketId` → `{ id: SMART_EDIT_CONTEXT, type: CONTENT_PAGE }` (core-libs/core/src/routing/store/reducers/router.reducer.ts:L175-L184);
- altrimenti, in ordine: `params.productCode` → `PRODUCT_PAGE`; `categoryCode` → `CATEGORY_PAGE`; `brandCode` → `CATEGORY_PAGE`; `data.pageLabel` → `CONTENT_PAGE` (core-libs/core/src/routing/store/reducers/router.reducer.ts:L185-L196);
- se ancora nessun contesto: `'/' + segmenti della foglia` come content page, oppure `HOME_PAGE_CONTEXT` se la foglia non ha segmenti (core-libs/core/src/routing/store/reducers/router.reducer.ts:L198-L217).

Le costanti valgono `HOME_PAGE_CONTEXT = '__HOMEPAGE__'` e `SMART_EDIT_CONTEXT = 'smartedit-preview'` (core-libs/core/src/routing/models/page-context.model.ts:L9-L20). `PageContext` è una classe con `id` e `type?` (core-libs/core/src/routing/models/page-context.model.ts:L22-L30). Come `CmsService` usa il contesto per caricare la pagina è spiegato nel cap. 15.

3.7 `CmsPageGuard`: decidere quale pagina mostrare

1. **Guard globali prima di tutto.** `BeforeCmsPageGuardService` esegue i guard registrati in `BEFORE_CMS_PAGE_GUARD` tramite `GuardsComposer` (core-libs/storefront/cms-structure/guards/before-cms-page-guard.service.ts:L17-L33). Lo storefront vi registra `ProtectedRoutesGuard` e `FederatedLoginGuard` (core-libs/storefront/cms-structure/routing/routing.module.ts:L27-L36). Il token è pensato **solo** per guard globali (core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L10-L24).
2. **Composizione.** `GuardsComposer` invoca subito `canActivate` di ogni guard (dentro `guards.map(...)`) e poi si sottoscrive ai risultati in sequenza con `concat` (core-libs/storefront/cms-structure/services/guards-composer.ts:L45-L48): eventuali effetti collaterali sincroni o Promise già avviate partono anche se un guard precedente restituisce `false`/`UrlTree`. Salta i `true` ed emette il primo risultato diverso da `true` (`false` o `UrlTree`), oppure `true` alla fine (core-libs/storefront/cms-structure/services/guards-composer.ts:L39-L86). Ogni guard è normalizzato con `wrapIntoObservable(...).pipe(first())` (core-libs/storefront/cms-structure/services/guards-composer.ts:L59-L69).
3. **Contesto in arrivo.** Se i before-guard danno `true`, il guard prende il primo `getNextPageContext()` definito (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L55-L60). Il valore è già presente grazie al timing di `ROUTER_NAVIGATION` (§3.5).
4. **Caricamento pagina.** `cmsService.getPage(pageContext, shouldReload())`, dove `shouldReload()` è `loadStrategy !== ONCE` (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L61-L62, core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L89-L91, core-libs/core/src/cms/facade/cms.service.ts:L274-L278).
5. **Pagina trovata → `canActivatePage`.** Prende i tipi di componente della pagina, ne determina i mapping e ne esegue i guard (`cmsComponents[...].guards`). Se l'esito è `true`, carica i chunk i18n. Se la route non ha già un `cxCmsRouteContext`, chiama `handleCmsRoutesInGuard` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L99). I guard dei componenti sono istanziati via `UnifiedInjector` (core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48).

6. **Child routes CMS.** Se i componenti dichiarano `childRoutes`, `CmsRoutesImplService` aggiunge in testa una route `{ path: label, component: PageLayoutComponent, children, data: { cxCmsRouteContext } }`, rinaviga allo stesso URL e ritorna `false` (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L124). Queste route vengono rimosse da `RouterEffects.clearCmsRoutes$` su `LANGUAGE_CHANGE`, `LOGIN`, `LOGOUT` (core-libs/core/src/routing/store/effects/router.effect.ts:L19-L37).
7. **Pagina non trovata → `canActivateNotFoundPage`.** Il guard risolve `notFound`; se non è configurato ritorna `false`. Carica la pagina not-found come `CONTENT_PAGE` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L107-L121), imposta il "fail index" della pagina richiesta, chiama `changeNextPageContext(notFound)`, attende l'aggiornamento dell'indice e prosegue con `canActivatePage` sui dati not-found (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L122-L142). L'URL resta quello richiesto. Il contesto not-found sopravvive a `ROUTER_NAVIGATED` grazie alla regola del reducer (§3.5).

3.8 Protected routes

- `ProtectedRoutesService.shouldProtect = !!routing.protected` (core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L24-L26). Se è vero già nel costruttore, il servizio pre-calcola i segmenti dei path con `protected === false` esplicito (core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L28-L38, core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L89).
- `isUrlProtected(segs) = shouldProtect && !matchAnyPath(segs, nonProtectedPaths)`, con `UrlParsingService.matchPath` (core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L43-L67). `matchPath` richiede lo stesso numero di segmenti e confronta solo quelli statici (core-libs/core/src/routing/configurable-routes/url-translation/url-parsing.service.ts:L28-L63).
- `ProtectedRoutesGuard` usa i segmenti di `route.url` (`['']` per la root). Se l'URL è protetto delega ad `AuthGuard.canActivate()` (core-libs/core/src/routing/protected-routes/protected-routes.guard.ts:L23-L33), che per un anonimo salva l'URL corrente e ritorna la `UrlTree` di `login` (core-libs/core/src/auth/user-auth/guards/auth.guard.ts:L30-L42).
- `authFlow: true` è letto da `AuthFlowRoutesService` (core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts:L23-L47) e da `FederatedLoginGuard` (core-libs/core/src/auth/user-auth/guards/federated-login.guard.ts:L21-L57), trattati nel cap. 22.

3.9 External routes

`ExternalRoutesModule.forRoot()` registra un APP_INITIALIZER (core-libs/core/src/routing/external-routes/external-routes.module.ts:L11-L29, core-libs/core/src/routing/external-routes/external-routes.providers.ts:L9-L16). Senza pattern `internal` non aggiunge nulla. Altrimenti aggiunge in testa una route `pathMatch: 'full'` con matcher `getOpposite(getFromGlob(internal))`, `canActivate: [ExternalRoutesGuard]` e `component: {} as any` (core-libs/core/src/routing/external-routes/external-routes.service.ts:L35-L70). Il guard, solo nel browser, imposta `window.location.href = state.url`, e ritorna **sempre** `false` (core-libs/core/src/routing/external-routes/external-routes.guard.ts:L22-L41).

3.10 Context dalle route

- `ContextService.get(token)` → `RoutingContextService.get(token)`, con `distinctUntilChanged` e `shareReplay` (core-libs/storefront/context/context.service.ts:L16-L37).
- `RoutingContextService` fonde con `Object.assign` i `data.cxContext` di tutte le route attive, dalla root alla foglia (core-libs/storefront/context/routing-context.service.ts:L31-L46). Risolve il provider token con

`UnifiedInjector.get`, oppure emette `undefined` (core-libs/storefront/context/routing-context.service.ts:L54-L63, core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68).

- Le route attive vengono da `ActivatedRoutesService.routes$`, che emette a ogni `NavigationEnd` più una volta all'iscrizione (core-libs/core/src/routing/services/activated-routes.service.ts:L22-L39). Sulla stessa sorgente `RoutingParamsService` fonde i `params` (core-libs/core/src/routing/facade/routing-params.service.ts:L20-L44).
- Esempio: la route `cart` mappa `ORDER_ENTRIES_CONTEXT` su `ActiveCartOrderEntriesContextToken` (feature-libs/cart/base/root/cart-base-root.module.ts:L64-L74).

3.11 Dopo la navigazione: `OnNavigateService`

`AppRoutingModule` fornisce `defaultOnNavigateConfig` e un APP_INITIALIZER che chiama `initializeWithConfig()` (core-libs/storefront/router/app-routing.module.ts:L22-L37). Il servizio si attiva solo con `active` truthy (core-libs/storefront/router/on-navigate.service.ts:L69-L73). Imposta la scroll restoration `manual` e ascolta gli eventi `Scroll` (core-libs/storefront/router/on-navigate.service.ts:L80-L119):

- back/forward (`position` presente) → ripristino della posizione;
- altrimenti esce subito se il path è uguale e `ignoreQueryString` è attivo, o se l'URL contiene un segmento di `ignoreRoutes`;
- altrimenti scrolla all'anchor (con `anchorScrolling === 'enabled'`) o a `[0,0]`, con `setTimeout` di 0 o 100 ms (core-libs/storefront/router/on-navigate.service.ts:L138-L163);
- infine sposta il focus su `selectedHostElement` (default `'body'`) o sull'host del componente bootstrap (core-libs/storefront/router/on-navigate.service.ts:L45-L57, core-libs/storefront/router/on-navigate.service.ts:L125-L131).

`previousScroll` sostituisce `pairwise` perché con l'hydration il primo evento `Scroll` è soppresso (core-libs/storefront/router/on-navigate.service.ts:L36-L43).

---

4. Diagrammi

4.1 Bootstrap del routing

■ **TAVOLA 14.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/router/app-routing.module.ts:L15-L18, core-libs/core/src/routing/routing.module.ts:L44-L52, core-libs/core/src/routing/routing.module.ts:L84-L108, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L48-L106, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25, core-libs/core/src/routing/external-routes/external-routes.service.ts:L35-L70, angular:packages/router/src/provide_router.ts:L441-L470, angular:packages/core/src/application/application_init.ts:L244-L255, angular:packages/core/src/application/application_init.ts:L263-L263.

4.2 Navigazione verso una pagina CMS: store, guard, reducer

■ **TAVOLA 14.2** diagramma a tutta pagina, in fondo al capitolo

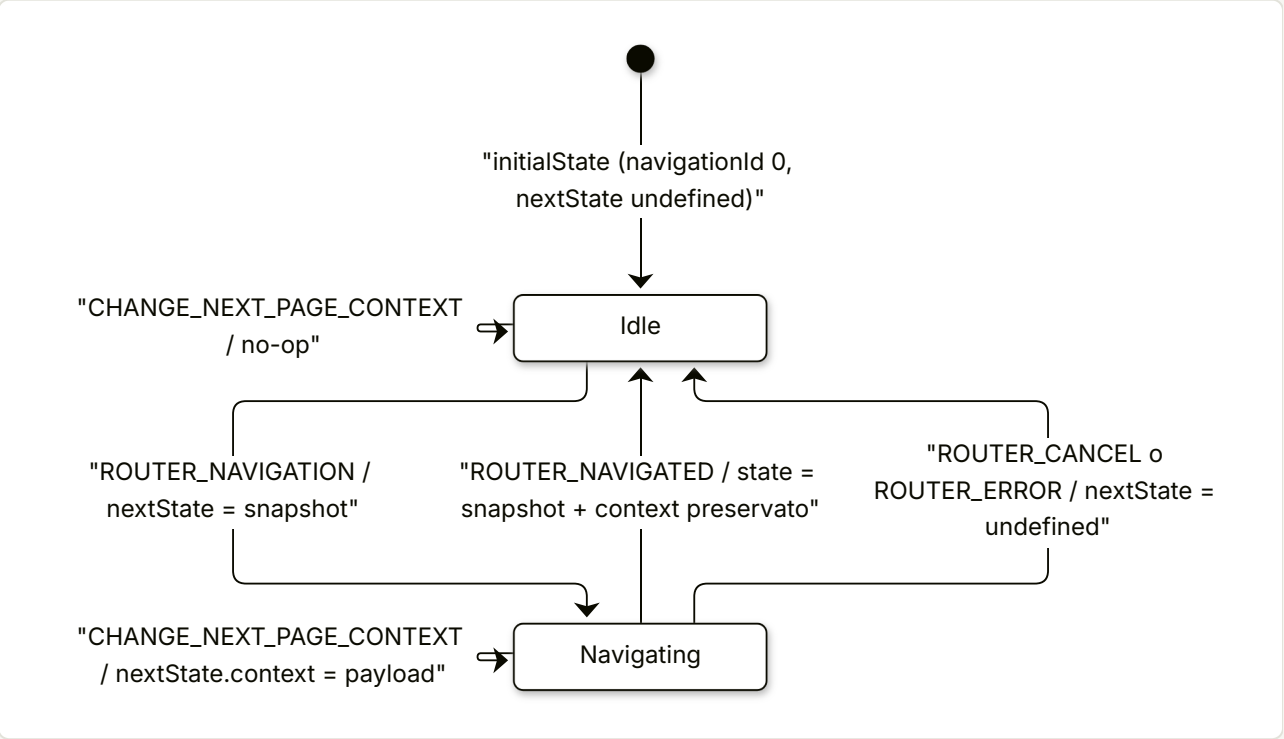
Riferimenti: deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L217-L282, angular:packages/router/src/navigation_transition.ts:L564-L572, angular:packages/router/src/navigation_transition.ts:L630-L649, core-libs/core/src/routing/store/reducers/router.reducer.ts:L52-L89, core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84, core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L142.

4.3 Risoluzione del `PageContext`

■ **TAVOLA 14.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/routing/store/reducers/router.reducer.ts:L120-L133, core-libs/core/src/routing/store/reducers/router.reducer.ts:L175-L217.

4.4 Ciclo di vita dello slice `router`



`isNavigating` è vero esattamente nello stato *Navigating* (core-libs/core/src/routing/store/selectors/routing.selector.ts:L34-L46). Riferimenti delle transizioni: core-libs/core/src/routing/store/reducers/router.reducer.ts:L26-L89.

---

5. Estratti di codice reale

5.1 Riscrittura di una route semantica

core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L109

```ts
protected configureRoute(route: Route): Route {
  const routeName = this.getRouteName(route);
  if (routeName) {
    const routeConfig = this.routingConfigService.getRouteConfig(routeName);
    this.validateRouteConfig(routeConfig, routeName, route);

    if (routeConfig?.disabled) {
      delete route.path;
      return {
        ...route,
        matcher: this.urlMatcherService.getFalsy(),
      };
    } else if (routeConfig?.matchers) {
      delete route.path;
      return {
        ...route,
        matcher: this.resolveUrlMatchers(route, routeConfig?.matchers),
      };
    } else if (routeConfig?.paths?.length === 1) {
      delete route.matcher;
      return { ...route, path: routeConfig?.paths[0] };
    } else {
      delete route.path;
      return {
        ...route,
        matcher: this.urlMatcherService.getFromPaths(
          routeConfig?.paths || []
        ),
      };
    }
  }
  return route; // if route doesn't have a name, just pass the original route
}
```

Nota il `delete route.path` sull'oggetto presente in `router.config` (già una copia shallow fatta da Angular), prima dello spread (vedi §9).

5.2 Traduzione dei comandi semantici

core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L41-L68

```ts
transform(commands: UrlCommands): any[] {
  if (!Array.isArray(commands)) {
    commands = [commands];
  }

  const result: (string | undefined)[] = [];
  for (const command of commands) {
```

```
      if (!this.isRouteCommand(command)) {
        // don't modify segment that is not route command:
        result.push(command);
      } else {
        // generate array with url segments for given route command:
        const partialResult = this.generateUrlPart(command);

        if (partialResult === null) {
          return this.ROOT_URL;
        }

        result.push(...partialResult);
      }
    }

    if (this.shouldOutputAbsolute(commands)) {
      result.unshift('/');
    }

    return result;
  }
```

5.3 Reducer del router con `nextState`

core-libs/core/src/routing/store/reducers/router.reducer.ts:L52-L89

TS

```
    case fromNgrxRouter.ROUTER_NAVIGATION: {
      return {
        ...state,
        nextState: action.payload.routerState,
        navigationId: action.payload.event.id,
      };
    }

    case fromNgrxRouter.ROUTER_ERROR:
    case fromNgrxRouter.ROUTER_CANCEL: {
      return {
        ...state,
        nextState: undefined,
      };
    }

    case CHANGE_NEXT_PAGE_CONTEXT: {
      return state.nextState
        ? {
            ...state,
            nextState: { ...state.nextState, context: action.payload },
          }
```

```
      : state;
  }

  case fromNgrxRouter.ROUTER_NAVIGATED: {
    return {
      state: {
        ...action.payload.routerState,
        context:
          // we want to preserve already resolved context,
          // in case it was changed while navigating
          state.nextState?.context ?? action.payload.routerState.context,
      },
      navigationId: action.payload.event.id,
      nextState: undefined,
    };
  }
```

5.4 Da dove arriva `ROUTER_NAVIGATION` (NgRx 21)

deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L232-L254

```JS
        else if (event instanceof RoutesRecognized) {
            routesRecognized = event;
            if (!dispatchNavLate && this.trigger !== RouterTrigger.STORE) {
                this.dispatchRouterNavigation(event);
            }
        }
        else if (event instanceof NavigationCancel) {
            this.dispatchRouterCancel(event);
            this.reset();
        }
        else if (event instanceof NavigationError) {
            this.dispatchRouterError(event);
            this.reset();
        }
        else if (event instanceof NavigationEnd) {
            if (this.trigger !== RouterTrigger.STORE) {
                if (dispatchNavLate) {
                    this.dispatchRouterNavigation(routesRecognized);
                }
                this.dispatchRouterNavigated(event);
            }
            this.reset();
        }
```

5.5 Il serializer riconosce le pagine CMS

core-libs/core/src/routing/store/reducers/router.reducer.ts:L120-L150

```ts
  while (state.firstChild) {
    state = state.firstChild as CmsActivatedRouteSnapshot;
    urlString +=
      '/' + state.url.map((urlSegment) => urlSegment.path).join('/');

    // we use semantic route information embedded from any parent route
    if (state.data?.cxRoute) {
      semanticRoute = state.data?.cxRoute;
    }

    // we use context information embedded in Cms driven routes from any parent route
    if (state.data?.cxCmsRouteContext) {
      context = state.data?.cxCmsRouteContext;
    }

    // we assume, that any route that has CmsPageGuard or it's child
    // is cmsRequired
    if (
      !cmsRequired &&
      (context ||
        state.routeConfig?.canActivate?.find(
          (x) =>
            x &&
            typeof x === 'function' &&
            'guardName' in x &&
            x.guardName === 'CmsPageGuard'
        ))
    ) {
      cmsRequired = true;
    }
  }
```

5.6 `CmsPageGuard.canActivate`

core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84

```ts
  canActivate(
    route: CmsActivatedRouteSnapshot,
    state: RouterStateSnapshot
  ): Observable<GuardResult> {
    return this.beforeCmsPageGuardService.canActivate(route, state).pipe(
      switchMap((canActivate) =>
        canActivate === true
          ? this.routingService.getNextPageContext().pipe(
              filter(isNotUndefined),
              take(1),
              switchMap((pageContext) =>
                this.cmsService.getPage(pageContext, this.shouldReload()).pipe(
```

```
                first(),
                switchMap((pageData) =>
                  pageData
                    ? this.service.canActivatePage(
                        pageContext,
                        pageData,
                        route,
                        state
                      )
                    : this.service.canActivateNotFoundPage(
                        pageContext,
                        route,
                        state
                      )
                )
              )
            )
          )
        : of(canActivate)
    )
  );
}
```

5.7 La route wildcard CMS

core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25

```ts
const cmsRoute: Route = {
  path: '**',
  canActivate: [CmsPageGuard],
  component: PageLayoutComponent,
};

export function addCmsRoute(injector: Injector): () => void {
  const result = () => {
    // Router could not be injected via `deps[]` due to a cyclic dependency with APP_INITIALIZER:
    const router = injector.get(Router);
    router.config.push(cmsRoute);
  };
  return result;
}
```

5.8 Route di default PLP/PDP

core-libs/storefront/cms-structure/routing/default-routing-config.ts:L61-L73

```ts
// plp routes
search: { paths: ['search/:query'] },
category: {
  paths: ['category/:categoryCode'],
  paramsMapping: { categoryCode: 'code' },
},
brand: { paths: ['Brands/:brandName/c/:brandCode'] },

// pdp routes
product: {
  paths: ['product/:productCode/:name', 'product/:productCode'],
  paramsMapping: { productCode: 'code' },
},
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes	RoutesConfig	route di §3.1 via provideDefaultConfigFactory	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L34-L34; default core-libs/storefront/cms-structure/routing/default-routing-config.ts:L17-L114	core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L26-L34; core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L80; core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts:L23-L32
routing.routes[x].paths	string[]	—	core-libs/core/src/routing/configurable-routes/routes-config.ts:L22-L22	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L104; core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L135-L154; core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L13-L24
routing.routes[x].matchers	(UrlMatcher | InjectionToken<UrlMatcherFactory>)[]	— (PDP: [PRODUCT_DETAILS_URL_MATCHER])	core-libs/core/src/routing/configurable-routes/routes-config.ts:L31-L31	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L89-L94
routing.routes[x].paramsMapping	ParamsMapping	— (category/product/couponClaim → code)	core-libs/core/src/routing/configurable-routes/routes-config.ts:L36-L36	core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L104-L108, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L141-L145
routing.routes[x].disabled	boolean	—	core-libs/core/src/routing/configurable-routes/routes-config.ts:L41-L41	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L83-L88

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes[x].protected	boolean (solo false conta)	— (route di login: false)	core-libs/core/src/routing/configurable-routes/routes-config.ts:L47-L47	core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L80
routing.routes[x].authFlow	boolean	— (route di login: true)	core-libs/core/src/routing/configurable-routes/routes-config.ts:L56-L56	core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts:L23-L32
routing.protected	boolean	undefined → ricavato da BaseSite.requiresAuthentication	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L39-L39	core-libs/core/src/routing/routing.module.ts:L44-L52; core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L24-L26
routing.loadStrategy	RouteLoadStrategy	ALWAYS (fallback nel servizio)	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L44-L44	core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L45-L47; core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L89-L91
routing.internal	string[] (glob che iniziano con / o !/)	— (nessuna route esterna)	core-libs/core/src/routing/external-routes/external-routes-config.ts:L10-L12	core-libs/core/src/routing/external-routes/external-routes.service.ts:L46-L70
enableResetViewOnNavigate.active	boolean	true	core-libs/storefront/router/config/on-navigate-config.ts:L16-L16; default core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16	core-libs/storefront/router/on-navigate.service.ts:L69-L73
enableResetViewOnNavigate.ignoreQueryString	boolean	false	core-libs/storefront/router/config/on-navigate-config.ts:L17-L17	core-libs/storefront/router/on-navigate.service.ts:L80-L119
enableResetViewOnNavigate.ignoreRoutes	string[]	[]	core-libs/storefront/router/config/on-navigate-config.ts:L18-L18	core-libs/storefront/router/on-navigate.service.ts:L171-L179
enableResetViewOnNavigate.selectedHostElement	string	'body'	core-libs/storefront/router/config/on-navigate-config.ts:L25-L25	core-libs/storefront/router/on-navigate.service.ts:L45-L57
Feature toggle authorizationCodeFlowByDefault	boolean	vedi cap. 42	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L161-L161	core-libs/storefront/cms-structure/routing/default-routing-config.ts:L10-L15
Feature toggle oauthCallbackPage	boolean	—	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L914-L914	core-libs/storefront/cms-structure/routing/default-routing-config.ts:L104-L104
Feature toggle asyncAuthConfigInitializer	boolean	—	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L637-L637	core-libs/storefront/cms-structure/routing/default-routing-config.ts:L105-L105
Opzioni Router anchorScrolling / initialNavigation	ExtraOptions di Angular	'enabled' / 'enabledBlocking'	core-libs/storefront/router/app-routing.module.ts:L15-L18	core-libs/storefront/router/on-navigate.service.ts:L31-L32; angular:packages/router/src/provide_router.ts:L441-L470

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
StoreRouter Connecting Module.forRoot	routerState / stateKey	Minimal / 'router' (timing Pre Activation implicito)	core-libs/core/src/routing/routing.module.ts:L77-L80	deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L102-L109
Route data.cx Route	string	—	(Route data di Angular)	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L149-L151; core-libs/core/src/routing/store/reducers/router.reducer.ts:L125-L128; core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L13-L24
Route data.cx Context	mappa ContextToken → token	—	(Route data di Angular)	core-libs/storefront/context/routing-context.service.ts:L31-L46
Route data.cxCms RouteContext / data.pageLabel	PageContext / string	—	core-libs/core/src/routing/models/cms-route.ts:L10-L13	core-libs/core/src/routing/store/reducers/router.reducer.ts:L130-L133, core-libs/core/src/routing/store/reducers/router.reducer.ts:L185-L196; core-libs/core/src/routing/store/effects/router.effect.ts:L19-L37; core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L86-L95

Il catalogo completo delle chiavi di config è nell'appendice A02.

7. Commenti e note notevoli dal sorgente

- *"guard not to call init() more than once"*: ConfigurableRoutesService è pensato per essere richiamato senza danni (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L31-L31).
- *"Router could not be injected in constructor due to cyclic dependency with APP_INITIALIZER"* (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L49-L49). Lo stesso workaround compare in addCmsRoute (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L20-L20).
- *"null value of routeConfig or routeConfig.paths means explicit switching off the route - it's valid config"* (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L159-L160).
- *"workaround for AOT compilation"* su initConfigurableRoutes (core-libs/core/src/routing/routing.module.ts:L40-L40).
- JSDoc su LOCATION_INITIALIZED: *"Only when the Promise in this injection token is resolved, then Angular will start the initial navigation in the Router."* (core-libs/core/src/routing/routing.module.ts:L54-L65).
- *"CAUTION: Don't use Router and don't perform any Router navigation in its implementations."* per i LocationInitializer (core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L30-L35). Inoltre l'injection context si perde dopo la parte sincrona (core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L16-L21).
- @license MIT di Google sul matcher copiato da Angular (core-libs/core/src/routing/services/url-matcher.service.ts:L76-L83).
- *"we assume, that any route that has CmsPageGuard or it's child is cmsRequired"* (core-libs/core/src/routing/store/reducers/router.reducer.ts:L135-L136).

- *"NOTE: It works only for simple static urls that are equal to the page label of cms-driven content page"* su `lookupSemanticRoute` (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L222-L232`).
- *"we want to preserve already resolved context, in case it was changed while navigating"* (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L82-L83`).
- Commento sul tipo `CONTENT_PAGE` "hardcoded" per la home (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L208-L214`).
- *"since 2211.24 not used anymore, but called indirectly via BeforeCmsPageGuardService"* sul parametro `protectedRoutesGuard` di `CmsPageGuard` (`core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L29-L38`).
- *"for content pages the page label returned from backend can be different than ID initially assumed from route"* (`core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L87-L87`).
- *"CAUTION: Suitable for global guards ONLY!"* su `BEFORE_CMS_PAGE_GUARD` (`core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L13-L20`).
- *"Public injection token for the private implementation... After #7070"* su `CmsRoutesService` (`core-libs/storefront/cms-structure/services/cms-routes.service.ts:L11-L12`).
- *"the external route should be activated only when it's NOT an internal route"* (`core-libs/core/src/routing/external-routes/external-routes.service.ts:L69-L69`).
- Il commento su hydration e `pairwise` in `OnNavigateService` (`core-libs/storefront/router/on-navigate.service.ts:L36-L42`).
- JSDoc di NgRx: *"During the navigation, before any guards or resolvers run, the router will dispatch a ROUTER_NAVIGATION action"* (`deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L375-L376`).

8. Extension point

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`routing.routes` (`RoutesConfig`)	`provideConfig({ routing: { routes: { product: { paths: [...] } } } })` per cambiare URL, aggiungere alias o pagine custom	`core-libs/core/src/routing/configurable-routes/routes-config.ts:L11-L61`	51 — route custom e page layout
Route Angular con `data.cxRoute`	Route `path: null` + `canActivate: [CmsPageGuard]` + `component: PageLayoutComponent` + `data: { cxRoute }`	`core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L16-L35`	51
`matchers` + `DEFAULT_URL_MATCHER` / `getSuffixUrlMatcher`	Matcher custom che non perde i `paths` di default (modello: `PRODUCT_DETAILS_URL_MATCHER`)	`core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L35-L45`; `core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L18-L55`; `core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L42-L50`	51
`UrlMatcherService`	Classe sostituibile via DI (metodi pubblici e `protected getFromPath`)	`core-libs/core/src/routing/services/url-matcher.service.ts:L17-L198`	50 — override di service via DI

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`BEFORE_CMS_PAGE_GUARD`	`{ provide: BEFORE_CMS_PAGE_GUARD, useExisting: MyGlobalGuard, multi: true }`, solo per guard globali	core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L10-L24	51
Guard per componente (`cmsComponents[...].guards`)	Alternativa a `BEFORE_CMS_PAGE_GUARD` per guard legati a una pagina	core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48	51
`routing.protected` + `protected: false`	Secure portal: storefront chiuso con eccezioni	core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L24-L89	50
`routing.internal` + `ExternalRoutesModule.forRoot()`	Migrazione graduale da un altro storefront	core-libs/core/src/routing/external-routes/external-routes.service.ts:L35-L70	50
`LOCATION_INITIALIZED_MULTI` / `provideLocationInitializer(Factory)`	Trattenere la navigazione iniziale finché una Promise non si risolve (senza usare il Router)	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L13-L62	50
`data.cxContext` + `ContextService`	Fornire un contesto ai componenti in base alla route (es. `ORDER_ENTRIES_CONTEXT`)	core-libs/storefront/context/routing-context.service.ts:L31-L63; feature-libs/cart/base/root/cart-base-root.module.ts:L64-L74	51
`ContextService.resolveContext` (protected)	Override per sorgenti di contesto diverse dalla route	core-libs/storefront/context/context.service.ts:L33-L37	50
`OnNavigateConfig`	Disattivare o regolare scroll/focus dopo la navigazione	core-libs/storefront/router/config/on-navigate-config.ts:L10-L31	50
`RoutingActions.ChangeNextPageContext` / `RoutingService.changeNextPageContext`	Cambiare la pagina CMS mostrata durante una navigazione (come fa il not-found)	core-libs/core/src/routing/facade/routing.service.ts:L72-L74	51
`UrlTestingModule` / `MockUrlPipe`	Test di componenti che usano `cxUrl` (non nel `public_api`; alias tsconfig negli spec)	core-libs/core/src/routing/configurable-routes/url-translation/testing/url-testing.module.ts:L11-L29; core-libs/storefront/tsconfig.spec.json:L29-L30	54 — API pubblica vs interna

Non sono nel public API: `path-utils`, `addExternalRoutesFactory`, `getDefaultUrlMatcherFactory`, `defaultRoutesConfigFactory`, `CustomSerializer`, il `reducer` e gli `effects` del router. Il barrel `core-libs/core/src/routing/index.ts` non esporta `store/reducers` né `store/effects` (core-libs/core/src/routing/index.ts:L7-L20, core-libs/core/src/routing/url-matcher/index.ts:L7-L8, core-libs/core/src/routing/external-routes/index.ts:L7-L10). Sostituirli significa dipendere da API interne: vedi cap. 54. Il catalogo dei guard è nell'appendice A05, quello degli injection token nell'appendice A01.

9. Trappole, casi limite, comportamenti controintuitivi

1. `configureRoute` **muta le route della config del Router.** `delete route.path` / `delete route.matcher` mutano l'oggetto presente in `router.config`, prima dello spread (core-libs/core/src/routing/configurable-

routes/configurable-routes.service.ts:L84-L99). Quell'oggetto è già una copia shallow fatta da Angular: il costruttore del `Router` chiama `resetConfig`, che applica `standardizeConfig` (`{...r}`) a ogni route (angular:packages/router/src/router.ts:L163-L163, angular:packages/router/src/router.ts:L181-L181, angular:packages/router/src/router.ts:L389-L393, angular:packages/router/src/components/empty_outlet.ts:L37-L39). L'oggetto dichiarato in `RouterModule.forChild` non viene toccato; restano condivisi solo i sotto-oggetti come `data`.

2. **Uno o più path cambiano il motore di matching.** Con un solo path si usa il `path` nativo di Angular. Con 0 o ≥ 2 si usa il matcher Spartacus, e `paths: []` non matcha mai (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L105, core-libs/core/src/routing/services/url-matcher.service.ts:L33-L58).
3. `matchers` **non sostituisce** `paths` **per i link.** La PDP fa il matching con `PRODUCT_DETAILS_URL_MATCHER` ma genera i link dai `paths` di default (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L89-L94, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L87-L97).
4. **Un solo comando non risolvibile azzera l'URL.** `transform` restituisce `['/']` e perde gli altri segmenti (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L55-L57). In dev mode compare un warning solo se la route non è configurata (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L29-L32); se la route esiste ma mancano i parametri richiesti da tutti i `paths`, `generateUrlPart` restituisce `null` senza alcun avviso (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L99-L102).
5. `standarizeRouteCommand` **muta l'input**, aggiungendo `params: {}` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L113-L115).
6. **L'ordine dei** `paths` **conta e** `null` **vale come presente.** Il test è `!== undefined` (core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L139-L147).
7. `cxUrl` **è una pipe pura.** Non dichiara `pure: false` (core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts:L11-L11) e Angular usa `pure: true` di default (angular:packages/core/src/metadata/directives.ts:L750-L756). Se si muta l'oggetto comando senza cambiarne il riferimento, il link non si aggiorna.
8. **Mappa inversa costruita una sola volta.** Modifiche alla config dopo la prima chiamata a `getRouteName` non sono viste (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L63-L68).
9. `ProtectedRoutesService` **calcola i path pubblici nel costruttore** e solo se `shouldProtect` è già vero (core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L28-L38). Se il servizio fosse istanziato con `routing.protected` ancora falso e il flag diventasse vero dopo, `nonProtectedPaths` resterebbe indefinito. `DA VERIFICARE`: in pratica il servizio nasce con `ProtectedRoutesGuard` durante la prima navigazione, quindi dopo la stabilità della config (core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64). Un'iniezione anticipata da codice custom non è stata analizzata.
10. `ProtectedRoutesGuard` **guarda** `route.url`**, non l'URL completo.** Con route annidate i segmenti sono quelli della route su cui gira il guard (core-libs/core/src/routing/protected-routes/protected-routes.guard.ts:L24-L27). Per la wildcard `**` coincidono con tutto l'URL residuo.
11. **Un** `protected: false` **con** `/` **iniziale non matcha mai**, perché `getSegments` divide per `/` senza normalizzare (core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L87-L89).
12. `routing.protected: false` **statico disattiva il secure portal del BaseSite** (core-libs/core/src/routing/routing.module.ts:L48-L51). Se il BaseSite manca, `SecurePortalConfigInitializer` lancia un errore (core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L33-L37).
13. `CmsPageGuard` **dipende dal router-store.** Il guard aspetta un `nextPageContext` definito (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L58-L60). In una navigazione normale arriva prima

dei guard (§3.5). Il caso tipico che lo lascia senza contesto è un `CmsPageGuard` invocato a mano fuori dal Router (o un'app che cambia `stateKey` o la struttura dello slice). La navigazione con trigger "store", in cui NgRx non dispatcha `ROUTER_NAVIGATION` (`deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L232-L236`), con la configurazione standard non si verifica: il trigger `STORE` è impostato solo in `navigateIfNeeded`, che esce sempre per lo slice annidato `router.router` (`deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L198-L214`, trappola 14). In questi casi il guard può non emettere mai.

14. **Modificare lo stato del router nello store non fa navigare.** Il connettore NgRx ascolta `select(stateKey)` e, in `navigateIfNeeded`, legge `store[stateKey].state.url` (`deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L191-L216`). Lo slice Spartacus però è annidato (`router.router`): la feature `'router'` è registrata con una `ActionReducerMap` `{ router: reducer }` e `stateKey: 'router'` (`core-libs/core/src/routing/routing.module.ts:L75-L80`, `core-libs/core/src/routing/store/reducers/router.reducer.ts:L41-L45`). Quindi `navigateIfNeeded` trova `state` indefinito ed esce subito (`deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L198-L201`): modificare lo stato del router nello store **non** provoca navigazioni. Chi si aspetta la navigazione "store-driven" di NgRx non la otterrà.
15. `cmsRequired` **riconosce solo** `CmsPageGuard` **"di classe".** Il controllo cerca `guardName` su una funzione in `canActivate`. Un guard funzionale che incapsula `CmsPageGuard` non viene riconosciuto (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L137-L149`).
16. **I param vincono su** `cxCmsRouteContext`**.** `productCode`/`categoryCode`/`brandCode`/`pageLabel` sovrascrivono un contesto CMS trovato più in alto (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L185-L196`).
17. **Il label di fallback usa solo la foglia**, mentre `lookupSemanticRoute` usa l'URL accumulato su tutta la catena (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L120-L123`, `core-libs/core/src/routing/store/reducers/router.reducer.ts:L198-L201`).
18. `ROUTER_NAVIGATED` **ricostruisce lo slice da zero.** Mantiene solo `state`, `navigationId`, `nextState` (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L77-L89`). Eventuali campi aggiunti allo slice vanno persi.
19. `ChangeNextPageContext` **fuori navigazione è un no-op silenzioso** (`core-libs/core/src/routing/store/reducers/router.reducer.ts:L68-L75`).
20. `clearCmsRoutes$` **pulisce solo il primo livello** di `router.config` (`core-libs/core/src/routing/store/effects/router.effect.ts:L27-L30`).
21. `handleCmsRoutesInGuard` **può annullare e rilanciare la navigazione.** Lancia `navigateByUrl(currentUrl)` e ritorna `false` (`core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L82-L88`). Nel log vedrete un `ROUTER_CANCEL` seguito da una nuova navigazione.
22. **La route esterna in SSR blocca e basta.** Ritorna `false` senza redirect (`core-libs/core/src/routing/external-routes/external-routes.guard.ts:L26-L29`).
23. `getFullUrl` **e** `back()` **in SSR.** `getFullUrl` lancia se mancano `SERVER_REQUEST_URL`/`SERVER_REQUEST_ORIGIN` (`core-libs/core/src/window/window-ref.ts:L70-L82`). `back()` senza `nativeWindow` va a `['/']` (`core-libs/core/src/routing/facade/routing.service.ts:L138-L148`).
24. **Param omonimi: vince il figlio** in `RoutingParamsService` (`core-libs/core/src/routing/facade/routing-params.service.ts:L43-L43`). `ActivatedRoutesService` segue solo `firstChild` e ignora gli outlet secondari (`core-libs/core/src/routing/services/activated-routes.service.ts:L31-L34`).
25. `ContextService` **può emettere più istanze**, una per ogni injector (root e lazy) in cui il token è registrato (`core-libs/storefront/context/routing-context.service.ts:L57-L60`, `core-libs/core/src/lazy-loading/unified-injector.ts:L56-L66`).

26. `ignoreRoutes` **confronta segmenti esatti in qualsiasi posizione.** `['my-account']` ignora ogni URL che contiene quel segmento (core-libs/storefront/router/on-navigate.service.ts:L171-L178). Lo scroll avviene in un `setTimeout` di 100 ms, di cui i test asincroni devono tenere conto (core-libs/storefront/router/on-navigate.service.ts:L155-L162).
27. **Suffix matcher e ultima occorrenza.** Su `/p/x/p/123` il risultato è `productCode = 123`, con `param0 = 'p'` e `param1 = 'x'` (core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L31-L45).

---

10. Come fare debug

- **Vedere la config effettiva del Router.** Dopo il bootstrap ispezionate `inject(Router).config`. Le route semantiche con matcher custom hanno, in dev mode, la proprietà `_paths` (core-libs/core/src/routing/services/url-matcher.service.ts:L36-L38). I suffix matcher hanno `_suffixRouteConfig` (core-libs/storefront/cms-structure/routing/suffix-routes/suffix-url-matcher.ts:L50-L52). La wildcard `**` deve essere l'ultima e la route esterna, se presente, precede tutte le route configurate, tranne le eventuali route CMS con child routes aggiunte in testa a runtime (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L119-L119).
- **Warning di dev mode.** "No path was configured for the named route" (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L30-L32), "Could not configure the named route" (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L153-L179) e l'errore sui path duplicati (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L84-L89) compaiono solo con `isDevMode()`. In produzione un link sbagliato diventa `/` senza avvisi.
- **Redux DevTools sullo slice** `router.router`**.** Seguite la sequenza `@ngrx/router-store/request` → `/navigation` (con `nextState.context`) → eventuale `[Router] Change Next PageContext` → `/navigated` oppure `/cancel`. I type sono definiti in `deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L12-L33` e core-libs/core/src/routing/store/actions/router.action.ts:L10-L17. Un `nextState` che resta valorizzato indica una navigazione appesa, tipicamente un guard che non emette.
- **Quale pagina CMS viene chiesta.** Leggete `nextState.context` (`id`, `type`) e confrontatelo con le regole di §3.6. Un `id` come `'/foo/bar'` di tipo `ContentPage` significa che nessun param/`pageLabel` ha prodotto un contesto e che si è usato il fallback della foglia (core-libs/core/src/routing/store/reducers/router.reducer.ts:L198-L203).
- **Eventi del Router.** Con `withDebugTracing()` o sottoscrivendo `router.events` verificate l'ordine `RoutesRecognized` → `GuardsCheckStart` → `GuardsCheckEnd` (angular:packages/router/src/navigation_transition.ts:L564-L660). Un `NavigationCancel` subito dopo `GuardsCheckEnd` con nuova `NavigationStart` sullo stesso URL è il segno delle child routes CMS (§9.21).
- **Protected routes.** Chiamate `ProtectedRoutesService.isUrlProtected(['my','path'])` dalla console o da un test. Ricordate che i segmenti vanno senza `/` iniziale (core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L43-L45).
- **Test.** `UrlTestingModule` stampa i comandi `cxUrl` come stringhe leggibili (`cxRoute:product code:123`) (core-libs/core/src/routing/configurable-routes/url-translation/testing/mock-url.pipe.ts:L20-L52). Con `allowlistParams` riduce il rumore (core-libs/core/src/routing/configurable-routes/url-translation/testing/url-testing.module.ts:L11-L29).

11. Domande di autoverifica

1. **Perché** `CmsPageGuard` **trova già valorizzato** `getNextPageContext()`**?** Perché `@ngrx/router-store` 21, con il timing di default `PreActivation`, dispatcha `ROUTER_NAVIGATION` su `RoutesRecognized`. Angular emette quell'evento prima di `GuardsCheckStart` e dei guard, e il reducer Spartacus vi salva `nextState`. — deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L102-L109, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L232-L236, angular:packages/router/src/navigation_transition.ts:L564-L572, core-libs/core/src/routing/store/reducers/router.reducer.ts:L52-L58.
2. **Una route con** `paths: ['a', 'b']` **usa il** `path` **nativo di Angular?** No. Il `path` nativo si usa solo con esattamente un path. Con due path si genera un matcher combinato con `getFromPaths`. — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L105.
3. **Che URL produce** `{ cxRoute: 'product', params: { code: '42' } } | cxUrl` **con la config di default?** `['/', 'product', '42']`. Il primo path richiede `name`, che manca, quindi si usa `product/:productCode`. `productCode` è mappato su `code` e il risultato diventa assoluto perché il primo comando è una route. — core-libs/storefront/cms-structure/routing/default-routing-config.ts:L61-L73, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L63-L65, core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L135-L154.
4. **Come fa la pagina not-found a mantenere l'URL richiesto e a mostrare il contenuto** `notFound`**?** `canActivateNotFoundPage` dispatcha `ChangeNextPageContext(notFound)` e attiva la route originale. Su `ROUTER_NAVIGATED` il reducer usa `nextState.context` al posto del contesto ricalcolato. — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L122-L142, core-libs/core/src/routing/store/reducers/router.reducer.ts:L77-L89.
5. **Quale serializer usa davvero lo store, visto che** `StoreRouterConnectingModule.forRoot` **ne registra uno minimale?** `CustomSerializer`. Il provider di `RoutingModule.forRoot()` è elaborato dopo quelli dei moduli importati e sovrascrive il record del token `RouterStateSerializer`. — core-libs/core/src/routing/routing.module.ts:L84-L92, angular:packages/core/src/di/provider_collection.ts:L300-L321, angular:packages/core/src/di/r3_injector.ts:L473-L522.
6. **Con** `routing.protected: true`**, come si rende pubblica una pagina custom?** Si imposta `protected: false` esplicito sulla sua `RouteConfig`, con path senza `/` iniziale. `undefined` è ignorato. — core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L89.
7. **Dove registrereste un guard che deve girare su tutte le pagine CMS prima del caricamento della pagina? E uno solo per una pagina?** Il primo va in `BEFORE_CMS_PAGE_GUARD` (multi). Il secondo va in `cmsComponents[tipo].guards` di un componente della pagina, come indica la nota "CAUTION" del token. — core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L10-L24, core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48.
8. **Perché le route CMS con child routes spariscono dopo un cambio lingua o un login?** `RouterEffects.clearCmsRoutes$` rimuove su `LANGUAGE_CHANGE`/`LOGIN`/`LOGOUT` le route con `data.cxCmsRouteContext` e chiama `resetConfig`. Alla navigazione successiva `CmsPageGuard` le ricrea. — core-libs/core/src/routing/store/effects/router.effect.ts:L19-L37, core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L93-L124.
9. **Quando parte la prima navigazione dell'app?** Dopo che si è risolta `LOCATION_INITIALIZED`, che Spartacus sostituisce con il `Promise.all` dei `LOCATION_INITIALIZED_MULTI`: config stabile e initializer dell'auth. Lo prevede `initialNavigation: 'enabledBlocking'`. — core-libs/core/src/routing/routing.module.ts:L54-L71, angular:packages/router/src/provide_router.ts:L441-L470, core-libs/core/src/config/config-initializer/config-initializer.module.ts:L60-L64.

10. **Che** `PageContext` **ha l'URL** `/` **della home?** `{ id: '__HOMEPAGE__', type: CONTENT_PAGE }`: la foglia non ha segmenti e non ci sono param né `pageLabel`. — core-libs/core/src/routing/store/reducers/router.reducer.ts:L198-L217, core-libs/core/src/routing/models/page-context.model.ts:L9-L20.

TAVOLA 14.1 **4.1 Bootstrap del routing**

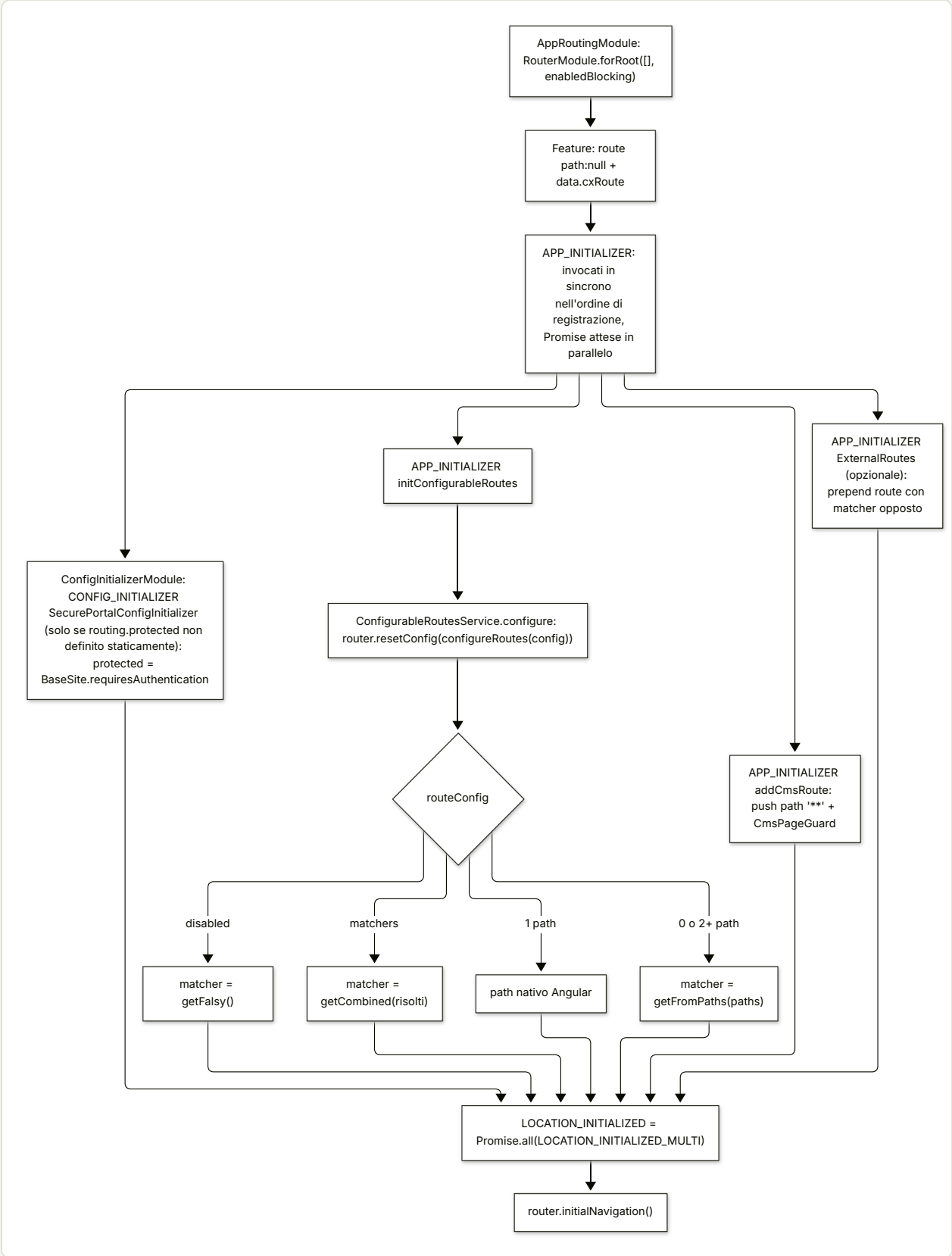


TAVOLA 14.2 **4.2 Navigazione verso una pagina CMS: store, guard, reducer**

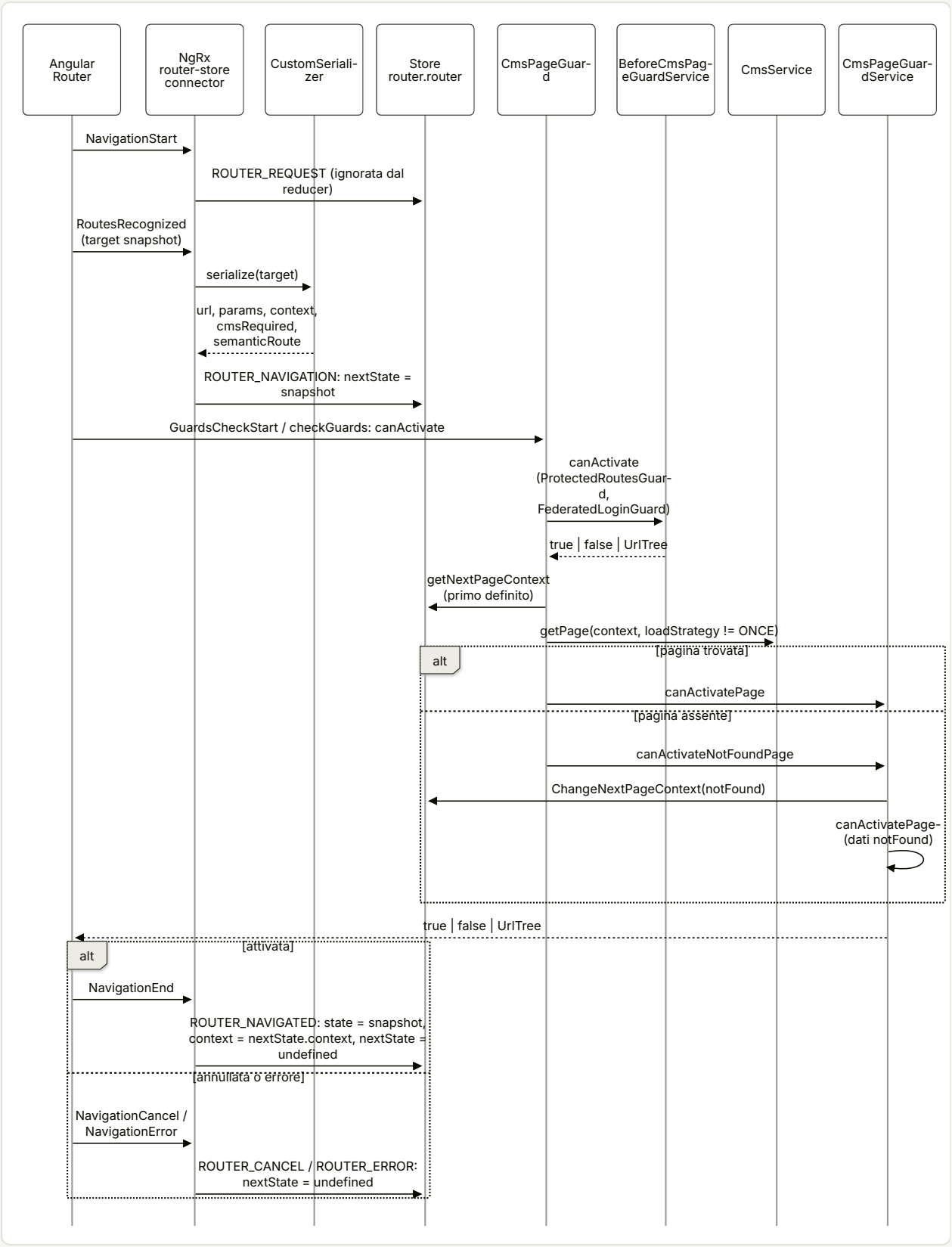
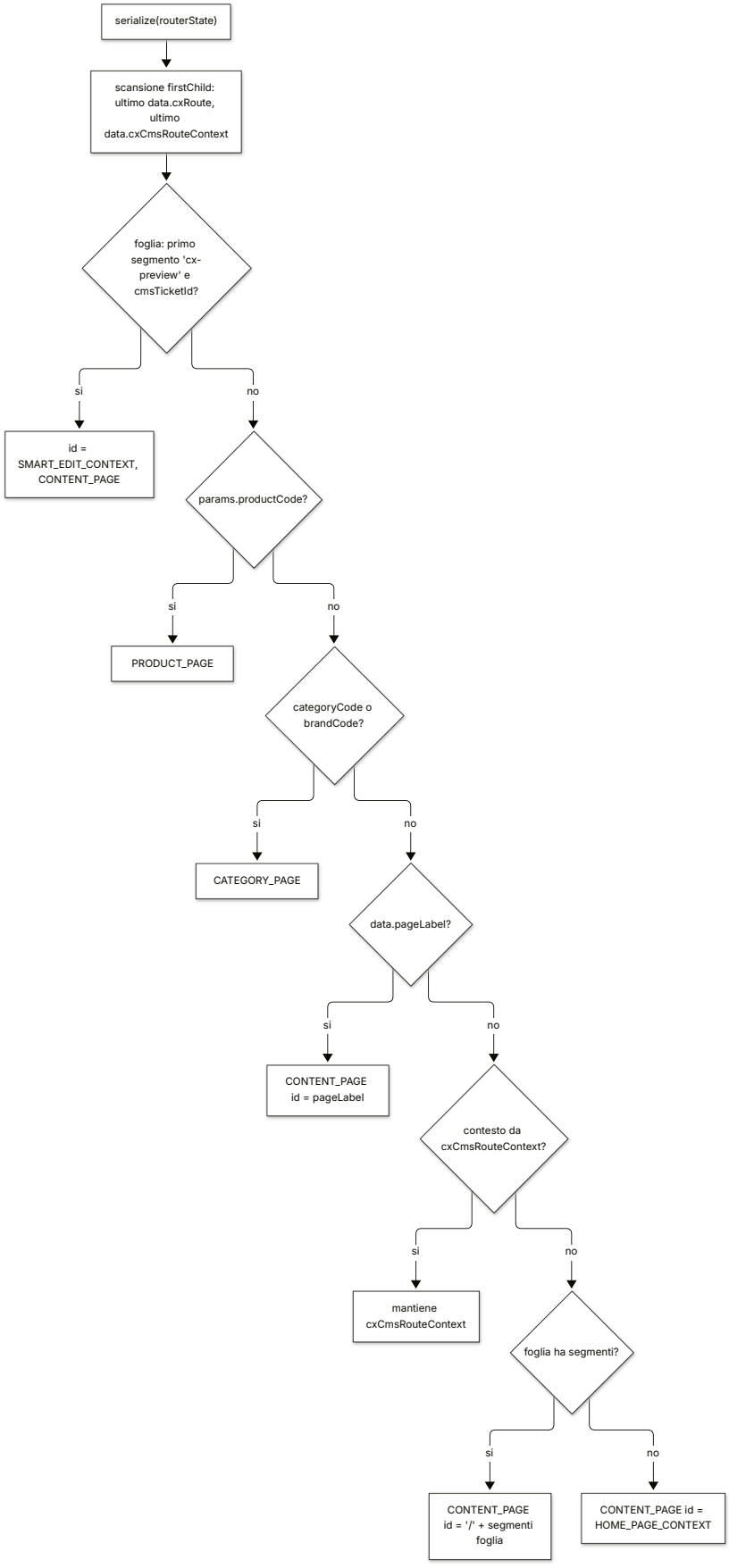


TAVOLA 14.3 **4.3 Risoluzione del PageContext**



15

CMS (core): pagine, slot, componenti, caricamento

Struttura di pagina da `PageContext`, separazione struttura/dati, caricamento accorpato dei componenti, `CmsService`.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Questo capitolo descrive la metà "dati" del CMS di Spartacus, quella che vive in `@spartacus/core`: come una pagina CMS viene richiesta, normalizzata, fusa con la struttura statica, salvata nello store NgRx e riletta; come i dati dei singoli componenti vengono caricati a blocchi; come i decorator permettono a SmartEdit di agganciarsi. La metà "rendering" (`PageLayoutComponent`, `PageSlotComponent`, `ComponentWrapperDirective`, mapping verso le classi Angular) è nel capitolo 16 — CMS (storefront): rendering dinamico delle pagine. Il dettaglio SEO dei file `cms/page/**` è nel capitolo 40 — SEO e PWA: qui compaiono solo come contorno.

---

1. A cosa serve e perché esiste

Spartacus è uno storefront "guidato dal CMS": la quasi totalità delle pagine non è definita da route Angular statiche, ma da una struttura che arriva da SAP Commerce Cloud (pagina → template → slot → componenti). Il modulo `cms` di core risolve quattro problemi distinti:

1. **Ottenere la struttura di una pagina** a partire da un `PageContext` (id + `PageType`), indipendentemente dalla sorgente: backend OCC, configurazione statica, o un CMS di terze parti tramite un adapter custom. Il connettore di pagina accetta sia l'adapter sia la config statica (`core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L30-L54`), e l'adapter astratto dichiara esplicitamente lo scopo "third-party CMS" (`core-libs/core/src/cms/connectors/page/cms-page.adapter.ts:L11-L24`).
2. **Separare struttura e dati dei componenti.** Il modello di risposta normalizzato è la coppia `{ page, components }` (`core-libs/core/src/cms/model/page.model.ts:L28-L34`): lo slot contiene solo riferimenti leggeri (`uid`, `typeCode`, `flexType`, `properties`) (`core-libs/core/src/cms/model/content-slot-component-data.model.ts:L7-L12`), mentre i dati completi stanno in un ramo separato dello store, indicizzato per `uid` (`core-libs/core/src/cms/store/cms-state.ts:L22-L62`).
3. **Caricare i dati dei componenti in modo efficiente.** Quando un componente non è arrivato con la pagina, le richieste emesse nello stesso turno dell'event loop vengono accorpate in un'unica chiamata di rete, come promesso dal JSDoc di `getComponentData` (`core-libs/core/src/cms/facade/cms.service.ts:L61-L73`) e realizzato dall'effect `loadComponent$` con `bufferDebounceTime` (`core-libs/core/src/cms/store/effects/components.effect.ts:L41-L65`).
4. **Definire il contratto di configurazione** con cui lo storefront mappa i tipi CMS su componenti Angular (`cmsComponents`) e su feature lazy (`featureModules`). La *definizione* è in core (`core-libs/core/src/cms/config/cms-config.ts:L124-L138`); la *lettura* è nello storefront (`core-libs/storefront/cms-structure/services/cms-components.service.ts:L67-L67`, `core-libs/storefront/cms-structure/services/cms-features.service.ts:L60-L60`).

Il tutto è esposto tramite un'unica facade, `CmsService` (`core-libs/core/src/cms/facade/cms.service.ts:L33-L46`), che i consumatori (guard, componenti, SmartEdit) usano senza toccare direttamente NgRx.

Dove viene attivato

- `CmsModule` importa `CmsStoreModule` e `PageMetaModule.forRoot()`; il suo `forRoot()` fornisce `CmsService` e la config di default `defaultCmsModuleConfig` (`core-libs/core/src/cms/cms.module.ts:L14-L24`).
- `CmsModule.forRoot()` è importato da `BaseCoreModule` (`core-libs/core/src/base-core.module.ts:L34-L34`).
- I binding OCC (adapter di pagina e componenti, normalizer di pagina) **non** sono in `CmsModule`, ma in `CmsOccModule` (`core-libs/core/src/occ/adapters/cms/cms-occ.module.ts:L16-L34`), importato da `BaseOccModule` (`core-libs/core/src/occ/base-occ.module.ts:L18-L18`). È la separazione standard connector/adapter descritta nel capitolo 21 — Livello OCC.

2. Mappa dei file

Tutti i path sotto `core-libs/core/src/cms/` salvo dove indicato.

PATH	RUOLO
core-libs/core/src/cms/cms.module.ts	`CmsModule` e `forRoot()`: importa store e page meta, fornisce `CmsService` e default config
core-libs/core/src/cms/index.ts	Barrel pubblico: non esporta `utils/`, effects né reducers
core-libs/core/src/cms/config/cms-config.ts	`CmsConfig`, `CmsComponentMapping`, `CMSComponentConfig`, `FeatureModuleConfig`, `DeferLoadingStrategy`, costanti JSP/Flex
core-libs/core/src/cms/config/cms-structure.config.ts	`CmsStructureConfig`, `CmsPageConfig`, `CmsPageSlotConfig`, `CmsPageSlotsConfig`
core-libs/core/src/cms/config/default-cms-config.ts	`defaultCmsModuleConfig`: 4 endpoint OCC, `pageSize: 50`
core-libs/core/src/cms/connectors/page/cms-page.adapter.ts	`CmsPageAdapter` astratta
core-libs/core/src/cms/connectors/page/cms-page.connector.ts	`CmsPageConnector`: `ignoreBackend`, 400 → `{}`, merge statico
core-libs/core/src/cms/connectors/page/converters.ts	`CMS_PAGE_NORMALIZER`
core-libs/core/src/cms/connectors/component/cms-component.adapter.ts	`CmsComponentAdapter` astratta
core-libs/core/src/cms/connectors/component/cms-component.connector.ts	`CmsComponentConnector`: config statica + paginazione delle richieste
core-libs/core/src/cms/connectors/component/converters.ts	`CMS_COMPONENT_NORMALIZER` (nessun provider di default)
core-libs/core/src/cms/decorators/component-decorator.ts	`ComponentDecorator` astratta (hook DOM componente)
core-libs/core/src/cms/decorators/slot-decorator.ts	`SlotDecorator` astratta (hook DOM slot)
core-libs/core/src/cms/facade/cms.service.ts	`CmsService`: facade pagine/slot/componenti/navigazione, refresh, cache componenti
core-libs/core/src/cms/facade/page-meta.service.ts	`PageMetaService` (contorno, cap. 40)
core-libs/core/src/cms/model/page.model.ts	`Page`, `CmsStructureModel`, `PageMeta`, `BreadcrumbMeta`, `PageRobotsMeta`
core-libs/core/src/cms/model/content-slot-data.model.ts	`ContentSlotData`
core-libs/core/src/cms/model/content-slot-component-data.model.ts	`ContentSlotComponentData`
core-libs/core/src/cms/model/node-item.model.ts	`NodeItem`
core-libs/core/src/cms/services/cms-structure-config.service.ts	`CmsStructureConfigService`: merge delle strutture statiche

PATH	RUOLO
core-libs/core/src/cms/services/dynamic-attribute.service.ts	`DynamicAttributeService`: applica i decorator
core-libs/core/src/cms/store/cms-state.ts	Tipi di stato, `CMS_FEATURE`, entity type
core-libs/core/src/cms/store/cms-store.module.ts	`CmsStoreModule`: feature store, effects, SSR transfer
core-libs/core/src/cms/store/actions/page.action.ts	Azioni pagina e indice
core-libs/core/src/cms/store/actions/components.action.ts	Azioni componenti
core-libs/core/src/cms/store/actions/navigation-entry-item.action.ts	Azioni navigazione
core-libs/core/src/cms/store/effects/page.effect.ts	`PageEffects`: load pagina, refresh su cambio contesto
core-libs/core/src/cms/store/effects/components.effect.ts	`ComponentsEffects`: batch/debounce per contesto
core-libs/core/src/cms/store/effects/navigation-entry-item.effect.ts	`NavigationEntryItemEffects`
core-libs/core/src/cms/store/reducers/index.ts	`getReducers`, `reducerToken`, `clearCmsState`, `metaReducers`
core-libs/core/src/cms/store/reducers/page-data.reducer.ts	`Page` per `pageId`
core-libs/core/src/cms/store/reducers/page-index.reducer.ts	Indice contesto → `pageId`, per tipo
core-libs/core/src/cms/store/reducers/components.reducer.ts	`ComponentsContext` per uid
core-libs/core/src/cms/store/reducers/navigation-entry-item.reducer.ts	`NodeItem` per nodo
core-libs/core/src/cms/store/selectors/page.selectors.ts	Selettori pagina, indice, slot, tipi
core-libs/core/src/cms/store/selectors/components.selectors.ts	Selettori componenti (tri-stato)
core-libs/core/src/cms/store/selectors/navigation-entry-item.selectors.ts	Selettori navigazione
core-libs/core/src/cms/utils/cms-utils.ts	`serializePageContext`, `CURRENT_CONTEXT_KEY` (non pubblici)
core-libs/core/src/cms/page/	Page meta resolver, canonical, breadcrumb (contorno, cap. 40)
core-libs/core/src/occ/adapters/cms/cms-occ.module.ts	Binding OCC degli adapter e del normalizer
core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts	`OccCmsPageAdapter`
core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts	`OccCmsComponentAdapter`
core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts	`LegacyOccCmsComponentAdapter` (POST, backend pre-1905)

PATH	RUOLO
core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts	OccCmsPageNormalizer
core-libs/storefront/cms-structure/utils/cms-structure.util.ts	provideCmsStructure()
core-libs/storefront/recipes/config/static-cms-structure.ts	Provider statici esportati defaultCmsContentProviders (HamburgerMenuComponent, LoginComponent), inclusi dall'app
core-libs/storefront/cms-structure/guards/cms-page.guard.ts	CmsPageGuard: principale consumatore di getPage()
feature-libs/smartedit/core/services/smart-edit.service.ts	Refresh pagina/componenti da SmartEdit
feature-libs/smartedit/core/decorators/index.ts	Decorator SmartEdit (multi-provider)

3. Come funziona passo per passo

3.1 Il modello dati: pagina, slot, riferimento, componente

Il modello normalizzato ha quattro livelli:

- Page: pageId, name, type, title, template, slots (mappa posizione → ContentSlotData), properties, label, robots e altri campi (core-libs/core/src/cms/model/page.model.ts:L10-L26).
- ContentSlotData: { components?: ContentSlotComponentData[]; properties?: any } (core-libs/core/src/cms/model/content-slot-data.model.ts:L9-L12).
- ContentSlotComponentData: riferimento leggero { uid, typeCode, flexType, properties } (core-libs/core/src/cms/model/content-slot-component-data.model.ts:L7-L12).
- CmsComponent: i dati veri del componente, con uid, typeCode, name, modifiedTime, otherProperties, composition.inner, styleClasses (core-libs/core/src/model/cms.model.ts:L7-L32).

La risposta di un caricamento pagina è CmsStructureModel = { page?: Page; components?: CmsComponent[] } (core-libs/core/src/cms/model/page.model.ts:L28-L34).

Il contesto che identifica una pagina è PageContext { id: string; type?: PageType }, con due id speciali: HOME_PAGE_CONTEXT = '__HOMEPAGE__' e SMART_EDIT_CONTEXT = 'smartedit-preview' (core-libs/core/src/routing/models/page-context.model.ts:L15-L30). PageType ha quattro valori: ContentPage, ProductPage, CategoryPage, CatalogPage (core-libs/core/src/model/cms.model.ts:L38-L42). Come il router produce il PageContext è argomento del capitolo 14 — Routing.

3.2 La chiave del contesto: serializePageContext

Quasi tutto il modulo indicizza i dati per "contesto serializzato". La funzione serializePageContext restituisce 'current' se il contesto è assente, solo '<type>' se ignoreContentPageId è vero e il tipo è ContentPage, altrimenti '<type>-<id>' (core-libs/core/src/cms/utils/cms-utils.ts:L10-L34). La variante "senza id" serve perché un componente su una ContentPage non dipende dalla specifica content page (a differenza di PDP/PLP, dove l'id del prodotto/categoria entra nei parametri della richiesta — vedi 3.10). Questa funzione **non** è API pubblica: core-libs/core/src/cms/index.ts non riesporta utils/ (core-libs/core/src/cms/index.ts:L7-L21).

3.3 Lo store `cms`

`CmsStoreModule` registra la feature `'cms'` con il reducer fornito via token, i meta-reducer e gli effects (core-libs/core/src/cms/store/cms-store.module.ts:L33-L45). La forma dello stato è:

- `CmsState = { page: PageState; components: ComponentsState; navigation: EntityLoaderState<NodeItem> }`, con `PageState = { pageData: EntityState<Page>; index: IndexType }` (core-libs/core/src/cms/store/cms-state.ts:L75-L84).
- `IndexType` ha quattro `EntityLoaderState<string>` (content, product, category, catalog) (core-libs/core/src/cms/store/cms-state.ts:L64-L69): ogni ramo mappa `pageContext.id` → `pageId`, con il loader state (loading/success/error) dell'ultimo caricamento.
- `ComponentsContext` (per uid) contiene **un solo** `component` e una mappa `pageContext[context] → LoaderState<boolean>` il cui `value` indica se il componente esiste per quel contesto (core-libs/core/src/cms/store/cms-state.ts:L22-L62).

`getReducers()` compone il tutto: `pageData` è un reducer semplice; ogni ramo dell'indice è un `entityLoaderReducer` con entity type uguale al `PageType` corrispondente; `components` è un `entityReducer` su `COMPONENT_ENTITY`; `navigation` un `entityLoaderReducer` su `NAVIGATION_DETAIL_ENTITY` (core-libs/core/src/cms/store/reducers/index.ts:L33-L65). I reducer sono forniti via `InjectionToken('CmsReducers')` con `useFactory: getReducers` (core-libs/core/src/cms/store/reducers/index.ts:L67-L73). Le utility generiche (`loaderReducer`, `entityReducer`, `entityLoaderReducer`) sono descritte nel capitolo 19 — State management.

Due proprietà trasversali dello store:

- **Transfer state SSR.** L'intero ramo `cms` è trasferito dal server al client con `StateTransferType.TRANSFER_STATE`; la config è prodotta da una factory solo per poter usare la costante `CMS_FEATURE` come chiave (core-libs/core/src/cms/store/cms-store.module.ts:L21-L31).
- **Reset su cambio contesto utente.** Il meta-reducer `clearCmsState` azzera l'intero stato CMS su `LANGUAGE_CHANGE`, `LOGOUT` e `LOGIN` (core-libs/core/src/cms/store/reducers/index.ts:L75-L90).

3.4 Caricare una pagina: dalla guard a `hasPage`

Il consumatore principale è `CmsPageGuard`: prende il `nextPageContext`, chiama `cmsService.getPage(pageContext, shouldReload())` e, a seconda che la pagina esista, delega a `canActivatePage` o `canActivateNotFoundPage`; `shouldReload()` è vero se la load strategy di routing non è `ONCE` (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L91).

`CmsService.getPage` è `hasPage(...)` seguito da `getPageState(...)` se la pagina esiste, altrimenti `of(null)` (core-libs/core/src/cms/facade/cms.service.ts:L274-L283). Il cuore è `hasPage` (core-libs/core/src/cms/facade/cms.service.ts:L247-L269):

1. seleziona il loader state dell'indice per il contesto (`getPageStateIndexLoaderState`);
2. in un `tap`, se il caricamento non è mai stato tentato (`!loading && !success && !error`) oppure se `forceReload` è vero e non c'è un caricamento in corso, dispatcha `LoadCmsPageData(pageContext)` e azzera `forceReload` (mutando il parametro, così le emissioni successive non ri-dispatchano);
3. scarta gli stati privi della proprietà `value` — il caso di uno stato incompleto trasferito da un SSR fallito (core-libs/core/src/cms/facade/cms.service.ts:L258-L263);
4. lascia passare solo stati conclusi (`success`, oppure `error && !loading`) e li mappa a `!!success`; un errore nello stream diventa `false`.

3.5 L'azione `LoadCmsPageData` e l'indice

Le azioni pagina sono entity-loader action il cui *entity type* è il `PageType` del contesto (`payload.type ?? ''`) e il cui *entity id* è `pageContext.id` (core-libs/core/src/cms/store/actions/page.action.ts:L18-L24). Di conseguenza `LoadCmsPageData` mette `loading: true` solo nel ramo indice del tipo corrispondente. I tipi d'azione sono `[Cms] Load Page Data` (+ Fail, Success), `[Cms] Set Page Success Index`, `[Cms] Set Page Fail Index` (core-libs/core/src/cms/store/actions/page.action.ts:L12-L16).

3.6 `PageEffects.loadPageData$`

L'effect raggruppa i `LoadCmsPageData` per contesto serializzato (forma completa, **senza** ignorare l'id delle content page) e dentro ogni gruppo usa `switchMap` su `CmsPageConnector.get(pageContext)` (core-libs/core/src/cms/store/effects/page.effect.ts:L58-L66). Effetto pratico: due richieste per lo stesso contesto si annullano a vicenda (vince l'ultima), richieste per contesti diversi procedono in parallelo.

A successo emette, nell'ordine (core-libs/core/src/cms/store/effects/page.effect.ts:L67-L93):

1. opzionalmente `CmsSetPageSuccessIndex({ id: pageLabel, type }, page)`, anteposto con `unshift` se la `label` restituita dal backend differisce dall'id del contesto: la pagina viene registrata anche sotto la label corretta (core-libs/core/src/cms/store/effects/page.effect.ts:L81-L91);
2. `CmsGetComponentFromPage` con tutti i componenti inclusi nella risposta, ciascuno accoppiato al contesto;
3. `LoadCmsPageDataSuccess(pageContext, page ?? {})`.

In errore emette `LoadCmsPageDataFail(pageContext, tryNormalizeHttpError(error, logger))` (core-libs/core/src/cms/store/effects/page.effect.ts:L95-L102).

3.7 `CmsPageConnector.get`: backend, config statica, 400

Il connettore chiede a `CmsStructureConfigService.shouldIgnoreBackend(pageContext.id)`: se falso chiama `CmsPageAdapter.load`, altrimenti parte da `{}`; in entrambi i casi fonde la struttura statica con `mergePageStructure` (core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L30-L54). Un `HttpErrorResponse` con status 400 dall'adapter viene convertito in struttura vuota `{}`; gli altri errori sono rilanciati (core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L37-L46). `shouldIgnoreBackend` è vero solo se in `cmsStructure.pages` esiste una pagina con quel `pageId` e `ignoreBackend` truthy (core-libs/core/src/cms/services/cms-structure-config.service.ts:L56-L60, core-libs/core/src/cms/services/cms-structure-config.service.ts:L84-L92).

3.8 L'adapter OCC e il normalizer

`OccCmsPageAdapter.load` ottiene lo user id, costruisce l'URL con l'endpoint `page` (by id) se il contesto non ha `type`, altrimenti con `pages` e query params; fa la GET e passa la risposta per `CMS_PAGE_NORMALIZER`, con `take(1)` (core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L48-L66). I query params sono (core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L86-L102):

- nessuno per `HOME_PAGE_CONTEXT` e `SMART_EDIT_CONTEXT`;
- `pageType` = tipo del contesto;
- `pageLabelOrId=id` per `ContentPage`, altrimenti `code=id`.

`ConverterService.convertSource` applica in sequenza (`reduce`) tutti i converter multi-provider del token (core-libs/core/src/util/converter.service.ts:L130-L137). Per le pagine il default è `OccCmsPageNormalizer`, che esegue quattro passi in ordine (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L22-L35):

1. `normalizePageData`: `typeCode → type`, `uid → pageId`, più `name`, `label`, `template`, `title`, `description`, `properties`, e `normalizeRobots` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L40-L77);
2. `normalizePageSlotData`: forza `contentSlot` ad array se arriva come oggetto singolo e crea uno slot per ogni `position`, copiando le `properties` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L82-L105);
3. `normalizePageComponentData`: per ogni componente dello slot aggiunge un `ContentSlotComponentData {uid, typeCode, flexType, properties}`, ma solo se `components.component` è un array (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L110-L139);
4. `normalizeComponentData`: copia i dati completi dei componenti in `target.components`, rinomina `modifiedtime → modifiedTime` e azzera `properties`, che restano solo nel riferimento di slot (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L166-L193).

Il campo più importante prodotto dal normalizer è `flexType`: vale `component.flexType` per `CMSFlexComponent`, `component.uid` per `JspIncludeComponent`, altrimenti `typeCode` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L151-L158). È la chiave con cui lo storefront cerca il mapping in `cmsComponents` (cap. 16). Le costanti dei due tipi speciali sono in `cms-config.ts` (core-libs/core/src/cms/config/cms-config.ts:L42-L43).

`normalizeRobots` converte il `robotTag` OCC (es. `INDEX_FOLLOW`, `NOINDEX_NOFOLLOW`) in una coppia di `PageRobotsMeta`; senza tag produce `[]` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L198-L222).

3.9 Merge della struttura statica

`CmsStructureConfigService.mergePageStructure(pageId, structure)` esegue `mergePage` (slot della pagina configurata) e poi `mergeSlots(page)` senza argomento, cioè con gli slot globali `cmsStructure.slots` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L38-L45, core-libs/core/src/cms/services/cms-structure-config.service.ts:L135-L138).

- `mergePage`: se c'è una pagina configurata con quel `pageId` e la struttura non ha `page`, la crea copiando la config e svuotando `slots`; poi fonde gli slot configurati della pagina (core-libs/core/src/cms/services/cms-structure-config.service.ts:L99-L122).
- `mergeSlots`: per ogni posizione configurata, aggiunge lo slot **solo se la posizione non è già presente nella pagina**, anche se lo slot del CMS è vuoto. Il JSDoc lo dichiara: il CMS deve poter sovrascrivere la configurazione statica. Per ogni componente aggiunge il riferimento allo slot e il componente completo in `components` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L124-L167).
- `getComponentsByPosition` risolve i `componentIds` dello slot in `cmsStructure.components[id]`, aggiungendo `uid: componentId` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L169-L186).

La config `CmsStructureConfig` (classe astratta, alias di `Config`) aggiunge `cmsStructure.{components, pages, slots}` (core-libs/core/src/cms/config/cms-structure.config.ts:L68-L89). Lo storefront fornisce un helper, `provideCmsStructure({componentId, pageSlotPosition, ...})`, che genera `cmsStructure.components[componentId] = {typeCode, flexType}` e `cmsStructure.slots[pageSlotPosition] = {componentIds: [componentId]}` più una config di layout, tramite `provideConfig` (core-libs/storefront/cms-structure/utils/cms-structure.util.ts:L31-L66). Lo storefront esporta `defaultCmsContentProviders` (`HamburgerMenuComponent` in `PreHeader`, `LoginComponent` in `SiteLogin`), che l'app include nei propri provider: lo schematic `add-spartacus` lo aggiunge al modulo di configurazione generato, e l'app demo fa lo stesso (core-libs/storefront/recipes/config/static-cms-structure.ts:L10-L19, core-libs/schematics/src/add-spartacus/configuration.ts:L113-L119, projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L48-L48).

3.10 Reducer e indice dopo il successo

Una volta emesse le azioni di successo:

- `page-data.reducer` salva la `Page` in `entities[page.pageId ?? '']`, senza guardare l'entity type (core-libs/core/src/cms/store/reducers/page-data.reducer.ts:L13-L27).
- `page-index.reducer(entityType)` agisce solo se `meta.entityType === entityType`: su Success e `CmsSetPageSuccessIndex` scrive `payload.pageId`, su Fail torna a `undefined`, su `CmsSetPageFailIndex` scrive il `payload` (stringa) (core-libs/core/src/cms/store/reducers/page-index.reducer.ts:L28-L46). Il valore finisce nel loader state perché, su `success`, `loaderReducer` calcola `value` con il sub-reducer (core-libs/core/src/state/utils/loader/loader.reducer.ts:L53-L60, core-libs/core/src/state/utils/loader/loader.reducer.ts:L80-L82).
- `CmsGetComponentFromPage` porta come entity id l'**array** degli uid (core-libs/core/src/cms/store/actions/components.action.ts:L65-L80); `entityReducer`, quando sia `meta.entityId` sia `payload` sono array, passa al reducer interno un'azione con `payload: action.payload[i]` per ogni id (core-libs/core/src/state/utils/entity/entity.reducer.ts:L40-L53). Il components reducer riceve così un singolo `{component, pageContext}` e scrive sia `component` sia `pageContext[ctx]` con `value: true` (core-libs/core/src/cms/store/reducers/components.reducer.ts:L71-L87).

La lettura avviene in due passi: `getPageData(pageContext)` combina il valore dell'indice (contesto → `pageId`) con `pageData.entities[indexValue]` (core-libs/core/src/cms/store/selectors/page.selectors.ts:L88-L96). Il ramo indice è scelto da `getIndexByType` in base al `PageType`; per tipo assente o ignoto restituisce `{ entities: {} }` (core-libs/core/src/cms/store/selectors/page.selectors.ts:L17-L38).

Infine `hasPage` vede `success` ed emette `true`; `getPage` passa a `getPageState` e la guard procede con `canActivatePage`, che legge `getPageComponentTypes(pageContext)`, risolve i mapping, valuta le guard dei componenti, carica i chunk i18n e registra le child route (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L99). `getPageComponentTypes` raccoglie in un `Set` i `flexType` di tutti i componenti di tutti gli slot (core-libs/core/src/cms/store/selectors/page.selectors.ts:L40-L52).

3.11 Pagina non trovata: l'indice di fallback

Se la pagina non esiste, `canActivateNotFoundPage` carica la pagina `notFound` come ContentPage, ne legge l'indice, lo assegna al contesto richiesto con `setPageFailIndex(pageContext, notFoundIndex)`, cambia il `nextPageContext` e attende che `getPageIndex(pageContext)` coincida con l'indice della 404 (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L107-L143). `CmsSetPageFailIndex` porta come payload il `pageId` della pagina di fallback e crea un `Error('Failed to set cms page index')` (core-libs/core/src/cms/store/actions/page.action.ts:L53-L69): nell'indice del tipo originale, la entry dell'URL richiesto punta quindi alla pagina 404, con loader state in errore. Un'interessante conseguenza è che `getPageData` per l'URL inesistente restituisce i dati della 404 senza duplicarli in `pageData`.

3.12 Dati di un componente: `getComponentData`

`getComponentData(uid, pageContext?)` (core-libs/core/src/cms/facade/cms.service.ts:L74-L95):

1. per `uid === ''` restituisce `of(null)`;
2. calcola il contesto serializzato con `ignoreContentPageId = true`;
3. memoizza un observable per `uid` e contesto nella cache privata `components[uid][context]` (core-libs/core/src/cms/facade/cms.service.ts:L37-L41).

L'observable è creato da `createComponentData` (core-libs/core/src/cms/facade/cms.service.ts:L97-L146):

- **senza contesto**: attende il `PageContext` corrente non nullo e richiama `getComponentData` con quel contesto (core-libs/core/src/cms/facade/cms.service.ts:L101-L108);
- **con contesto**: costruisce `loading$` combinando `getNextPageContext()` e il loader state del componente per il contesto, osservati su `queueScheduler`. Dispatcha `LoadCmsComponent({uid, pageContext})` solo se non è mai stato tentato un caricamento **e** il contesto richiesto non coincide con quello della navigazione in corso — in quel caso i dati potrebbero arrivare a breve con la pagina (core-libs/core/src/cms/facade/cms.service.ts:L112-L135);
- il dato emesso è `componentsSelectorFactory(uid, context)` filtrato da `undefined`; `using()` lega la sottoscrizione di `loading$` alla vita di `component$`, e il tutto è condiviso con `shareReplay({bufferSize: 1, refCount: true})` (core-libs/core/src/cms/facade/cms.service.ts:L137-L145).

Il selettore è **tri-stato**: restituisce il componente se esiste per il contesto, `null` se è stato tentato ma non esiste (o è ristretto), `undefined` se lo stato è ignoto (core-libs/core/src/cms/store/selectors/components.selectors.ts:L67-L94). Poiché `CmsService` filtra solo `undefined`, il consumatore può ricevere `null`. L'esistenza per contesto è letta da `componentsLoaderStateSelectorFactory`, che ripiega su `initialLoaderState` (core-libs/core/src/cms/store/selectors/components.selectors.ts:L26-L38).

3.13 Batching: `ComponentsEffects.loadComponent$`

`LoadCmsComponent` ha payload `{uid, pageContext?}` ed entity id = `uid` (core-libs/core/src/cms/store/actions/components.action.ts:L19-L63). Il components reducer, su Load e Fail, aggiorna solo `pageContext[context]` con un `loaderReducer` interno il cui sub-reducer `componentExistsReducer` dà `false` su Fail e `true` su Success/GetFromPage (core-libs/core/src/cms/store/reducers/components.reducer.ts:L18-L55).

L'effect `loadComponent$` è definito come funzione con parametri `{scheduler, debounce = 0}` (core-libs/core/src/cms/store/effects/components.effect.ts:L41-L65):

1. `groupBy` per contesto serializzato (forma completa);
2. dentro ogni gruppo `bufferDebounceTime(debounce, scheduler)` accumula le azioni e le emette come array dopo `debounceTime`, poi svuota il buffer (core-libs/core/src/util/rxjs/buffer-debounce-time.ts:L10-L24);
3. `loadComponentsEffect(uids, ctx ?? { id: '' })` carica il blocco con una sola chiamata;
4. `withdrawOn(this.contextChange$)` reinizializza tutto il flusso su `LANGUAGE_CHANGE`/`LOGOUT`/`LOGIN` (core-libs/core/src/cms/store/effects/components.effect.ts:L33-L39): `withdrawOn(notifier)` è `notifier.pipe(startWith(undefined), switchMap(() => source))` (core-libs/core/src/util/rxjs/withdraw-on.ts:L10-L27).

NgRx invoca l'effect "funzione" senza argomenti, quindi `debounce = 0` e `scheduler` indefinito: `bufferDebounceTime(0)` raccoglie le azioni dello stesso turno (core-libs/core/src/cms/store/effects/components.effect.ts:L41-L46). `createEffect` senza `functional` chiama `source()` e salva come effect la funzione restituita (deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L85-L95); `mergeEffects` la invoca poi come `sourceInstance[propertyName]()` (deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L153-L155).

`loadComponentsEffect` chiama `CmsComponentConnector.getList(uids, pageContext)`, emette un `LoadCmsComponentSuccess` per ogni componente ricevuto e un `LoadCmsComponentFail` per ogni uid richiesto ma assente dalla risposta; se la chiamata fallisce, emette `LoadCmsComponentFail` per tutti gli uid del blocco con l'errore normalizzato (core-libs/core/src/cms/store/effects/components.effect.ts:L74-L117).

3.14 `CmsComponentConnector.getList` e l'adapter OCC

`getList` (core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L41-L86):

1. chiede a `CmsStructureConfigService.getComponentsFromConfig(ids)` i componenti statici (array allineato agli id, `undefined` per i mancanti) (core-libs/core/src/cms/services/cms-structure-config.service.ts:L75-L79, core-libs/core/src/cms/services/cms-structure-config.service.ts:L188-L192);
2. calcola gli id mancanti;
3. li divide in pagine da `componentsLoading.pageSize` (default 50; fallback: tutti in una pagina se il valore è falsy);
4. lancia una `findComponentsByIds` per pagina, le combina con `zip` e restituisce prima i componenti da config e poi quelli caricati.

`OccCmsComponentAdapter.findComponentsByIds` fa una GET sull'endpoint `components` con `componentIds` (lista separata da virgole), `fields='DEFAULT'`, `currentPage=0`, `pageSize=ids.length` e i parametri di contesto; estrae `componentList.component ?? []` e applica `pipeableMany(CMS_COMPONENT_NORMALIZER)` (core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts:L51-L78). I parametri di contesto sono `productCode`/`categoryCode`/`catalogCode` per Product/Category/Catalog page, nessuno per ContentPage (core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts:L132-L152) — ecco perché la cache usa la chiave `'ContentPage'` senza id.

Nel repo non esiste alcun provider per `CMS_COMPONENT_NORMALIZER`, e `pipeableMany` restituisce lo stream invariato se il token non ha converter (core-libs/core/src/util/converter.service.ts:L93-L101): i componenti OCC arrivano quindi nello store così come sono.

Per backend pre-1905 esiste `LegacyOccCmsComponentAdapter`, che carica i componenti con POST di `{ idList }` e va fornito esplicitamente per il token `CmsComponentAdapter` (core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts:L16-L56).

Il metodo `CmsComponentConnector.get` (singolo componente: config statica, altrimenti `adapter.load`) esiste (core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L26-L39), ma il caricamento runtime dei componenti passa solo da `getList`.

3.15 Slot e navigazione

- `getContentSlot(position)` usa il `PageContext` corrente e `getCurrentSlotSelectorFactory`, filtrando `undefined` (core-libs/core/src/cms/facade/cms.service.ts:L152-L165). Il selettore restituisce lo slot, oppure `{ components: [] }` se la pagina esiste ma lo slot no, oppure `undefined` se la pagina non c'è (core-libs/core/src/cms/store/selectors/page.selectors.ts:L105-L114).
- `loadNavigationItems(rootUid, itemList)` dispatcha `LoadCmsNavigationItems({nodeId, items})`; `getNavigationEntryItems(nodeUid)` legge dallo store (core-libs/core/src/cms/facade/cms.service.ts:L171-L192).
- `NavigationEntryItemEffects` divide gli item per `superType` (`AbstractCMSComponent`, `AbstractPage`, `AbstractMedia`), legge il contesto dallo stato del router (`take(1)`) e carica solo i componenti con una sola `getList` (core-libs/core/src/cms/store/effects/navigation-entry-item.effect.ts:L35-L63, core-libs/core/src/cms/store/effects/navigation-entry-item.effect.ts:L82-L102). Senza componenti emette `LoadCmsNavigationItemsFail(nodeId, new Error('navigation nodes are empty'))` (core-libs/core/src/cms/store/effects/navigation-entry-item.effect.ts:L70-L77).
- Il reducer fonde ogni componente nello stato del nodo sotto la chiave `` `${uid}_AbstractCMSComponent` `` (core-libs/core/src/cms/store/reducers/navigation-entry-item.reducer.ts:L16-L34); `NodeItem` è proprio una mappa `[id_type: string]: CmsLinkComponent` (core-libs/core/src/cms/model/node-item.model.ts:L9-L11).

3.16 Refresh e invalidazione

- `refreshLatestPage()` dispatcha `LoadCmsPageData` col contesto corrente (`take(1)`); `refreshPageById(pageId)` dispatcha `LoadCmsPageData({ id: pageId })` senza `type` (core-libs/core/src/cms/facade/cms.service.ts:L197-L213).
- `refreshComponent(uid, pageContext?)` dispatcha direttamente `LoadCmsComponent`, senza passare dalla cache del service (core-libs/core/src/cms/facade/cms.service.ts:L221-L223).
- `clearComponentState(uids?)` rimuove le entry della cache locale e dispatcha `ClearCmsComponent` per ogni uid (tutti quelli in cache se non specificati) (core-libs/core/src/cms/facade/cms.service.ts:L295-L309); `ClearCmsComponent` estende `EntityLoaderResetAction` (core-libs/core/src/cms/store/actions/components.action.ts:L82-L88) e il reducer riporta il `ComponentsContext` a `initialState` (core-libs/core/src/cms/store/reducers/components.reducer.ts:L13-L40).
- `PageEffects.refreshPage$`: su `LANGUAGE_CHANGE`/`LOGOUT`/`LOGIN` attende uno stato router con `cmsRequired` e senza `nextState`, poi ri-dispatcha `LoadCmsPageData(state.context)` (core-libs/core/src/cms/store/effects/page.effect.ts:L34-L56).
- In aggiunta, il costruttore di `PageMetaService` chiama `cms.refreshLatestPage()` a ogni cambio di lingua attiva, saltando la prima emissione (core-libs/core/src/cms/facade/page-meta.service.ts:L42-L50).

3.17 Decorator e hook SmartEdit

`ComponentDecorator` e `SlotDecorator` sono classi astratte `@Injectable()` con `decorate(element, renderer, data?)` (core-libs/core/src/cms/decorators/component-decorator.ts:L10-L23, core-libs/core/src/cms/decorators/slot-decorator.ts:L10-L23). `DynamicAttributeService` raccoglie i decorator multi-provider via `UnifiedInjector.getMulti` (con `shareReplay(1)`) e li applica sincronamente con `getLastValueSync` in `addAttributesToComponent`/`addAttributesToSlot` (core-libs/core/src/cms/services/dynamic-attribute.service.ts:L20-L64). `getMulti` filtra le liste vuote e accumula le istanze man mano che si aggiungono injector lazy (core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93); `getLastValueSync` sottoscrive, prende l'ultima emissione sincrona e disiscrive subito (core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23).

I chiamanti storefront sono `ComponentWrapperDirective` e `PageSlotComponent` (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L139-L139, core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L158-L158). La feature SmartEdit registra `SmartEditComponentDecorator` e `SmartEditSlotDecorator` come multi-provider (feature-libs/smartedit/core/decorators/index.ts:L12-L23), e quando l'editor modifica qualcosa `SmartEditService.renderComponent` invoca `clearComponentState()` + `refreshPageById`/`refreshLatestPage` per uno slot, oppure `refreshComponent(componentId)` per un componente (feature-libs/smartedit/core/services/smart-edit.service.ts:L126-L149). Il dettaglio SmartEdit è nel capitolo 33 — Tracking, SmartEdit, Qualtrics.

---

4. Diagrammi

4.1 Caricamento di una pagina (FL1)

Ogni freccia corrisponde ai passi 3.4–3.10.

TAVOLA 15.1 diagramma a tutta pagina, in fondo al capitolo

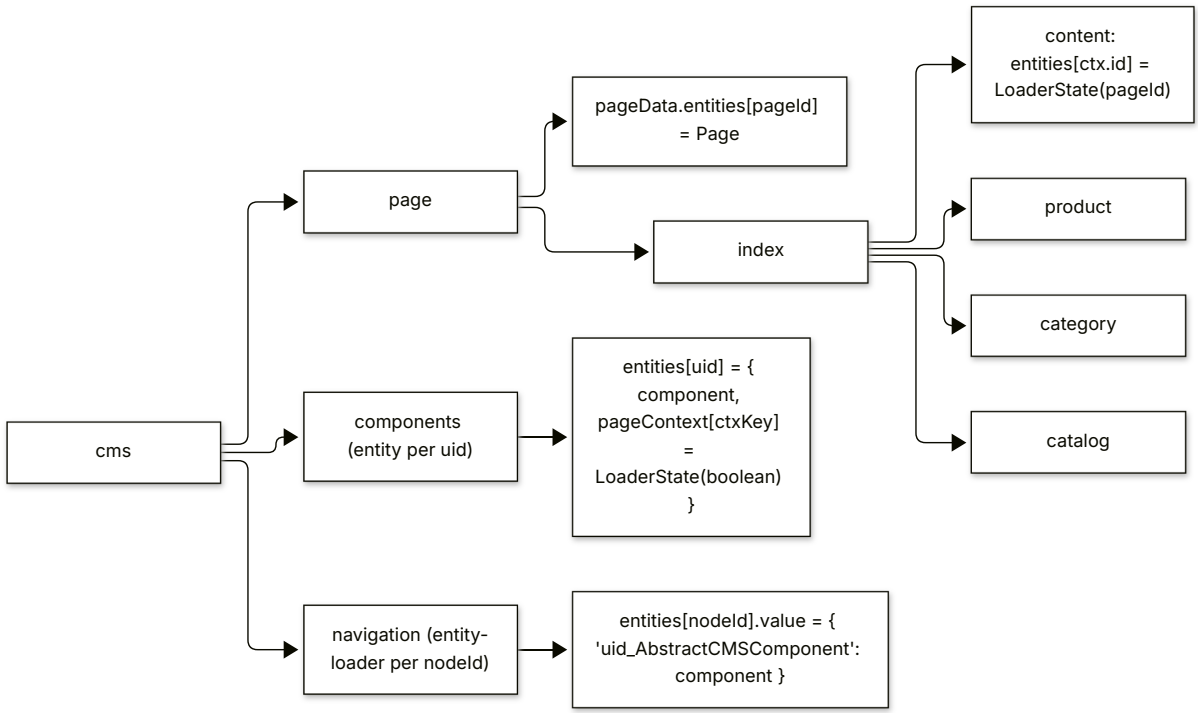
4.2 Dati di un componente e batching (FL2)

■ **TAVOLA 15.2** diagramma a tutta pagina, in fondo al capitolo

4.3 Come `createComponentData` decide se dispatchare

■ **TAVOLA 15.3** diagramma a tutta pagina, in fondo al capitolo

4.4 Forma dello store e chiavi



---

5. Estratti di codice reale

5.1 hasPage e getPage — `core-libs/core/src/cms/facade/cms.service.ts:L247-L283`

```ts
hasPage(pageContext: PageContext, forceReload = false): Observable<boolean> {
  return this.store.pipe(
    select(CmsSelectors.getPageStateIndexLoaderState(pageContext)),
    tap((entity: LoaderState<string>) => {
      const attemptedLoad = entity.loading || entity.success || entity.error;
      const shouldReload = forceReload && !entity.loading;
      if (!attemptedLoad || shouldReload) {
        this.store.dispatch(new CmsActions.LoadCmsPageData(pageContext));
        forceReload = false;
```

```
      }
    }),
    filter((entity) => {
      if (!entity.hasOwnProperty('value')) {
        // if we have incomplete state from SSR failed load transfer state,
        // we should wait for reload and actual value
        return false;
      }
      return Boolean(entity.success || (entity.error && !entity.loading));
    }),
    map((loaderState) => !!loaderState.success),
    catchError(() => of(false))
  );
}
// ...
getPage(
  pageContext: PageContext,
  forceReload = false
): Observable<Page | null> {
  return this.hasPage(pageContext, forceReload).pipe(
    switchMap((hasPage) =>
      hasPage ? this.getPageState(pageContext) : of(null)
    )
  );
}
```

5.2 createComponentData — core-libs/core/src/cms/facade/cms.service.ts:L110-L145

TS

```
  const context = serializePageContext(pageContext, true);

  const loading$ = combineLatest([
    this.routingService.getNextPageContext(),
    this.store.pipe(
      select(CmsSelectors.componentsLoaderStateSelectorFactory(uid, context))
    ),
  ]).pipe(
    observeOn(queueScheduler),
    tap(([nextContext, loadingState]) => {
      const attemptedLoad =
        loadingState.loading || loadingState.success || loadingState.error;
      // if the requested context is the same as the one that's currently being navigated to
      // (as it might already been triggered and might be available shortly from page data)
      // TODO(issue:3649), TODO(issue:3668) - this optimization could be removed
      const couldBeLoadedWithPageData = nextContext
        ? serializePageContext(nextContext, true) === context
        : false;

      if (!attemptedLoad && !couldBeLoadedWithPageData) {
```

```
          this.store.dispatch(
            new CmsActions.LoadCmsComponent({ uid, pageContext })
          );
        }
      })
    );

    const component$ = this.store.pipe(
      select(CmsSelectors.componentsSelectorFactory(uid, context)),
      filter(isNotUndefined)
    ) as Observable<T>;

    return using(
      () => loading$.subscribe(),
      () => component$
    ).pipe(shareReplay({ bufferSize: 1, refCount: true }));
```

5.3 PageEffects.loadPageData$ — core-libs/core/src/cms/store/effects/page.effect.ts:L58-L94

TS

```
  loadPageData$: Observable<Action> = createEffect(() =>
    this.actions$.pipe(
      ofType(CmsActions.LOAD_CMS_PAGE_DATA),
      map((action: CmsActions.LoadCmsPageData) => action.payload),
      groupBy((pageContext) => serializePageContext(pageContext)),
      mergeMap((group) =>
        group.pipe(
          switchMap((pageContext) =>
            this.cmsPageConnector.get(pageContext).pipe(
              mergeMap((cmsStructure: CmsStructureModel) => {
                const actions: Action[] = [
                  new CmsActions.CmsGetComponentFromPage(
                    (cmsStructure.components ?? []).map((component) => ({
                      component,
                      pageContext,
                    }))
                  ),
                  new CmsActions.LoadCmsPageDataSuccess(
                    pageContext,
                    cmsStructure.page ?? {}
                  ),
                ];

                const pageLabel = cmsStructure.page?.label;
                // For content pages the page label returned from backend can be different than page ID
initially assumed from route.
                // In such a case let's save the success response not only for initially assumed page
ID, but also for correct page label.
```

```
            if (pageLabel && pageLabel !== pageContext.id) {
              actions.unshift(
                new CmsActions.CmsSetPageSuccessIndex(
                  { id: pageLabel, type: pageContext.type },
                  cmsStructure.page ?? {}
                )
              );
            }

            return actions;
          }),
```

5.4 ComponentsEffects.loadComponent$ — core-libs/core/src/cms/store/effects/components.effect.ts:L41-L65

```ts
loadComponent$ = createEffect(
  () =>
    ({ scheduler, debounce = 0 } = {}): Observable<
      | CmsActions.LoadCmsComponentSuccess<CmsComponent>
      | CmsActions.LoadCmsComponentFail
    > =>
      this.actions$.pipe(
        ofType<CmsActions.LoadCmsComponent>(CmsActions.LOAD_CMS_COMPONENT),
        groupBy((actions) =>
          serializePageContext(actions.payload.pageContext)
        ),
        mergeMap((actionGroup) =>
          actionGroup.pipe(
            bufferDebounceTime(debounce, scheduler),
            mergeMap((actions) =>
              this.loadComponentsEffect(
                actions.map((action) => action.payload.uid),
                actions[0].payload.pageContext ?? { id: '' }
              )
            )
          )
        ),
        withdrawOn(this.contextChange$)
      )
);
```

5.5 `CmsPageConnector.get` — `core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L30-L54`

```ts
get(pageContext: PageContext): Observable<CmsStructureModel> {
  return this.cmsStructureConfigService
    .shouldIgnoreBackend(pageContext.id)
    .pipe(
      switchMap((loadFromConfig) => {
        if (!loadFromConfig) {
          return this.cmsPageAdapter.load(pageContext).pipe(
            catchError((error) => {
              if (
                error instanceof HttpErrorResponse &&
                error.status === 400
              ) {
                return of({});
              } else {
                throw error;
              }
            })
          );
        } else {
          return of({});
        }
      }),
      switchMap((page) => this.mergeDefaultPageStructure(pageContext, page))
    );
}
```

5.6 Paginazione in `getList` — `core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L55-L80`

```ts
      if (missingIds.length > 0) {
        const pageSize =
          this.config.componentsLoading?.pageSize || missingIds.length;
        const totalPages = Math.ceil(missingIds.length / pageSize);
        const cmsComponents: Observable<CmsComponent[]>[] = [];

        let currentPage = 0;
        while (currentPage < totalPages) {
          cmsComponents.push(
            this.cmsComponentAdapter.findComponentsByIds(
              missingIds.slice(
                currentPage * pageSize,
                (currentPage + 1) * pageSize
              ),
              pageContext
            )
```

```
          );
          currentPage++;
        }
        return zip(...cmsComponents).pipe(
          map((loadedComponents) =>
            [...configuredComponents.filter(Boolean)].concat(
              ...loadedComponents
            )
          )
        );
```

5.7 mergeSlots — core-libs/core/src/cms/services/cms-structure-config.service.ts:L131-L167

TS

```ts
protected mergeSlots(
  pageStructure: CmsStructureModel,
  slots?: CmsPageSlotsConfig
): Observable<CmsStructureModel> {
  // if no slots have been given, we use the global configured slots
  if (!slots && this.cmsDataConfig.cmsStructure?.slots) {
    slots = this.cmsDataConfig.cmsStructure.slots;
  }

  if (!slots) {
    return of(pageStructure);
  }

  for (const position of Object.keys(slots)) {
    if (
      pageStructure.page?.slots &&
      !Object.keys(pageStructure.page.slots).includes(position)
    ) {
      // the global slot isn't yet part of the page structure
      pageStructure.page.slots[position] = {};

      for (const component of this.getComponentsByPosition(slots, position)) {
        pageStructure.page.slots[position].components ??= [];
        pageStructure.page.slots[position].components.push({
          uid: component.uid,
          flexType: component.flexType,
          typeCode: component.typeCode,
        });

        pageStructure.components ??= [];
        pageStructure.components.push(component);
      }
    }
  }
```

```
    return of(pageStructure);
  }
```

5.8 Il reducer dei componenti — core-libs/core/src/cms/store/reducers/components.reducer.ts:L18-L31 e L56-L70

```ts
function componentExistsReducer<T extends CmsComponent>(
  state: boolean | undefined,
  action: CmsActions.CmsComponentAction<T>
): boolean | undefined {
  switch (action.type) {
    case CmsActions.LOAD_CMS_COMPONENT_FAIL:
      return false;

    case CmsActions.CMS_GET_COMPONENT_FROM_PAGE:
    case CmsActions.LOAD_CMS_COMPONENT_SUCCESS:
      return true;
  }
  return state;
}
// ...
    case CmsActions.LOAD_CMS_COMPONENT_SUCCESS: {
      const pageContextReducer = loaderReducer<boolean, any>(
        action.meta.entityType,
        componentExistsReducer
      );
      const context = serializePageContext(action.payload.pageContext, true);
      return {
        ...state,
        component: action.payload.component as T,
        pageContext: {
          ...state.pageContext,
          [context]: pageContextReducer(state.pageContext[context], action),
        },
      };
    }
```

5.9 flexType — core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L151-L158

```ts
  protected getFlexTypeFromComponent(component: Occ.Component | any): string {
    if (component.typeCode === CMS_FLEX_COMPONENT_TYPE) {
      return component.flexType;
    } else if (component.typeCode === JSP_INCLUDE_CMS_COMPONENT_TYPE) {
      return component.uid;
```

```
    }
    return component.typeCode;
  }
```

5.10 Meta-reducer di reset — core-libs/core/src/cms/store/reducers/index.ts:L75-L90

```ts
export function clearCmsState(
  reducer: ActionReducer<StateWithCms, Action>
): ActionReducer<StateWithCms, Action> {
  return function (state, action) {
    if (
      action.type === SiteContextActions.LANGUAGE_CHANGE ||
      action.type === AuthActions.LOGOUT ||
      action.type === AuthActions.LOGIN
    ) {
      state = undefined;
    }
    return reducer(state, action);
  };
}

export const metaReducers: MetaReducer<StateWithCms>[] = [clearCmsState];
```

6. Configurazione

`CmsConfig` è una classe astratta `@Injectable({providedIn: 'root', useExisting: Config})` che estende `OccConfig` e aggiunge `featureModules`, `cmsComponents`, `componentsLoading.pageSize`; viene fusa nell'interfaccia globale `Config` via module augmentation (core-libs/core/src/cms/config/cms-config.ts:L124-L138). Il trucco `useExisting: Config` è consentito perché `@Injectable` accetta un `InjectableProvider` che include `ExistingSansProvider` (angular:packages/core/src/di/injectable.ts:L29-L35, angular:packages/core/src/di/interface/provider.ts:L161-L166): iniettare `CmsConfig` equivale a iniettare `Config`. Il meccanismo generale è nel capitolo 11 — Configurazione.

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cmsComponents	CMSComponentConfig	{}	core-libs/core/src/cms/config/cms-config.ts:L130-L130; default core-libs/core/src/cms/config/default-cms-config.ts:L20-L20	core-libs/storefront/cms-structure/services/cms-components.service.ts:L67-L67 (cap. 16)

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cmsComponents[type]. component / providers / childRoutes / disableSSR / i18nKeys / guards / data / deferLoading	vedi CmsComponentMapping	—	core-libs/core/src/cms/config/cms-config.ts:L60-L93	storefront, cap. 16 (DA VERIFICARE riga per riga nel cap. 16: fuori perimetro di questo dossier)
featureModules	{[name]: FeatureModuleConfig | string}	nessuno	core-libs/core/src/cms/config/cms-config.ts:L129-L129	core-libs/storefront/cms-structure/services/cms-features.service.ts:L60-L60 (cap. 12)
featureModules[x]. cmsComponents	string[]	—	core-libs/core/src/cms/config/cms-config.ts:L121-L121	core-libs/storefront/cms-structure/services/cms-features.service.ts:L68-L70
componentsLoading. pageSize	number	50	core-libs/core/src/cms/config/cms-config.ts:L131-L133; default core-libs/core/src/cms/config/default-cms-config.ts:L21-L23	core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L56-L57
backend.occ.endpoints. component	string | OccEndpoint	'users/${userId}/cms/components/${id}'	core-libs/core/src/cms/config/default-cms-config.ts:L13-L13	core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts:L93-L93
backend.occ.endpoints. components	string | OccEndpoint	'users/${userId}/cms/components'	core-libs/core/src/cms/config/default-cms-config.ts:L14-L14	core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts:L110-L110
backend.occ.endpoints. pages	string | OccEndpoint	'users/${userId}/cms/pages'	core-libs/core/src/cms/config/default-cms-config.ts:L15-L15	core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L56-L59
backend.occ.endpoints. page	string | OccEndpoint	'users/${userId}/cms/pages/${id}'	core-libs/core/src/cms/config/default-cms-config.ts:L16-L16	core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L53-L55
cmsStructure.components	{[id]: ContentSlotComponentData | any}	nessuno (defaultCmsContentProviders, se l'app lo include: HamburgerMenu/Login)	core-libs/core/src/cms/config/cms-structure.config.ts:L80-L84	core-libs/core/src/cms/services/cms-structure-config.service.ts:L176-L178, core-libs/core/src/cms/services/cms-structure-config.service.ts:L191-L191
cmsStructure.pages	CmsPageConfig[]	nessuno	core-libs/core/src/cms/config/cms-structure.config.ts:L80-L84	core-libs/core/src/cms/services/cms-structure-config.service.ts:L87-L91

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cmsStructure.pages[].ignoreBackend	boolean	undefined (= false)	core-libs/core/src/cms/config/cms-structure.config.ts:L25-L25	core-libs/core/src/cms/services/cms-structure-config.service.ts:L56-L60
cmsStructure.slots	CmsPageSlotsConfig	nessuno (defaultCmsContentProviders, se l'app lo include: PreHeader, SiteLogin)	core-libs/core/src/cms/config/cms-structure.config.ts:L80-L84	core-libs/core/src/cms/services/cms-structure-config.service.ts:L136-L138
state.ssrTransfer.keys.cms	StateTransferType	TRANSFER_STATE	core-libs/core/src/cms/store/cms-store.module.ts:L21-L31	modulo state (cap. 19)
pageMeta.resolvers	PageMetaResolverConfig[]	7 voci	core-libs/core/src/cms/page/config/page-meta.config.ts:L22-L29; default core-libs/core/src/cms/page/config/default-page-meta.config.ts:L11-L40	core-libs/core/src/cms/facade/page-meta.service.ts:L115-L125 (cap. 40)
pageMeta.canonicalUrl	CanonicalUrlOptions	{forceHttps: true, forceWww: false, removeQueryParams: true, forceTrailingSlash: true}	core-libs/core/src/cms/page/config/page-meta.config.ts:L43-L71; default core-libs/core/src/cms/page/config/default-page-meta.config.ts:L41-L46	core-libs/core/src/cms/page/routing/page-link.service.ts:L34-L37 (cap. 40)

6.1 Il contratto `CmsComponentMapping`

`CmsComponentMapping<T>` definisce per ogni tipo CMS: `component`, `providers` (`StaticProvider[]`), `childRoutes`, `disableSSR`, `i18nKeys`, `guards`, `data` statica (tipizzata `T`) e `deferLoading` (core-libs/core/src/cms/config/cms-config.ts:L60-L93). Note dal sorgente:

- `providers`: i component service sono pensati come non-singleton, scoped all'albero di iniezione del componente; i provider configurabili permettono di sostituirli via config (core-libs/core/src/cms/config/cms-config.ts:L62-L73).
- `data`: dati iniziali senza chiamata API, proprietà UI non presenti in API, ghost design (core-libs/core/src/cms/config/cms-config.ts:L79-L85).
- `deferLoading`: `DeferLoadingStrategy.DEFER = 'DEFERRED-LOADING'` o `INSTANT = 'INSTANT-LOADING'` (core-libs/core/src/cms/config/cms-config.ts:L96-L101).
- `childRoutes`: un array di route oppure `CmsComponentChildRoutesConfig` con `parent.data` e `children` (core-libs/core/src/cms/config/cms-config.ts:L48-L58).

`CMSComponentConfig` unisce le chiavi tipizzate di `StandardCmsComponentConfig` (17 tipi standard) e `JspIncludeCmsComponentConfig` (6 tipi JSP) con un index signature `[componentType: string]` (core-libs/core/src/cms/config/cms-config.ts:L13-L40, core-libs/core/src/cms/config/cms-config.ts:L103-L107). `FeatureModuleConfig` ha `module` lazy, `dependencies` e `cmsComponents` (core-libs/core/src/cms/config/cms-config.ts:L109-L122). L'elenco completo dei mapping presenti nel repo è nell'Appendice A03.

6.2 Il contratto `CmsPageConfig`

`CmsPageConfig` ha `ignoreBackend`, `pageId`, `type`, `title`, `template` e `slots` obbligatorio (core-libs/core/src/cms/config/cms-structure.config.ts:L19-L49); `CmsPageSlotConfig` è `{ componentIds?: string[]; properties?: any }` e `CmsPageSlotsConfig` una mappa posizione → slot (core-libs/core/src/cms/config/cms-structure.config.ts:L54-L66).

---

7. Commenti e note notevoli dal sorgente

- **Ottimizzazione "rimovibile".** In `createComponentData`, il salto del dispatch quando il contesto coincide con quello in navigazione è accompagnato da `TODO(issue:3649), TODO(issue:3668) - this optimization could be removed` (core-libs/core/src/cms/facade/cms.service.ts:L122-L124).
- **SSR incompleto.** `hasPage` commenta: "if we have incomplete state from SSR failed load transfer state, we should wait for reload and actual value" (core-libs/core/src/cms/facade/cms.service.ts:L260-L261).
- **Label ≠ id.** "For content pages the page label returned from backend can be different than page ID initially assumed from route." (core-libs/core/src/cms/store/effects/page.effect.ts:L82-L83); lo stesso concetto riappare nella guard (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L87-L87).
- **Factory per la costante.** "if we want to reuse CMS_FEATURE const in config, we have to use factory instead of plain object" (core-libs/core/src/cms/store/cms-store.module.ts:L22-L22).
- **OCC fuori sync.** "while we're hoping to get this right from the backend api, the OCC api stills seems out of sync with the right model." — a proposito di `modifiedtime` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L176-L177).
- **Navigazione incompleta.** Gli item di navigazione di tipo pagina e media sono un "TODO: future work" commentato (core-libs/core/src/cms/store/effects/navigation-entry-item.effect.ts:L64-L69).
- **Ticket di tipizzazione.** `CXSPA-11104: Revise type to avoid redundant types` su `getComponentsFromConfig` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L74-L74).
- **Legacy adapter.** Il JSDoc spiega che il passaggio da POST a GET era inizialmente dietro un flag `legacy` e "from version 3.0 onwards" è stato spostato in un adapter opzionale (core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts:L16-L25).
- **Refresh su cambio lingua.** `PageMetaService` cita il ticket CXSPA-10923 e salta la prima emissione di `getActive()` per non duplicare il refresh all'avvio (core-libs/core/src/cms/facade/page-meta.service.ts:L42-L50); nello stesso file `platformId` è marcato `// @deprecated (will be removed in 12 months)` (core-libs/core/src/cms/facade/page-meta.service.ts:L39-L40).
- **Uso dichiarato di `clearComponentState`.** Il JSDoc cita esplicitamente SmartEdit: "for example in SmartEdit context when child components need to be refreshed" (core-libs/core/src/cms/facade/cms.service.ts:L295-L309).
- `CmsStructureConfigService` **e il suo intento.** "The intent of the `CmsStructureConfigService` however is to provide fast loading pages and default cms structure for commodity commerce." (core-libs/core/src/cms/services/cms-structure-config.service.ts:L18-L31).

8. Extension point

Tutto ciò che segue è esportato da `@spartacus/core` tramite core-libs/core/public_api.ts:L14-L14 (`export * from './src/cms/index';`), salvo dove indicato.

EXTENSION POINT	TIPO	COME SI USA	SORGENTE
`CmsPageAdapter`	classe astratta / token DI	sostituire per un CMS terzo (`{ provide: CmsPageAdapter, useClass: ... }`)	core-libs/core/src/cms/connectors/page/cms-page.adapter.ts:L15-L24
`CmsComponentAdapter`	classe astratta / token DI	idem per i componenti; alternativa pronta `LegacyOccCmsComponentAdapter` (export via core-libs/core/public_api.ts:L27-L27)	core-libs/core/src/cms/connectors/component/cms-component.adapter.ts:L11-L35, core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts:L29-L29
`CMS_PAGE_NORMALIZER`	`InjectionToken` multi	aggiungere un converter dopo `OccCmsPageNormalizer`, o estendere quest'ultimo (metodi `protected`)	core-libs/core/src/cms/connectors/page/converters.ts:L11-L13, core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L22-L35
`CMS_COMPONENT_NORMALIZER`	`InjectionToken` multi	nessun default: un converter registrato qui diventa l'unico passaggio di normalizzazione dei componenti	core-libs/core/src/cms/connectors/component/converters.ts:L11-L13
`CmsPageConnector`, `CmsComponentConnector`	`providedIn: 'root'`	override via DI	core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L16-L19, core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L16-L19
`CmsService`	`providedIn: 'root'`, campi `protected`	override via DI (facade)	core-libs/core/src/cms/facade/cms.service.ts:L33-L46
`CmsStructureConfigService`	`providedIn: 'root'`, metodi `protected`	override di `mergePage`/`mergeSlots`/…	core-libs/core/src/cms/services/cms-structure-config.service.ts:L27-L31
`CmsConfig`, `CmsStructureConfig`	config token alias di `Config`	`provideConfig` / `provideDefaultConfig`	core-libs/core/src/cms/config/cms-config.ts:L124-L138, core-libs/core/src/cms/config/cms-structure.config.ts:L75-L89
`provideCmsStructure()`	helper (`@spartacus/storefront`)	componente statico in uno slot globale	core-libs/storefront/cms-structure/utils/cms-structure.util.ts:L31-L38
`ComponentDecorator`, `SlotDecorator`	multi-provider	aggiungere attributi DOM a componenti/slot	core-libs/core/src/cms/decorators/component-decorator.ts:L10-L23, core-libs/core/src/cms/decorators/slot-decorator.ts:L10-L23

EXTENSION POINT	TIPO	COME SI USA	SORGENTE
`CmsActions`, `CmsSelectors`, `StateWithCms`, `CMS_FEATURE`	API NgRx pubblica	dispatch e selettori componibili; effects e reducers NON esportati	`core-libs/core/src/cms/index.ts:L19-L21`
`CmsComponentMapping.providers/guards/childRoutes/data/deferLoading`	config per componente	personalizzazione per tipo CMS	`core-libs/core/src/cms/config/cms-config.ts:L60-L93`
`PageMetaResolver` e `pageMeta.resolvers`	multi-provider + config	meta per tipo di pagina (cap. 40)	`core-libs/core/src/cms/page/page-meta.resolver.ts:L17-L54`

Ricette (Parte III)

- Sostituire o aggiungere un componente CMS tramite `cmsComponents`, page layout e struttura statica: `../parte-3/51-ricette-cms-outlet-layout.md` (cap. 51).
- Adapter e converter custom (es. `CmsPageAdapter` verso un CMS terzo, normalizer aggiuntivo su `CMS_PAGE_NORMALIZER`), endpoint OCC e meta resolver: `../parte-3/52-ricette-occ-dati.md` (cap. 52).
- Override di service/facade via DI (es. `CmsService`, `CmsStructureConfigService`) e di configurazione: `../parte-3/50-ricette-config-di.md` (cap. 50).
- Feature lazy con `featureModules[x].cmsComponents`: `../parte-3/53-ricette-feature-cart-checkout-stili-i18n.md` (cap. 53).
- Cosa è API pubblica (es. `serializePageContext` non lo è) e cosa si rompe negli upgrade: `../parte-3/54-upgrade-safe.md` (cap. 54).

Schizzo illustrativo (non presente nel repo) di un adapter di pagina per un CMS terzo, che usa solo simboli esportati da `@spartacus/core`:

```ts
// ESEMPIO ILLUSTRATIVO — non è codice del repository
import { Injectable } from '@angular/core';
import { Observable, of } from 'rxjs';
import { CmsPageAdapter, CmsStructureModel, PageContext } from '@spartacus/core';

@Injectable()
export class MyHeadlessCmsPageAdapter extends CmsPageAdapter {
  load(pageContext: PageContext): Observable<CmsStructureModel> {
    // costruire { page: { pageId, template, slots }, components: [...] }
    return of({ page: { pageId: pageContext.id, slots: {} }, components: [] });
  }
}

// providers: [{ provide: CmsPageAdapter, useClass: MyHeadlessCmsPageAdapter }]
```

Attenzione: `CmsOccModule` registra `CmsPageAdapter` con `useExisting: OccCmsPageAdapter` (`core-libs/core/src/occ/adapters/cms/cms-occ.module.ts:L16-L34`); quale provider vinca dipende dall'ordine degli injector/moduli (regole DI nel capitolo 01 — Dependency Injection in Angular e in Spartacus). Il `pageId` restituito deve essere coerente, perché diventa la chiave di `pageData.entities` e il valore dell'indice (3.10).

9. Trappole, casi limite, comportamenti controintuitivi

1. **Due serializzazioni diverse del contesto.** La cache di `CmsService` e il components reducer usano `ignoreContentPageId = true` (tutte le ContentPage condividono la chiave `'ContentPage'`), mentre i `groupBy` negli effects usano la forma completa `type-id` (core-libs/core/src/cms/facade/cms.service.ts:L82-L82, core-libs/core/src/cms/store/reducers/components.reducer.ts:L47-L47, core-libs/core/src/cms/store/effects/components.effect.ts:L49-L51, core-libs/core/src/cms/store/effects/page.effect.ts:L62-L62). Due content page diverse producono due gruppi nell'effect ma un'unica entry di stato.
2. **Un solo `component` per uid.** Lo stato componente ha un unico `component` condiviso tra contesti: un `LoadCmsComponentSuccess` per un contesto sovrascrive i dati visti da tutti i contesti in cui il componente "esiste" (core-libs/core/src/cms/store/cms-state.ts:L22-L23, core-libs/core/src/cms/store/reducers/components.reducer.ts:L62-L65, core-libs/core/src/cms/store/selectors/components.selectors.ts:L83-L86). Se il backend restituisce varianti diverse dello stesso uid per prodotti diversi, l'ultima vince.
3. `getComponentData` **può emettere** `null`**.** Il selettore tri-stato restituisce `null` per componente inesistente o ristretto, e il service filtra solo `undefined` (core-libs/core/src/cms/facade/cms.service.ts:L137-L140, core-libs/core/src/cms/store/selectors/components.selectors.ts:L87-L88). I consumatori devono gestirlo.
4. `refreshPageById` **senza tipo.** Crea un contesto senza `type`: entity type `''`, quindi nessun ramo dell'indice viene aggiornato (il page-index reducer filtra per tipo), mentre `pageData.entities[pageId]` sì; l'adapter OCC usa l'endpoint `page/{id}` (core-libs/core/src/cms/facade/cms.service.ts:L210-L213, core-libs/core/src/cms/store/reducers/page-index.reducer.ts:L28-L28, core-libs/core/src/cms/store/reducers/page-data.reducer.ts:L18-L23, core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L52-L55). La pagina corrente vede comunque i dati aggiornati se il suo indice punta già a quel `pageId`, perché `getPageData` risolve indice → `pageData.entities` (core-libs/core/src/cms/store/selectors/page.selectors.ts:L88-L96); un contesto che non è mai stato indicizzato, invece, non ne beneficia.
5. `refreshComponent(uid)` **senza contesto.** Il `LoadCmsComponent` scrive `loading: true` sotto la chiave `'current'`, ma l'effect sostituisce il contesto con `{ id: '' }`. Success e Fail vengono quindi scritti sotto la chiave `'undefined-'` (tipo `undefined`), e la entry `'current'` resta in loading. La richiesta OCC parte senza parametri di contesto (core-libs/core/src/cms/facade/cms.service.ts:L221-L223, core-libs/core/src/cms/utils/cms-utils.ts:L25-L33, core-libs/core/src/cms/store/effects/components.effect.ts:L58-L58, core-libs/core/src/cms/store/effects/components.effect.ts:L83-L87, core-libs/core/src/cms/store/reducers/components.reducer.ts:L61-L61).
6. **Un 400 diventa "pagina vuota" e poi 404.** `CmsPageConnector` converte un 400 in `{}` (core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L37-L46). Senza pagina configurata `mergeSlots` non aggiunge nulla perché richiede `pageStructure.page?.slots` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L145-L148); l'effect emette `LoadCmsPageDataSuccess(ctx, {})` (core-libs/core/src/cms/store/effects/page.effect.ts:L75-L78), che salva la pagina sotto `entities['']` (core-libs/core/src/cms/store/reducers/page-data.reducer.ts:L22-L22) e mette nell'indice `success: true` con `value: undefined` (core-libs/core/src/state/utils/loader/loader.reducer.ts:L53-L60). `hasPage` emette quindi `true`, ma `getPageData` restituisce `entities[undefined]`, cioè `undefined` (core-libs/core/src/cms/store/selectors/page.selectors.ts:L88-L96), e `CmsPageGuard` sceglie `canActivateNotFoundPage` (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L62-L77). Risultato finale corretto (404), ma per una strada indiretta: lo stato dice "successo".
7. **Il CMS vince sulla config statica, anche con slot vuoti.** Uno slot già presente nella pagina CMS, anche senza componenti, blocca l'iniezione dello slot statico omonimo (core-libs/core/src/cms/services/cms-

structure-config.service.ts:L124-L148). Se `provideCmsStructure` "non funziona", controllare che il template CMS non definisca già quella posizione.

8. **Mutazioni in place.** `mergePage`/`mergeSlots` mutano l'oggetto `pageStructure` ricevuto (core-libs/core/src/cms/services/cms-structure-config.service.ts:L107-L115, core-libs/core/src/cms/services/cms-structure-config.service.ts:L150-L161); `OccCmsPageNormalizer` muta la sorgente OCC (wrap di `contentSlot`, rename `modifiedtime`, `properties = undefined`). L'ordine dei passi conta: `normalizePageComponentData` legge `properties` prima che `normalizeComponentData` le azzeri (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L30-L33, core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L178-L186). Un converter custom aggiunto *dopo* su `CMS_PAGE_NORMALIZER` riceve quindi una sorgente già modificata.
9. **Componente singolo non array.** `normalizePageComponentData` salta un `components.component` non array, ma `normalizeComponentData` lo itera con `for...of`: un singolo oggetto lancia un `TypeError`, che non è un `HttpErrorResponse` 400, quindi `CmsPageConnector` lo rilancia e fa fallire il caricamento della pagina (`LoadCmsPageDataFail`). Un `contentSlot` singolo, invece, viene normalizzato ad array (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L89-L94, core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L118-L118, core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L175-L175, core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L37-L46, core-libs/core/src/cms/store/effects/page.effect.ts:L95-L102).
10. `pageSize` **falsy.** `componentsLoading.pageSize = 0` non disattiva il caricamento: fa fallback a tutti gli id in un'unica richiesta (core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L56-L57).
11. **Ordine dei risultati di** `getList`**.** Non rispetta l'ordine degli id: prima i componenti da config, poi quelli caricati (core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L74-L80).
12. **Il payload array è gestito "altrove".** Il components reducer ignora `CMS_GET_COMPONENT_FROM_PAGE` con payload array: funziona solo perché `entityReducer` partiziona prima il payload (core-libs/core/src/cms/store/reducers/components.reducer.ts:L76-L87, core-libs/core/src/state/utils/entity/entity.reducer.ts:L40-L53). Chi dispatcha a mano un'azione con payload array e `entityId` non array non otterrà nulla.
13. `CmsSetPageSuccessIndex` **fuori dall'unione.** Non è incluso nel tipo `CmsPageAction`, ma il page-index reducer lo gestisce (core-libs/core/src/cms/store/actions/page.action.ts:L72-L76, core-libs/core/src/cms/store/reducers/page-index.reducer.ts:L42-L44).
14. **Cache del service mai svuotata.** La mappa `components` di `CmsService` cresce per uid/contesto e si svuota solo con `clearComponentState` (core-libs/core/src/cms/facade/cms.service.ts:L37-L41, core-libs/core/src/cms/facade/cms.service.ts:L302-L309). Dopo `clearCmsState` (lingua/login) l'observable memoizzato continua a selezionare dallo store azzerato: il loader state torna iniziale e un nuovo dispatch avviene alla successiva emissione (core-libs/core/src/cms/store/reducers/index.ts:L75-L88).
15. **Slot assente ≠ slot inesistente.** `getContentSlot` emette `{ components: [] }` per una posizione assente su una pagina esistente, non `undefined` (core-libs/core/src/cms/store/selectors/page.selectors.ts:L109-L113).
16. **Decorator lazy non applicati.** `DynamicAttributeService` usa `getLastValueSync`: decorator forniti in moduli lazy non ancora caricati non vengono applicati; senza decorator `getMulti` non emette e si usa `|| []` (core-libs/core/src/cms/services/dynamic-attribute.service.ts:L45-L47, core-libs/core/src/lazy-loading/unified-injector.ts:L83-L91).
17. **Classe** `abstract` **istanziata.** `CmsStructureConfigService` è dichiarata `abstract` ma ha `providedIn: 'root'`: l'`abstract` di TypeScript non impedisce ad Angular di creare la factory, e nel repo non ci sono sottoclassi (core-libs/core/src/cms/services/cms-structure-config.service.ts:L27-L31).
18. `forceReload` **mutato.** `hasPage` modifica il proprio parametro dentro `tap` per evitare reload ripetuti a ogni emissione (core-libs/core/src/cms/facade/cms.service.ts:L252-L256).

19. `serializePageContext` **non è pubblico.** Non compare nel barrel `core-libs/core/src/cms/index.ts` (core-libs/core/src/cms/index.ts:L7-L21): import da path interni non sono upgrade-safe.

---

10. Come fare debug

- **Redux DevTools.** Filtrare le azioni per prefisso `[Cms]`: i tipi pagina sono in core-libs/core/src/cms/store/actions/page.action.ts:L12-L16, quelli dei componenti in core-libs/core/src/cms/store/actions/components.action.ts:L13-L17. Una navigazione sana mostra `[Cms] Load Page Data` → (opz.) `[Cms] Set Page Success Index` → `[Cms] Get Component from Page` → `[Cms] Load Page Data Success`, secondo l'ordine dell'effect (core-libs/core/src/cms/store/effects/page.effect.ts:L67-L93).
- **Leggere lo stato.** In `cms.page.index.<tipo>.entities[<id>]` controllare `loading/success/error/value`: `value` è il `pageId`; poi `cms.page.pageData.entities[<pageId>]` (core-libs/core/src/cms/store/cms-state.ts:L64-L84). Per una 404 attesa, la entry dell'URL avrà `error: true` e `value` uguale al `pageId` della pagina not-found (3.11).
- **Componente che non appare.** In `cms.components.entities[<uid>].pageContext` cercare la chiave del contesto (`'ContentPage'`, `'ProductPage-<code>'`, `'current'`…, vedi core-libs/core/src/cms/utils/cms-utils.ts:L21-L34): `value: false` significa che il backend non l'ha restituito, e il messaggio dell'errore è `Failed to load CmsComponent <type> uid: <uid>` (core-libs/core/src/cms/store/effects/components.effect.ts:L93-L103).
- **Verificare il batching.** Nella tab Network una navigazione dovrebbe produrre una GET su `.../cms/components?componentIds=a,b,c&fields=DEFAULT&...` per contesto (core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts:L51-L78), spezzata in blocchi da `pageSize` (core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L55-L73). Molte richieste singole indicano consumatori che passano contesti diversi (es. `undefined` → `'current'`).
- **Pagina statica che non si applica.** Verificare `cmsStructure.pages[].pageId` contro `pageContext.id` (il match è esatto, core-libs/core/src/cms/services/cms-structure-config.service.ts:L84-L92) e che lo slot non esista già nel template CMS (trappola 7).
- **Errori di pagina.** `LoadCmsPageDataFail` porta l'errore normalizzato da `tryNormalizeHttpError` (core-libs/core/src/cms/store/effects/page.effect.ts:L95-L102); ricordare che un 400 **non** arriva qui (trappola 6).
- **SSR.** Se la pagina lato client resta in attesa dopo un SSR fallito, guardare lo stato trasferito: entry dell'indice senza la proprietà `value` vengono scartate da `hasPage` (core-libs/core/src/cms/facade/cms.service.ts:L258-L263). Il meccanismo di transfer state è nel capitolo 07 — SSR, hydration e TransferState.
- **Breakpoint utili.** `CmsService.hasPage` (decisione di dispatch), `CmsPageConnector.get` (backend vs config, 400), `OccCmsPageNormalizer.convert` (forma della risposta), `ComponentsEffects.loadComponentsEffect` (uid richiesti vs ricevuti), `CmsStructureConfigService.mergeSlots`.

---

11. Domande di autoverifica

1. **Perché due ContentPage diverse condividono lo stesso stato di esistenza di un componente, mentre due PDP no?** Perché cache e components reducer serializzano il contesto con `ignoreContentPageId = true`, producendo `'ContentPage'` per ogni content page e `'ProductPage-<code>'` per i prodotti (core-libs/core/src/cms/utils/cms-utils.ts:L29-L33, core-libs/core/src/cms/store/reducers/components.reducer.ts:L47-L47); coerentemente, l'adapter OCC non invia parametri di contesto per le ContentPage (core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts:L132-L152).

2. **In che ordine** `PageEffects` **emette le azioni di successo e perché** `CmsGetComponentFromPage` **precede** `LoadCmsPageDataSuccess`**?** Ordine: (opz.) `CmsSetPageSuccessIndex`, `CmsGetComponentFromPage`, `LoadCmsPageDataSuccess` (core-libs/core/src/cms/store/effects/page.effect.ts:L67-L91). Così, quando `hasPage` vede il successo e i componenti iniziano a chiedere dati, lo stato dei componenti inclusi nella pagina è già marcato come esistente e `createComponentData` non dispatcha (core-libs/core/src/cms/facade/cms.service.ts:L119-L133).
3. **Quante richieste HTTP produce la richiesta di 120 componenti assenti dalla config nello stesso turno, con la config di default?** Tre: `pageSize` di default è 50 (core-libs/core/src/cms/config/default-cms-config.ts:L21-L23) e `getList` divide gli id mancanti in `Math.ceil(120 / 50) = 3` pagine combinate con `zip` (core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L55-L80). L'effect gira con `debounce = 0`, perché NgRx lo invoca senza argomenti (sezione 3.13, deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L153-L155).
4. **Cosa succede nello store quando l'utente fa login?** Il meta-reducer `clearCmsState` azzera tutto lo stato `cms` (core-libs/core/src/cms/store/reducers/index.ts:L75-L90); `ComponentsEffects` resetta i buffer con `withdrawOn` (core-libs/core/src/cms/store/effects/components.effect.ts:L33-L39, core-libs/core/src/cms/store/effects/components.effect.ts:L63-L63); `PageEffects.refreshPage$` ricarica la pagina corrente quando il router è stabile su una route `cmsRequired` (core-libs/core/src/cms/store/effects/page.effect.ts:L34-L56).
5. **Ho configurato con** `provideCmsStructure` **un componente nello slot** `SiteLogin`**, ma non compare. Prima ipotesi?** Il template CMS definisce già la posizione `SiteLogin` (anche vuota): `mergeSlots` aggiunge uno slot statico solo se la posizione non è già presente (core-libs/core/src/cms/services/cms-structure-config.service.ts:L144-L163).
6. **Che chiave usa lo storefront per trovare il mapping di un componente** `CMSFlexComponent`**? E di un** `JspIncludeComponent`**?** Il `flexType` calcolato dal normalizer: `component.flexType` per `CMSFlexComponent`, `component.uid` per `JspIncludeComponent` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L151-L158); `getPageComponentTypes` raccoglie proprio questi valori (core-libs/core/src/cms/store/selectors/page.selectors.ts:L40-L52).
7. **Come fa un URL inesistente a mostrare la pagina 404 senza duplicarne i dati?** La guard carica la pagina `notFound`, ne legge l'indice e con `setPageFailIndex` assegna quel `pageId` alla entry dell'URL richiesto nel ramo indice del suo tipo (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L107-L143, core-libs/core/src/cms/store/reducers/page-index.reducer.ts:L38-L40); `getPageData` risolve poi indice → `pageData.entities` (core-libs/core/src/cms/store/selectors/page.selectors.ts:L88-L96).
8. **Un converter registrato su** `CMS_COMPONENT_NORMALIZER` **affianca un normalizer di default?** No: nel repo non c'è alcun provider per quel token e `pipeableMany` lascia passare lo stream invariato se non ci sono converter (core-libs/core/src/util/converter.service.ts:L93-L101). Il converter custom sarebbe l'unico; con più converter, `convertSource` li applica in sequenza con `reduce` partendo da `undefined` (core-libs/core/src/util/converter.service.ts:L130-L137).
9. **Perché** `hasPage` **scarta certi stati anche se** `success` **è vero?** Scarta gli stati senza la proprietà `value`, che sono quelli incompleti arrivati dal transfer state di un SSR fallito; attende il reload (core-libs/core/src/cms/facade/cms.service.ts:L258-L263).
10. **Qual è la differenza tra** `refreshComponent(uid)` **e** `getComponentData(uid)` **rispetto alla cache del service?** `refreshComponent` dispatcha direttamente `LoadCmsComponent` senza toccare la cache `components` (core-libs/core/src/cms/facade/cms.service.ts:L221-L223); `getComponentData` passa dalla cache e dispatcha solo se il caricamento non è mai stato tentato (core-libs/core/src/cms/facade/cms.service.ts:L74-L95, core-libs/core/src/cms/facade/cms.service.ts:L129-L133).

TAVOLA 15.1 **4.1 Caricamento di una pagina (FL1)**

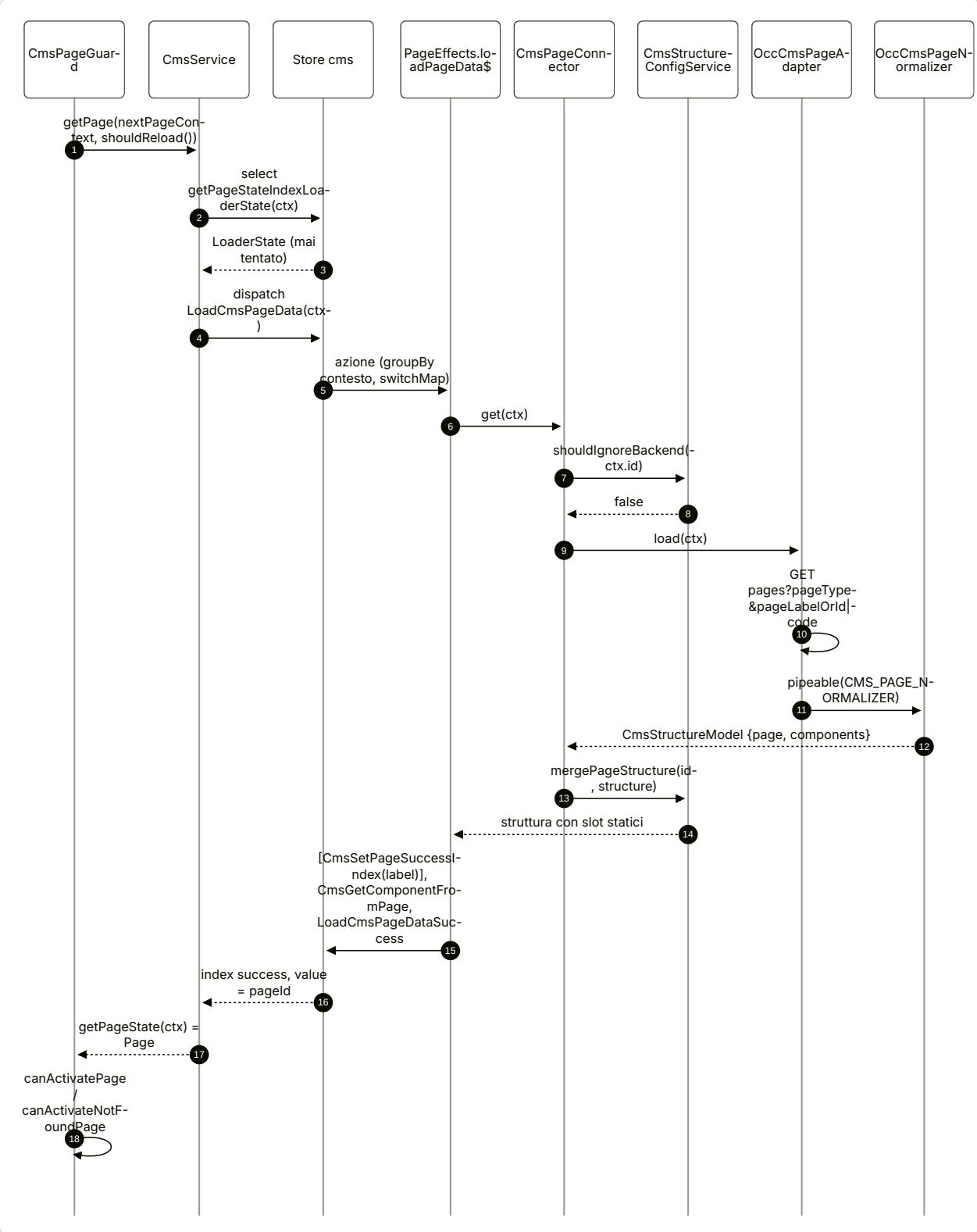


TAVOLA 15.2 **4.2 Dati di un componente e batching (FL2)**

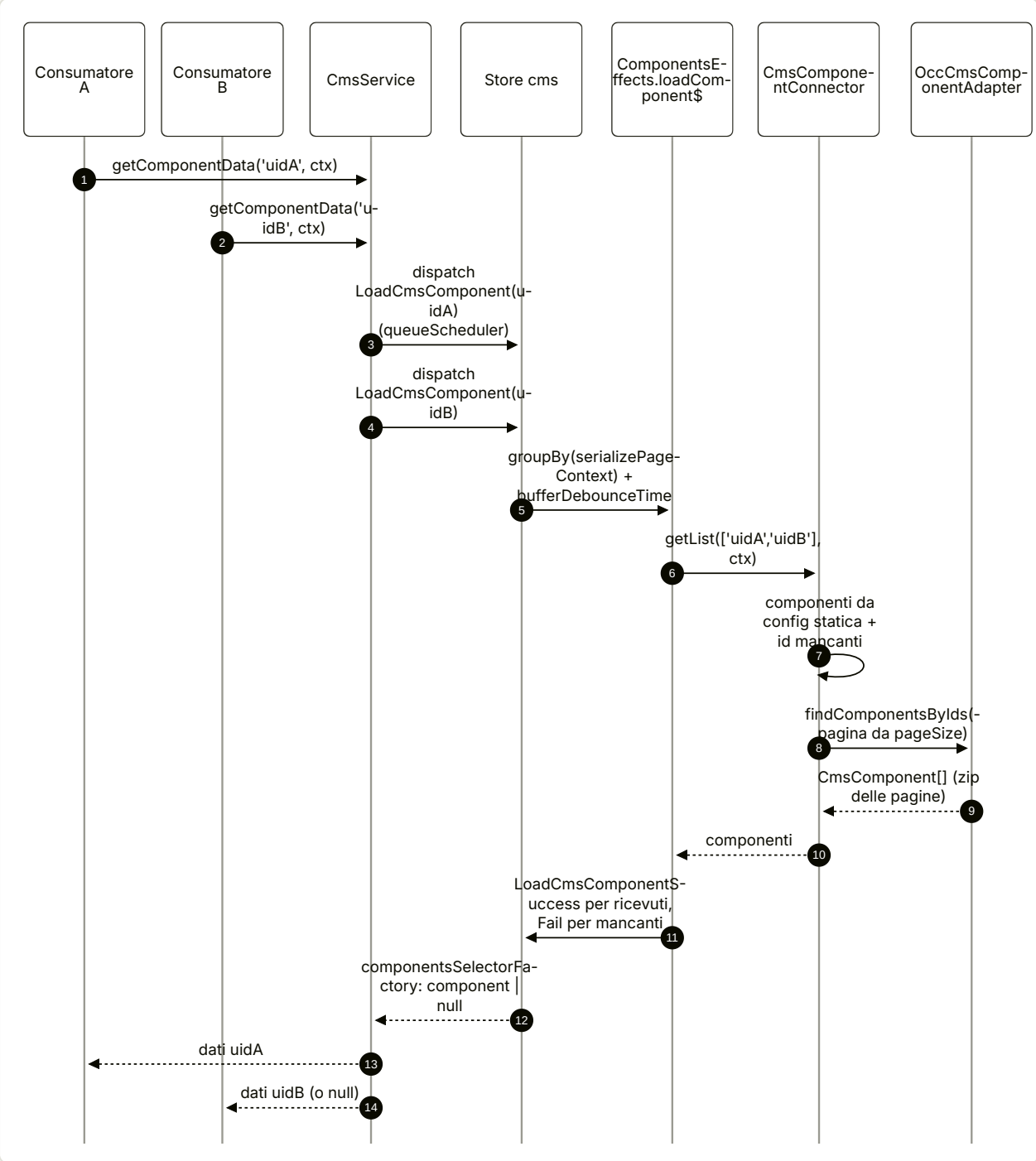
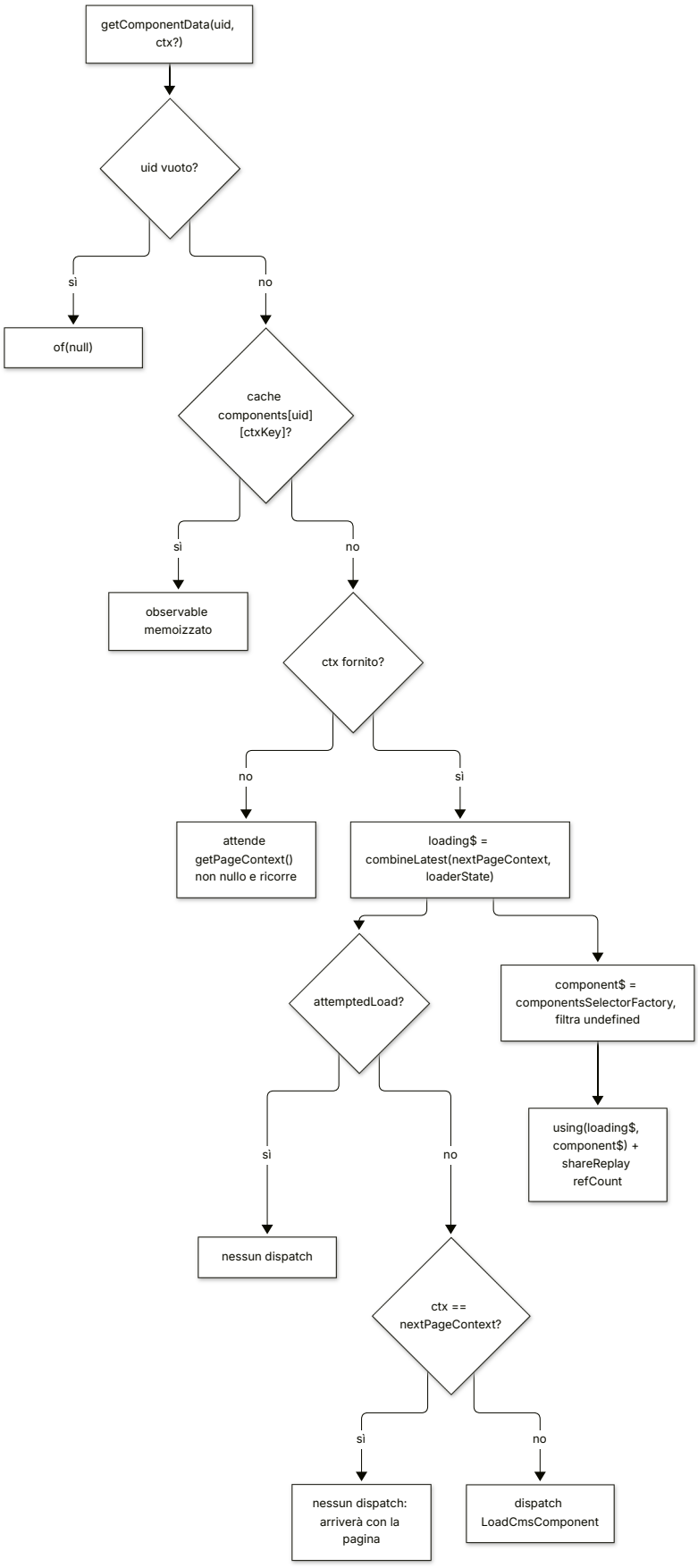


TAVOLA 15.3 **4.3 Come createComponentData decide se dispatchare**



16

CMS (storefront): rendering dinamico delle pagine

Layout degli slot, componenti differiti, mapping `cmsComponents`, injector per componente, guard CMS.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Questo capitolo segue il percorso di una pagina CMS dal momento in cui il router attiva una route Spartacus fino a quando un componente Angular (o un web component) viene istanziato in uno slot. Il capitolo 15 descrive come i dati della pagina arrivano nello store (`CmsService`, normalizer, connector) e come si definisce il mapping `cmsComponents`. Qui guardiamo il lato UI di `@spartacus/storefront`: chi legge quei dati e come li trasforma in DOM.

1. A cosa serve e perché esiste

In Spartacus quasi nessuna pagina ha un template Angular scritto a mano. Il backend SAP Commerce restituisce una struttura (page template, slot, componenti con `typeCode`/`flexType`) e lo storefront deve:

1. **scegliere quali slot mostrare e in che ordine**, in base a un `LayoutConfig` che dipende dal page template, dalla sezione (header/footer/...) e dal breakpoint (`PageLayoutService`, `core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L54-L80`);
2. **per ogni slot, rendere i componenti**, eventualmente in modo differito (lazy rendering sul viewport) e con classi CSS di stato (`PageSlotComponent`, `core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L48-L54`);
3. **per ogni componente, trovare l'implementazione** dal mapping `cmsComponents` (anche dentro feature module caricati in lazy), creare un injector dedicato con i dati CMS (`CmsComponentData`) e lanciare il componente con l'handler giusto: classe Angular, `() => import(...)` o web component (`ComponentWrapperDirective`, `core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L67-L119`);
4. **prima ancora di rendere**, durante la navigazione, risolvere i mapping, eseguire le guard dichiarate dai componenti, caricare i chunk i18n e, per le content page, generare al volo child route (`CmsPageGuardService.canActivatePage`, `core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L98`).

Il risultato: lo storefront è un *interprete* della struttura CMS. Aggiungere un componente a una pagina in Backoffice/SmartEdit non richiede nuovo codice Angular, purché esista un mapping per il suo `flexType`.

Il capitolo copre anche due argomenti collaterali assegnati agli stessi file: le route PDP/PLP (`cms-pages`) e il contesto LCP (Largest Contentful Paint), che sfrutta proprio l'injector per componente creato dal wrapper.

2. Mappa dei file

Il barrel pubblico è `core-libs/storefront/public_api.ts`, che riesporta `./cms-pages/index` (L22), `./cms-structure/index` (L23) e `./shared/index` (L29).

PATH	RUOLO
`core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts` (+ `.html`)	`PageLayoutComponent` (`cx-page-layout`): calcola nome layout, slot, page fold; rende un outlet e un `cx-page-slot` per slot
`core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts`	`PageLayoutService`: risolve gli slot da `LayoutConfig` (template/section/breakpoint) e li passa agli handler

PATH	RUOLO
core-libs/storefront/cms-structure/page/page-layout/page-layout-handler.ts	`PAGE_LAYOUT_HANDLER` (multi token) e interfaccia `PageLayoutHandler`
core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts	`PageTemplateDirective` (`cxPageTemplateStyle`): classe CSS = page template, con rehydration SSR
core-libs/storefront/cms-structure/page/page-layout/page-layout.module.ts	`PageLayoutModule`
core-libs/storefront/cms-structure/page/slot/page-slot.component.ts (+ .html)	`PageSlotComponent` (`cx-page-slot`): legge lo slot, gestisce classi `cx-pending`/`page-fold`, outlet per slot e per componente
core-libs/storefront/cms-structure/page/slot/page-slot.service.ts	`PageSlotService`: slot pre-renderizzati in SSR e opzioni di defer
core-libs/storefront/cms-structure/page/slot/page-slot.module.ts	`PageSlotModule` (istanzia subito `PageSlotService`)
core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts	`ComponentWrapperDirective` (`cxComponentWrapper`): mapping → injector → handler → eventi
core-libs/storefront/cms-structure/page/component/inner-components-host.directive.ts	`InnerComponentsHostDirective` (`cxInnerComponentsHost`): rende `composition.inner`
core-libs/storefront/cms-structure/page/component/handlers/component-handler.ts	`ComponentHandler`: classe astratta e token multi
core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts	`DefaultComponentHandler`: classi Angular (priorità `FALLBACK`)
core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts	`LazyComponentHandler`: `() => import()` con `PendingTasks` (priorità `LOW`)
core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts	`WebComponentHandler`: `'path#selector'` (non registrato né esportato)
core-libs/storefront/cms-structure/page/component/page-component.module.ts	`PageComponentModule` (+ `forRoot()` che registra gli handler)
core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts	`ComponentHandlerService`: sceglie l'handler con `resolveApplicable`
core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts	`CmsInjectorService`: injector per componente CMS
core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts	`ComponentDataProvider`: unisce dati statici e dati CMS (interno)
core-libs/storefront/cms-structure/page/component/services/cx-api.service.ts	`CxApiService`: facade esposti ai web component (interno)
core-libs/storefront/cms-structure/page/component/events/component.event.ts	`ComponentEvent`, `ComponentCreateEvent`, `ComponentDestroyEvent`
core-libs/storefront/cms-structure/page/model/cms-component-data.ts	`CmsComponentData<T>`
core-libs/storefront/cms-structure/services/cms-components.service.ts	`CmsComponentsService`: mapping (anche da feature lazy), SSR, child routes, guard, i18n, defer, dati statici
core-libs/storefront/cms-structure/services/cms-features.service.ts	`CmsFeaturesService`: feature module CMS e loro config
core-libs/storefront/cms-structure/services/cms-guards.service.ts	`CmsGuardsService`: guard dichiarate dai componenti della pagina

PATH	RUOLO
core-libs/storefront/cms-structure/services/guards-composer.ts	GuardsComposer e tipi CanActivate, CanActivateObservable
core-libs/storefront/cms-structure/services/cms-i18n.service.ts	CmsI18nService: chunk i18n dei componenti
core-libs/storefront/cms-structure/services/cms-routes.service.ts	CmsRoutesService (astratto pubblico, useExisting impl)
core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts	CmsRoutesImplService (interno): child route CMS con resetConfig
core-libs/storefront/cms-structure/services/utils.ts	isCanActivate (interno)
core-libs/storefront/cms-structure/utils/cms-structure.util.ts, core-libs/storefront/cms-structure/utils/cms-structure.model.ts	provideCmsStructure, PageSection, CmsStructureOptions
core-libs/storefront/cms-structure/cms-lcp-context/*	CmsLcpModule, CmsLcpService, LcpCmsComponentsConfig, provideLcpPresenceForCmsComponent
core-libs/storefront/shared/lcp-context/*	LCP_PRESENCE, LcpPresence, LcpContextDirective, ProvideLcpPresenceDirective, LcpPresenceMappingService
core-libs/storefront/cms-pages/product-details-page/*	ProductDetailsPageModule, PRODUCT_DETAILS_URL_MATCHER
core-libs/storefront/cms-pages/product-listing-page/*	ProductListingPageModule, PRODUCT_LISTING_URL_MATCHER

Collaboratori fuori dal perimetro del capitolo ma citati spesso: OutletDirective (cap. 17), BreakpointService e LayoutConfig (cap. 18), CmsPageGuard e routing (cap. 14), FeatureModulesService e UnifiedInjector (cap. 12).

3. Come funziona passo per passo

3.1 Registrazione

BaseStorefrontModule importa PageLayoutModule, PageComponentModule.forRoot(), PageSlotModule e CmsLcpModule.forRoot() (core-libs/storefront/base-storefront.module.ts:L26-L46). Di questi:

- PageComponentModule.forRoot() registra come multi-provider di ComponentHandler, tramite useExisting, DefaultComponentHandler e LazyComponentHandler (core-libs/storefront/cms-structure/page/component/page-component.module.ts:L23-L40). WebComponentHandler **non** è registrato (core-libs/storefront/cms-structure/page/component/handlers/index.ts:L7-L9; un grep sul repo, spec esclusi, trova la classe solo nella sua dichiarazione).
- PageSlotModule inietta PageSlotService nel costruttore per crearlo "ASAP", così che possa ispezionare il DOM pre-renderizzato dal server prima che Angular lo tocchi (core-libs/storefront/cms-structure/page/slot/page-slot.module.ts:L14-L23).
- CmsLcpModule.forRoot() fornisce la config di default lcpCmsComponents: { idMarker: '__cxLCP__', ids: [] } (core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.module.ts:L11-L19, core-libs/storefront/cms-structure/cms-lcp-context/config/default-lcp-cms-components.config.ts:L9-L14).

Le route con PageLayoutComponent arrivano da più parti; nel perimetro del capitolo ci sono i moduli PDP/PLP (§3.10) e la route generata da CmsRoutesImplService (§3.9).

3.2 Navigazione: `CmsPageGuardService.canActivatePage`

Prima che `PageLayoutComponent` venga creato, la guard CMS prepara il terreno. `canActivatePage` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L98):

1. ottiene i tipi di componente della pagina con `cmsService.getPageComponentTypes(pageContext)` e `take(1)` (L71-L72);
2. chiama `CmsComponentsService.determineMappings(componentTypes)`, che risolve anche i mapping che vivono in feature module lazy (L73-L75; dettagli in §3.7);
3. esegue le guard dichiarate dai componenti con `CmsGuardsService.cmsPageCanActivate` (L76-L80);
4. se il risultato è `true`, avvia il caricamento dei chunk i18n con `CmsI18nService.loadForComponents` (L81-L85);
5. se il risultato è `true` e la route **non** ha `data.cxCmsRouteContext`, delega a `CmsRoutesService.handleCmsRoutesInGuard`, che può generare child route e annullare la navigazione per rifarla (L86-L95; §3.9).

La costruzione di `pageContext`/`pageData` e la gestione della pagina not-found appartengono alla guard (cap. 14) e non sono trattate qui.

Guard dei componenti

`CmsGuardsService.cmsPageCanActivate` prende l'unione deduplicata delle guard dei mapping (`CmsComponentsService.getGuards`, che usa un `Set`: core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261), ottiene ogni istanza con `getLastValueSync(this.unifiedInjector.get(guardClass))` (quindi anche da injector di moduli lazy), tiene solo gli oggetti con metodo `canActivate` (`isCanActivate`, core-libs/storefront/cms-structure/services/utils.ts:L9-L11) e passa tutto a `GuardsComposer.canActivate` (core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48).

`GuardsComposer.canActivate` ritorna `of(true)` se non ci sono guard; altrimenti trasforma ogni risultato in Observable con `wrapIntoObservable(...).pipe(first())` (core-libs/storefront/cms-structure/services/guards-composer.ts:L39-L69) e li compone **in sequenza** con `concat`, saltando i `true` e fermandosi al primo `false`/`UrlTree`, oppure emettendo `true` alla fine (core-libs/storefront/cms-structure/services/guards-composer.ts:L78-L86). Il file definisce anche un proprio tipo `CanActivate = { canActivate: CanActivateFn }` come sostituto del tipo Angular deprecato (core-libs/storefront/cms-structure/services/guards-composer.ts:L16-L24).

i18n

`CmsI18nService.loadForComponents` raccoglie le `i18nKeys` dei componenti renderizzabili (`getI18nKeys` filtra con `shouldRender`: core-libs/storefront/cms-structure/services/cms-components.service.ts:L266-L276), le converte nei nomi di chunk con `TranslationChunkService.getChunkNameForKey`, deduplica e chiama `translation.loadChunks` (core-libs/storefront/cms-structure/services/cms-i18n.service.ts:L21-L28). Il caricamento è *fire-and-forget*: la guard non aspetta i chunk.

3.3 `PageLayoutComponent`: dal template agli slot

`PageLayoutComponent` è standalone, `OnPush`, selettore `cx-page-layout` (core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L17-L29). Tutto lo stato è reattivo:

- l'input `section` alimenta un `BehaviorSubject` `section$` che parte da `undefined` (L31-L34);
- `layoutName$` è la section se valorizzata, altrimenti il `templateName$` della pagina corrente (L36-L41);
- `slots$` è `pageLayoutService.getSlots(section)` per ogni valore di `section$` (L43-L45);

- `pageFoldSlot$` deriva **sempre** dal template (non dalla section) tramite `getPageFoldSlot` (L47-L53);
- il costruttore chiama `useFeatureStyles('a11yB2BRegisterComponent')` (L55-L57).

Il template (core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L1-L18) mette su un unico `ng-template` sia `[cxPageTemplateStyle]` sia `[cxOutlet]="layoutName"`, con contesto `{ templateName$, slots$, section$ }`. Dentro: `<ng-content />` e poi un `<cx-page-slot>` per ogni slot, con `isPageFold` vero solo per lo slot uguale a `pageFoldSlot$`. Il fatto che l'intero layout sia un outlet con nome = template (o section) permette di sostituire il layout di un'intera pagina con un outlet (cap. 17).

Lo stesso componente serve sia come componente di route (pagina intera, `section` assente) sia come contenitore di sezione: per esempio `<cx-page-layout section="header">` (la chiave `section` è una stringa libera; `PageSection` definisce `header`, `footer`, `navigation`: core-libs/storefront/cms-structure/utils/cms-structure.model.ts:L9-L13).

3.4 `PageLayoutService`: quali slot, in che ordine

`PageLayoutService` è `providedIn: 'root'` e nel costruttore si abbona a `unifiedInjector.getMulti(PAGE_LAYOUT_HANDLER)`, quindi raccoglie gli handler anche da moduli caricati in lazy (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L26-L46; `UnifiedInjector.getMulti` accumula con `scan` e scarta gli array vuoti: core-libs/core/src/lazy-loading/unified-injector.ts:L75-L94).

`getSlots(section?)` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L54-L80):

1. combina `page$` (`cms.getCurrentPage()` filtrato sui valori truthy, L129-L132) e `breakpointService.breakpoint$`;
2. calcola gli slot con `resolveSlots(page, section, breakpoint)`;
3. li fa passare **in catena** attraverso tutti gli handler: l'Observable restituito da uno è l'input del successivo (L61-L67);
4. deduplica con un confronto elemento per elemento (L68-L78).

`resolveSlots` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L106-L127) ha tre rami:

- **c'è config con** `slots`: tiene solo gli slot configurati che esistono davvero in `page.slots`, nell'ordine della configurazione (L117-L119);
- **nessuna config, nessuna section**: rende *tutti* gli slot della pagina nell'ordine delle chiavi di `page.slots` e logga un avviso (L120-L122);
- **nessuna config, con section**: rende `[]` e logga un avviso (L123-L125).

La ricerca della config passa da `getSlotConfig` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L146-L173): senza `layoutSlots` ritorna `undefined`; con section delega a `getSlotConfigForSection`; altrimenti risolve la config del template per breakpoint con `getResponsiveSlotConfig`.

`getSlotConfigForSection` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L175-L210) cerca prima `layoutSlots[template][section]`, poi la config globale `layoutSlots[section]`; ma se il template corrente non ha alcuna voce in `layoutSlots`, esce subito con `undefined` (L181-L185). Dopo la risoluzione responsive, se l'attributo manca prova `pageTemplateConfig[section]` e poi `layoutSlots[section]` (L201-L209).

`getResponsiveSlotConfig` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243) usa la config del breakpoint corrente se contiene l'attributo; altrimenti scorre i breakpoint **pre-**

cedenti nell'elenco ordinato di `BreakpointService.breakpoints` (ordinati per dimensione dello schermo, dal più piccolo al più grande: core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L54-L65; l'ordine crescente si vede in `isDown`, che considera "più piccoli o uguali" i breakpoint fino all'indice incluso, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L89-L97) e tiene l'ultimo che ha l'attributo; in mancanza, la config "piatta". Nota: il comparatore `sortByScreenSize` restituisce solo `-1` o `0`, mai `1` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L139-L152), quindi l'ordine crescente è garantito solo se le chiavi della config sono già in un ordine compatibile, come nel default.

`getPageFoldSlot(pageTemplate)` applica la stessa risoluzione responsive all'attributo `pageFold` di `layoutSlots[pageTemplate]` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L88-L104).

Gli avvisi passano da `logMissingLayoutConfig`: solo in `isDevMode()`, una volta per template con `logger.info` (elenco degli slot disponibili) e una volta per chiave (section o template) con `logger.warn` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L251-L273).

Esempio reale di handler: `CartPageLayoutHandler` agisce solo su `CartPageTemplate` senza section ed esclude slot in base allo stato del carrello (feature-libs/cart/base/components/cart-page-layout-handler.ts:L28-L60); è registrato come multi-provider in `CartBaseComponentsModule` (feature-libs/cart/base/components/cart-base-components.module.ts:L42-L46). Altri provider di `PAGE_LAYOUT_HANDLER` si trovano nel product configurator (feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-layout.module.ts:L86-L86, feature-libs/product-configurator/rulebased/root/variant/variant-configurator-overview-layout.module.ts:L62-L62).

3.5 `PageTemplateDirective`: la classe CSS del template

`PageTemplateDirective` (`[cxPageTemplateStyle]`) aggiunge all'host una classe CSS uguale al nome del page template (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L22-L47). Se è applicata a un `ng-template` (come in `PageLayoutComponent`), l'host è il **parent element** (`templateRef.elementRef.nativeElement.parentElement`, L147-L151): in pratica il tag `cx-page-layout`.

- Con input valorizzato usa quel valore; un input falsy pulisce la classe solo se era stata impostata da input (L66-L74).
- Senza input, in `ngOnInit` si abbona a `pageLayoutService.templateName$` (L96-L104).
- `addStyleClass` rimuove la classe precedente, aggiunge la nuova con `classList.add`, la persiste con `DirectiveStateTransferService.set` e chiama `markForCheck()`; il JSDoc spiega che non usa `@HostBinding` per non cancellare classi già presenti (L106-L129).
- `currentTemplate` è inizializzato leggendo l'attributo trasferito dal server (`DirectiveStateTransferService.get(host, 'page-template')`, L48-L51 e L83-L87), così dopo l'idratazione la direttiva sa quale classe rimuovere al primo cambio di template.

`DirectiveStateTransferService.set` scrive `data-cx-state-transfer-<key>` **solo lato server**; `get` legge l'attributo ovunque (core-libs/storefront/utils/directive-state-transfer.service.ts:L23-L41).

3.6 `PageSlotComponent` e `PageSlotService`

`PageSlotComponent` ha selettore `cx-page-slot,[cx-page-slot]`, è `OnPush` (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L48-L54) e il JSDoc dichiara le classi CSS che gestisce: la `position`, `cx-pending` "for as long as the slot hasn't start loading" (nel codice, in realtà, finché tutti i componenti non hanno emesso `loaded(true)`, vedi sotto: core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L45-L45, :L170-L173), `page-fold` per lo slot di fold (L39-L47).

Stato e host binding. `position` è sia `@Input` sia `@HostBinding('attr.position')` e alimenta `position$` (L62-L69, L93). Gli altri binding: `class`, `class.page-fold` ← `isPageFold`, `class.cx-pending` ← `isPending` (default `true`), `class.has-components` ← `hasComponents`, che il commento dichiara non più usato ma tenuto per retrocompatibilità (L74-L91).

Dati. `slot$` = `position$` non-undefined → `cmsService.getContentSlot(position)` → `distinctUntilChanged(this.isDistinct)`; `components$` mappa `slot.components ?? []` (L97-L106). `isDistinct` considera uguali due slot con gli stessi `uid` nello stesso ordine (L201-L213).

Decorazione. In `ngOnInit` sottoscrive `slot$`, chiama `decorate(slot)`, assegna `components` e fa `markForCheck()` (L124-L131). `decorate` (L133-L164):

- sostituisce la vecchia position nella stringa `class` e aggiunge la nuova;
- imposta `pending` = numero di componenti dello slot (quindi `cx-pending` attivo se > 0) e `hasComponents`;
- scrive a mano l'attributo `class` con `Renderer2.setAttribute`, "for when change detection will not update host binding" (L149);
- applica i decorator di slot con `dynamicAttributeService.addAttributesToSlot` (che esegue in sequenza i decorator registrati: `core-libs/core/src/cms/services/dynamic-attribute.service.ts:L40-L65`).

Template (`core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L1-L17`): due livelli di outlet.

1. Un outlet esterno con nome = `position` e contesto `{ components$ }`: si può sostituire un intero slot.
2. Per ogni componente **con** `flexType`, un outlet interno con nome = `flexType`, contesto `{ component }`, `cxOutletDefer` = `getComponentDeferOptions(flexType)` e `(loaded)="isLoaded($event)"`; dentro, `<ng-container [cxComponentWrapper]="component" />`.

Defer e conteggio. Con `cxOutletDefer` valorizzato, `OutletDirective` emette `loaded(false)`, aspetta `DeferLoaderService.load(host, options)`, poi esegue `build()` ed emette `loaded(true)` (`core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L85-L116`). L'output `loaded` è un `EventEmitter` asincrono (`new EventEmitter<boolean>(true)`, `core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L58-L58`), che in Angular consegna i valori dentro un `setTimeout` (`angular:packages/core/src/event_emitter.ts:L156-L189`). `isLoaded(true)` decrementa `pending`; a zero `isPending` diventa `false` e la classe `cx-pending` sparisce (`core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L170-L188`).

`DeferLoaderService` carica subito se è SSR, se la strategia dell'elemento è `INSTANT`, oppure se la strategia globale (default `INSTANT`) è `INSTANT` e l'elemento non chiede `DEFER` (`core-libs/storefront/layout/loading/defer-loader.service.ts:L31-L31`, `:L55-L64`).

`PageSlotService` **e passaggio SSR → CSR.** Nel costruttore (`core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L13-L25`) chiama `resolvePrerenderedSlots()`, che **solo nel browser** raccoglie le `position` dei `cx-page-slot` già presenti nel DOM (quindi arrivati dall'SSR) la cui parte superiore cade nel viewport iniziale (L30-L42). `getComponentDeferOptions(slot, type)` ritorna `INSTANT` se lo slot è fra quelli prerenderizzati, altrimenti il `deferLoading` del tipo da `CmsComponentsService.getDeferLoadingStrategy` (L63-L73). `shouldNotDefer` restituisce `true` **una sola volta per slot**, perché lo rimuove con `splice` (L51-L57). Lo scopo, scritto nel commento, è evitare lo sfarfallio degli slot visibili nel passaggio da SSR a CSR (L44-L50).

3.7 `CmsComponentsService`: mapping, feature lazy e regole

`CmsComponentsService` (root) mantiene `missingComponents`, `mappings` (già risolti), `staticCmsConfig` e `mappingResolvers` (`core-libs/storefront/cms-structure/services/cms-components.service.ts:L36-L51`). Nel

costruttore cattura `cmsComponents` da `configInitializer.getStable('cmsComponents')` come **config statica**, per non dipendere da ciò che i moduli lazy aggiungeranno alla config globale (L55-L69).

`determineMappings(types)` (L79-L115), dentro `defer`, per ogni tipo non ancora mappato:

- se `featureModules.hasFeatureFor(type)` → accoda un resolver della feature;
- altrimenti → usa la config statica, se c'è.

Se ci sono resolver li esegue con `forkJoin` e poi emette i tipi; altrimenti emette subito `of(types)`.

Il resolver della feature (`getFeatureMappingResolver`, L117-L141) chiede a `CmsFeaturesService.getCmsMapping(type)` il mapping dichiarato dal modulo e salva `deepMerge({}, featureMapping, staticConfig)`: la config della feature fa da default e quella dell'app la sovrascrive (commento a L127-L128). Il resolver è condiviso con `share()` ed è rimosso dalla Map dopo l'uso.

Gli altri metodi:

METODO	COMPORTAMENTO	RIFERIMENTO
`getMapping(type)`	`mappings[type]` o fallback alla config statica; in dev mode logga una volta "No component implementation found..."	core-libs/storefront/cms-structure/services/cms-components.service.ts:L165-L181
`getModule(type)`	`NgModuleRef` della feature, solo se il tipo è coperto da una feature	core-libs/storefront/cms-structure/services/cms-components.service.ts:L149-L153
`shouldRender(type)`	`false` solo su server con `disableSSR`	core-libs/storefront/cms-structure/services/cms-components.service.ts:L187-L190
`getDeferLoadingStrategy(type)`	`deferLoading` letto **solo** dalla config statica	core-libs/storefront/cms-structure/services/cms-components.service.ts:L195-L200
`getChildRoutes(types)`	`childRoutes` dei tipi renderizzabili, normalizzati	core-libs/storefront/cms-structure/services/cms-components.service.ts:L205-L214
`standardizeChildRoutes`	unisce array e oggetti `{ parent, children }`; vince l'ultimo `parent`	core-libs/storefront/cms-structure/services/cms-components.service.ts:L231-L248
`getStaticData(type)`	`getMapping(type)?.data`	core-libs/storefront/cms-structure/services/cms-components.service.ts:L219-L223
`getGuards(types)`	unione deduplicata, **senza** filtro `shouldRender`	core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261
`getI18nKeys(types)`	unione deduplicata dei tipi renderizzabili	core-libs/storefront/cms-structure/services/cms-components.service.ts:L266-L276

`CmsFeaturesService`

`CmsFeaturesService` costruisce la mappa tipo → feature da `getStable('featureModules')`, includendo solo feature non-stringa con `module` e `cmsComponents` non vuoti (core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76). `getCmsMapping(type)` risolve l'istanza della feature (`resolveFeatureInstance`, che memorizza `featureModules.resolveFeature(name)` con `shareReplay()`: L134-L152) e ne restituisce `componentsMappings[type]`, oppure `of(undefined)` (L89-L103). `createFeatureInstance`

estrae, per ogni tipo dichiarato in `featureModules[name].cmsComponents`, la voce `cmsComponents[type]` dalla configurazione **del modulo feature** (L157-L183), letta con `{ self: true }` da `ConfigChunk` e `DefaultConfigChunk` del suo injector e fusa (default prima, poi config: L188-L205). `getModule(type)` restituisce il `moduleRef` solo per feature già risolte, con subscribe e unsubscribe sincroni (L110-L127).

3.8 `ComponentWrapperDirective`: dal dato al componente

`ComponentWrapperDirective` (`[cxComponentWrapper]`) riceve un `ContentSlotComponentData` ed espone l'output `cxComponentRef` (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L39-L45). Il suo campo `cmpRef` è dichiarato "unsafe": può arrivare tardi (lazy/defer) o mai (web component) (L47-L52).

1. `ngOnInit` chiama `determineMappings([flexType])` e, se `shouldRender(flexType)`, `launchComponent()` (L67-L79).
2. `launchComponent` legge il mapping e, se manca, esce (L81-L88).
3. Chiede un *launcher* a `ComponentHandlerService.getLauncher` passando il mapping, il `ViewContainerRef`, l'injector costruito da `CmsInjectorService.getInjector(flexType, uid, this.injector)` e il modulo della feature (`getModule`) (L90-L102).
4. Alla emissione del launcher salva `cmpRef`, emette `cxComponentRef`, invia `ComponentCreateEvent`, applica i decorator di componente e fa `markForCheck()` sul `ChangeDetectorRef` del proprio injector; con `finalize` invia `ComponentDestroyEvent` (L103-L116).
5. `ngOnDestroy` disiscrive `launcherResource`: l'handler distrugge il componente e scatta il `finalize` (L146-L150).

Il payload degli eventi contiene `typeCode` e `id` (= `uid`); solo `ComponentCreateEvent` porta anche `host` (`elementRef.nativeElement`). Gli eventi passano da `eventService.dispatch(payload, event)` (L124-L136). Le classi sono in core-libs/storefront/cms-structure/page/component/events/component.event.ts:L9-L48 (`type = 'ComponentCreate'` / `'ComponentDestroy'`).

Injector per componente: `CmsInjectorService`

`getInjector(type, uid, parentInjector?)` crea con `Injector.create` un injector figlio con tre gruppi di provider (core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50):

- `CmsComponentData` → `{ uid, data$: ComponentDataProvider.get(uid, type) }`;
- `provideLcpPresenceForCmsComponent()` (§3.11);
- i `providers` del mapping (`CmsComponentMapping.providers`, documentati come servizi non singleton con scope sull'albero del componente: core-libs/core/src/cms/config/cms-config.ts:L60-L73).

Il parent è l'injector del wrapper (quindi l'albero degli element injector dello slot). `Injector.create` in Angular 21 restituisce un `DestroyableInjector` (angular:packages/core/src/di/injector.ts:L100-L104).

`CmsComponentData<T>` è una classe astratta usata come token con `uid` e `data$` (core-libs/storefront/cms-structure/page/model/cms-component-data.ts:L10-L13). `ComponentDataProvider.get(uid, type?)` (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L35-L59):

- con `uid` e dati statici (`mapping.data`): emette prima i dati statici (`startWith`), poi `{ ...static, ...cms }`;
- con `uid` senza dati statici: solo `cmsService.getComponentData(uid)`;
- senza `uid`: `of(static)` oppure `EMPTY`.

Scelta dell'handler

`ComponentHandlerService` inietta l'array `@Optional() @Inject(ComponentHandler)` e sceglie con `resolveApplicable(this.handlers, [componentMapping])` (core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L33-L52); in dev mode, se nessuno fa match, logga una volta per mapping "Can't resolve handler for component mapping: " (L54-L62); `getLauncher` restituisce `undefined` se manca l'handler (L74-L88).

`resolveApplicable` (core-libs/core/src/util/applicable.ts:L47-L87) filtra gli applicabili con `hasMatch` vero (o senza `hasMatch`); fra più candidati vince la priorità più alta e, a parità, quello fornito per ultimo; senza `getPriority` vale `Priority.NORMAL`. Valori: `FALLBACK = -50`, `LOW = -10`, `NORMAL = 0`, `HIGH = 10` (core-libs/core/src/util/applicable.ts:L7-L24).

HANDLER	HASMATCH	PRIORITÀ	RIFERIMENTO
`DefaultComponentHandler`	`typeof component === 'function'`	FALLBACK	core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L24-L34
`LazyComponentHandler`	funzione il cui `toString()` inizia con `function()` o `()=>`	LOW	core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L35-L52
`WebComponentHandler`	stringa che contiene `#`	LOW	core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L27-L47

Il contratto comune è scritto nel JSDoc di `ComponentHandler`: abbonarsi al launcher istanzia il componente, l'emissione porta `ElementRef` (e `ComponentRef` per i componenti Angular), la disiscrizione lo distrugge (core-libs/storefront/cms-structure/page/component/handlers/component-handler.ts:L22-L35).

- **Default**: usa `elementInjector ?? viewContainerRef.injector`, risolve una `ComponentFactory` con `ComponentFactoryResolver` e chiama `viewContainerRef.createComponent(factory, undefined, injector, undefined, module)`; il teardown è `componentRef.destroy()` (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L42-L85). Sia `ComponentFactoryResolver` sia questo overload di `createComponent` sono `@deprecated` in Angular 21 (angular:packages/core/src/linker/component_factory_resolver.ts:L30-L41, angular:packages/core/src/linker/view_container_ref.ts:L263-L275).
- **Lazy**: dentro `defer` registra un task con `PendingTasks.add()`, risolve la Promise di `component()` e rimuove il task in `finalize`; poi passa la classe risolta al `DefaultComponentHandler` (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L54-L81). Il commento spiega che senza il task l'SSR serializzerebbe e distruggerebbe l'app mentre l'`import()` è ancora in volo (NG0205). `PendingTasks.add()` restituisce una funzione di cleanup idempotente (angular:packages/core/src/pending_tasks.ts:L42-L60).
- **Web component**: crea l'elemento custom con `Renderer2`, gli assegna `cxApi = { ...CxApiService, cmsComponentData }` e lo appende al `parentElement` dell'anchor del VCR; emette solo `elementRef` (core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L49-L96). `initWebComponent` divide `path#selector`, carica lo script una sola volta (cache `loadedWebComponents`) e concatena le `onload` se lo script è ancora in caricamento (L98-L136). `CxApiService` espone come proprietà pubbliche opzionali numerosi facade core più `ngZone` (core-libs/storefront/cms-structure/page/component/services/cx-api.service.ts:L33-L75).

Componenti annidati: `InnerComponentsHostDirective`

`[cxInnerComponentsHost]` legge `data$.composition.inner` dal `CmsComponentData` iniettato (core-libs/storefront/cms-structure/page/component/inner-components-host.directive.ts:L28-L33) e, per ogni nome, crea a mano (`new`) un `ComponentWrapperDirective` con `cxComponentWrapper = { flexType: component, uid: '' }` e ne chiama `ngOnInit()`; a ogni nuova lista distrugge i wrapper precedenti (L57-L81). È usata per esem-

pio nella search box e negli item di product grid/list/carousel (core-libs/storefront/cms-components/product/product-list/product-grid-item/product-grid-item.component.html:L45-L45, core-libs/storefront/cms-components/navigation/search-box/search-box.component.html:L213-L213).

3.9 Child route CMS (`CmsRoutesImplService`)

Un componente CMS può dichiarare `childRoutes` nel mapping (`Route[]` o `{ parent?: Pick<Route, 'data'>; children?: Route[] }`: core-libs/core/src/cms/config/cms-config.ts:L48-L58). `CmsRoutesService` è la facciata pubblica astratta con `useExisting: CmsRoutesImplService` (core-libs/storefront/cms-structure/services/cms-routes.service.ts:L11-L24); l'implementazione non è esportata in attesa del refactor della issue #7070 (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L33).

`handleCmsRoutesInGuard` (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L91):

1. se esiste già una route CMS per la label (`cmsRouteExists`: URL che inizia con `/` e route in `router.config` con `data.cxCmsRouteContext` e `path` uguale, L42-L58) → `true`;
2. se i componenti hanno child route e `updateRouting` riesce → `router.navigateByUrl(currentUrl)` e ritorna `false` (la navigazione corrente è annullata e rifatta);
3. altrimenti → `true`.

`updateRouting` (L93-L124) agisce solo per `PageType.CONTENT_PAGE` con label che inizia per `/` e lunga più di 1: crea `{ path: label senza '/', component: PageLayoutComponent, children, data: deepMerge(parent.data, { cxCmsRouteContext: { type, id } }) }` e la mette **in testa** con `router.resetConfig`. Le guard delle child route vengono avvolte ricorsivamente (`wrapCmsGuardsRecursively`, L126-L155) da un wrapper che cerca un'istanza di classe con `UnifiedInjector` e, se non la trova, tratta la guard come `CanActivateFn`, eseguendola con `GuardsComposer` (L164-L182). Il JSDoc spiega lo scopo: permettere al router di usare guard fornite solo nell'injector di un modulo lazy.

Alla seconda navigazione la route ha `cxCmsRouteContext`, quindi `canActivatePage` non richiama `handleCmsRoutesInGuard` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L88-L88).

3.10 `cms-pages`: route PDP e PLP

`ProductDetailsPageModule` registra una route `path: null` (con `@ts-ignore`), `canActivate: [CmsPageGuard]`, `component: PageLayoutComponent`, `data: { cxRoute: 'product' }` (core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L14-L24) e fornisce come default `routing.routes.product.matchers = [PRODUCT_DETAILS_URL_MATCHER]` (L26-L36). Il token combina il `DEFAULT_URL_MATCHER` con un suffix matcher `marker: 'p', paramName: 'productCode'` tramite `UrlMatcherService.getCombined` (core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L16-L50). Il JSDoc ricorda che i matcher "matchano" una route ma non la creano né la proteggono (L31-L41).

`ProductListingPageModule` registra tre route `path: null` → `PageLayoutComponent` con `CmsPageGuard`: `search` (`pageLabel: 'search'`), `brand` e `category`, quest'ultima **per ultima** perché il suo matcher `**/c/:categoryCode` è greedy e oscurerebbe `brand` (core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L14-L39). Il default è `routing.routes.category.matchers = [PRODUCT_LISTING_URL_MATCHER]` (L41-L51), con suffix matcher `marker: 'c', paramName: 'categoryCode'` (core-libs/storefront/cms-pages/product-listing-page/product-listing-url-matcher.ts:L20-L28).

Nell'app demo entrambi i moduli sono importati in `SpartacusFeaturesModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L224-L225). La sostituzione di `path: null` con i path/matcher configurati avviene nel routing core (cap. 14): `ConfigurableRoutesService.configureRoute` usa `data.cxRoute`

come nome della route e, se la config ha `matchers`, cancella `path` e imposta `matcher` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L94).

3.11 Contesto LCP

Obiettivo (JSDoc di `LcpCmsComponentsConfig`): sapere quali componenti contengono l'elemento LCP per ottimizzarne il caricamento, per esempio dando priorità all'immagine principale (core-libs/storefront/cms-structure/cms-lcp-context/config/lcp-cms-components.config.ts:L19-L26).

1. `LCP_PRESENCE` è un `InjectionToken<Observable<LcpPresence>>` root con default `of(LcpPresence.NO_LCP)`, pensato per essere ri-fornito a livello DOM da un antenato (core-libs/storefront/shared/lcp-context/lcp-presence.token.ts:L11-L32).
2. Per ogni componente CMS, `CmsInjectorService` aggiunge `provideLcpPresenceForCmsComponent()` (core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L45-L45): una factory che da `CmsComponentData.data$` calcola `CmsLcpService.getLcpPresence(data)` con `distinctUntilChanged` e `shareReplay({ bufferSize: 1, refCount: true })` (core-libs/storefront/cms-structure/cms-lcp-context/provide-lcp-presence-for-cms-component.ts:L14-L36).
3. `CmsLcpService.getLcpPresence` ritorna `HAS_LCP` se l'`uid` contiene `idMarker` o è in `ids`, altrimenti `NO_LCP` (core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L50-L65).
4. Un discendente usa `*cxLcpContext`, che inietta `LCP_PRESENCE` e ricava `fetchPriority$` con `LcpPresenceMappingService` (`HAS_LCP` → `ImageFetchPriority.HIGH`, altrimenti `undefined`) e crea la view con contesto `$implicit: { lcpPresence$, fetchPriority$ }` (core-libs/storefront/shared/lcp-context/lcp-context.directive.ts:L45-L76, core-libs/storefront/shared/lcp-context/lcp-presence-mapping.service.ts:L14-L22).
5. Fuori dai componenti CMS si può fornire il valore a mano con `[cxProvideLcpPresence]` (core-libs/storefront/shared/lcp-context/provide-lcp-presence.directive.ts:L24-L47).

3.12 `provideCmsStructure`

Helper per aggiungere a una pagina un componente che il backend non fornisce: `provideCmsStructure(options)` restituisce un `provideConfig` che fonde `buildCmsStructure` (`cmsStructure.components[id] = { typeCode, flexType }` ed eventualmente `cmsStructure.slots[pos] = { componentIds: [id] }`) e `buildLayoutConfig` (`layoutSlots[template]` e/o `layoutSlots[section]` con `slots: [pos]`, eventualmente sotto la chiave del breakpoint) (core-libs/storefront/cms-structure/utils/cms-structure.util.ts:L31-L108). Il JSDoc propone come esempio `{ componentId: 'LoginComponent', pageSlotPosition: 'SiteLogin' }` (L12-L30). Il trattamento di `cmsStructure` lato dati è nel capitolo 15.

---

4. Diagrammi

4.1 Pipeline completa: dalla route al componente

Ogni nodo corrisponde ai passi di §3.2-§3.8.

■ **TAVOLA 16.1** diagramma a tutta pagina, in fondo al capitolo

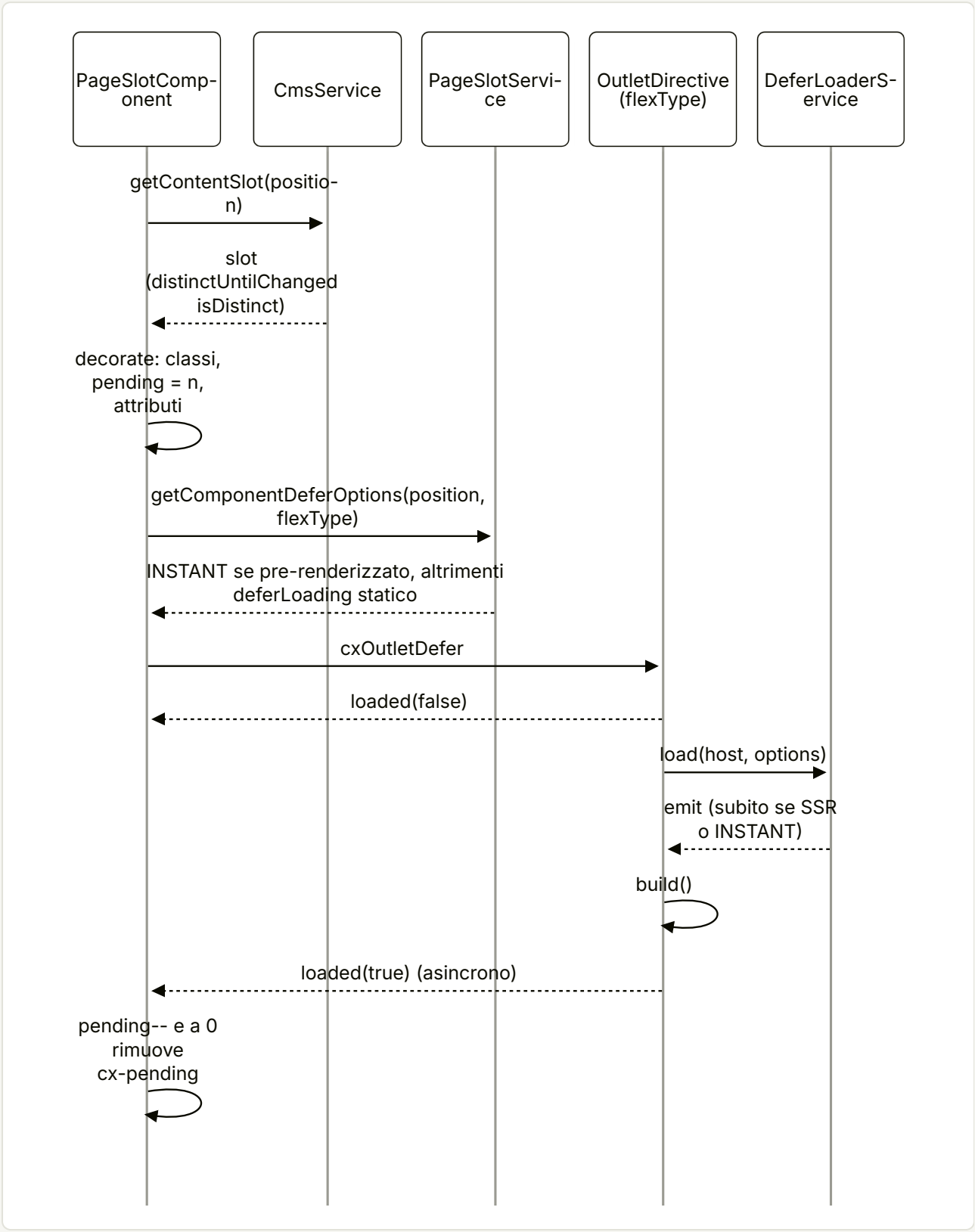
4.2 Lancio di un componente (`ComponentWrapperDirective`)

■ TAVOLA 16.2 diagramma a tutta pagina, in fondo al capitolo

4.3 Risoluzione degli slot (`resolveSlots` / `getSlotConfig`)

■ TAVOLA 16.3 diagramma a tutta pagina, in fondo al capitolo

4.4 Ciclo di vita di uno slot e defer



5. Estratti di codice reale

Il template del layout — core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L1-L18

```HTML
<ng-template
  *ngIf="layoutName$ | async as layoutName"
  [cxPageTemplateStyle]="layoutName"
  [cxOutlet]="layoutName"
  [cxOutletContext]="{
    templateName$: templateName$,
    slots$: slots$,
    section$: section$,
  }"
>
  <ng-content />

  <cx-page-slot
    *ngFor="let slot of slots$ | async"
    [position]="slot"
    [isPageFold]="slot === (pageFoldSlot$ | async)"
  />
</ng-template>
```

Gli stream del layout — core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L31-L53

```TS
  @Input() set section(value: string) {
    this.section$.next(value);
  }
  readonly section$ = new BehaviorSubject<string | undefined>(undefined);

  readonly templateName$: Observable<string> =
    this.pageLayoutService.templateName$;

  readonly layoutName$: Observable<string> = this.section$.pipe(
    switchMap((section) => (section ? of(section) : this.templateName$))
  );

  readonly slots$: Observable<string[]> = this.section$.pipe(
    switchMap((section) => this.pageLayoutService.getSlots(section))
  );

  readonly pageFoldSlot$: Observable<string | undefined> =
    this.templateName$.pipe(
      switchMap((templateName) =>
        this.pageLayoutService.getPageFoldSlot(templateName)
```

```
    ),
    distinctUntilChanged()
  );
```

Catena degli handler di layout — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L54-L67

```ts
getSlots(section?: string): Observable<string[]> {
  return combineLatest([this.page$, this.breakpointService.breakpoint$]).pipe(
    map(([page, breakpoint]) => {
      const pageTemplate = page.template;
      const slots = this.resolveSlots(page, section, breakpoint);
      return { slots, pageTemplate, breakpoint };
    }),
    switchMap(({ slots, pageTemplate, breakpoint }) => {
      let result = of(slots);
      for (const handler of this.handlers || []) {
        result = handler.handle(result, pageTemplate, section, breakpoint);
      }
      return result;
    }),
```

I tre rami di `resolveSlots` — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L111-L126

```ts
  const config = this.getSlotConfig(
    page.template ?? '',
    'slots',
    section,
    breakpoint
  );
  if (config && config.slots) {
    const pageSlots = page.slots ? Object.keys(page.slots) : [];
    return config.slots.filter((slot) => pageSlots.includes(slot));
  } else if (!section) {
    this.logMissingLayoutConfig(page);
    return page.slots ? Object.keys(page.slots) : [];
  } else {
    this.logMissingLayoutConfig(page, section);
    return [];
  }
```

Breakpoint "più vicino" — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L229-L242

```ts
    // we have a config for the specific breakpoint
    if (layoutSlotConfig[breakpoint]?.hasOwnProperty(configAttribute)) {
      return <SlotConfig>layoutSlotConfig[breakpoint];
    }

    // find closest config
    const all = this.breakpointService.breakpoints;

    for (const br of all.slice(0, all.indexOf(breakpoint))) {
      if (layoutSlotConfig[br]?.hasOwnProperty(configAttribute)) {
        slotConfig = <SlotConfig>layoutSlotConfig[br];
      }
    }
    return slotConfig;
```

Il template dello slot — core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L1-L17

```html
<ng-template
  *ngIf="position"
  [cxOutlet]="position"
  [cxOutletContext]="{ components$: components$ }"
>
  <ng-container *ngFor="let component of components">
    <ng-template
      *ngIf="component.flexType"
      [cxOutlet]="component.flexType"
      [cxOutletContext]="{ component: component }"
      [cxOutletDefer]="getComponentDeferOptions(component.flexType)"
      (loaded)="isLoaded($event)"
    >
      <ng-container [cxComponentWrapper]="component" />
    </ng-template>
  </ng-container>
</ng-template>
```

Slot pre-renderizzati e defer — core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L51-L73

```ts
  shouldNotDefer(slot: string): boolean {
    if (this.prerenderedSlots?.includes(slot)) {
      this.prerenderedSlots.splice(this.prerenderedSlots.indexOf(slot), 1);
      return true;
    }
    return false;
  }

  /**
   * Returns the defer options for the given component. If the wrapping
```

```
   * page slot is prerendered, we would ignore the defer options altogether.
   */
  getComponentDeferOptions(
    slot: string | undefined,
    componentType: string
  ): IntersectionOptions {
    if (slot && this.shouldNotDefer(slot)) {
      return { deferLoading: DeferLoadingStrategy.INSTANT };
    }
    const deferLoading =
      this.cmsComponentsService.getDeferLoadingStrategy(componentType);
    return { deferLoading };
  }
```

Il cuore del wrapper — `core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L81-L117`

```ts
  private launchComponent() {
    const componentMapping = this.cmsComponentsService.getMapping(
      this.cxComponentWrapper.flexType ?? ''
    );

    if (!componentMapping) {
      return;
    }

    this.launcherResource = this.componentHandler
      .getLauncher(
        componentMapping,
        this.vcr,
        this.cmsInjector.getInjector(
          this.cxComponentWrapper.flexType ?? '',
          this.cxComponentWrapper.uid ?? '',
          this.injector
        ),
        this.cmsComponentsService.getModule(
          this.cxComponentWrapper.flexType ?? ''
        )
      )
      ?.pipe(
        filter(isNotUndefined),
        tap(({ elementRef, componentRef }) => {
          this.cmpRef = componentRef;

          this.cxComponentRef.emit(componentRef);

          this.dispatchEvent(ComponentCreateEvent, elementRef);
          this.decorate(elementRef);
```

```
        this.injector.get(ChangeDetectorRef).markForCheck();
      }),
      finalize(() => this.dispatchEvent(ComponentDestroyEvent))
    )
    .subscribe();
}
```

L'injector per componente — core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50

```ts
public getInjector(
  type: string,
  uid: string,
  parentInjector?: Injector
): Injector {
  const configProviders =
    this.cmsComponentsService.getMapping(type)?.providers ?? [];
  return Injector.create({
    providers: [
      {
        provide: CmsComponentData,
        useFactory: (dataProvider: ComponentDataProvider) => ({
          uid,
          data$: dataProvider.get(uid, type),
        }),
        deps: [ComponentDataProvider],
      },
      provideLcpPresenceForCmsComponent(),
      ...configProviders,
    ],
    parent: parentInjector ?? this.injector,
  });
}
```

Handler lazy con PendingTasks — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L80

```ts
  return defer(() => {
    // Register the dynamic import as a pending task so Angular's stability
    // counter stays > 0 until the chunk is loaded. This prevents SSR from
    // serializing and destroying the app while the import() is still in flight
    // (which would cause NG0205 — destroyed injector). The task is removed by
    // finalize(), which fires on complete, error, and unsubscribe.

    const removeTaskTracking = this.pendingTasks.add();
    return from(componentMapping.component()).pipe(
      finalize(removeTaskTracking)
    );
```

```
    }).pipe(
      switchMap((component) =>
        this.defaultHandler.launcher(
          { ...componentMapping, component },
          viewContainerRef,
          elementInjector,
          module
        )
      )
    );
```

Handler di default — core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L48-L72

```ts
    const injector = elementInjector ?? viewContainerRef.injector;

    const dispose = () => {
      if (componentRef) {
        componentRef.destroy();
      }
    };

    const factory = this.getComponentFactory(
      injector,
      componentMapping.component
    );

    if (factory) {
      componentRef = viewContainerRef.createComponent(
        factory,
        undefined,
        injector,
        undefined,
        module
      );
      subscriber.next({ elementRef: componentRef.location, componentRef });
    }

    return dispose;
```

Scelta per priorità — core-libs/core/src/util/applicable.ts:L63-L87

```ts
export function resolveApplicable<T extends Applicable>(
  applicables: T[],
  matchParams: any[] = [],
  priorityParams: any[] = []
): T | undefined {
  const matchedApplicables = (applicables ?? []).filter(
```

```
    (applicable) => !applicable.hasMatch || applicable.hasMatch(...matchParams)
  );

  if (matchedApplicables.length < 2) {
    return matchedApplicables[0];
  }

  let lastPriority = -Infinity;
  return matchedApplicables.reduce((acc: T | undefined, curr: T) => {
    const currPriority = curr.getPriority
      ? curr.getPriority(...priorityParams)
      : Priority.NORMAL;
    if (lastPriority > currPriority) {
      return acc;
    }
    lastPriority = currPriority;
    return curr;
  }, undefined);
}
```

Mapping da feature: la feature è il default — core-libs/storefront/cms-structure/services/cms-components.service.ts:L122-L139

TS

```
      const mappingResolver$ = this.featureModules
        .getCmsMapping(componentType)
        .pipe(
          filter(isNotUndefined),
          tap((featureComponentMapping) => {
            // We treat cms mapping configuration from a feature as a default,
            // that can be overridden by app/static configuration
            this.mappings[componentType] = deepMerge(
              {},
              featureComponentMapping,
              staticConfig
            );
            this.mappingResolvers.delete(componentType);
          }),
          share()
        );
```

Dati statici + dati CMS — core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L43-L57

TS

```
      if (uid) {
        if (staticComponentData) {
          return this.cmsService.getComponentData<T>(uid).pipe(
            map((data) => ({
              ...staticComponentData,
```

```
            ...data,
          })),
          startWith(staticComponentData)
        );
      } else {
        return this.cmsService.getComponentData<T>(uid);
      }
    } else {
      return staticComponentData ? of(staticComponentData) : EMPTY;
    }
```

Composizione sequenziale delle guard — core-libs/storefront/cms-structure/services/guards-composer.ts:L78-L86

```ts
protected composeCanActivateObservables(
  canActivateObservables: CanActivateObservable[]
): CanActivateObservable {
  return concat(...canActivateObservables).pipe(
    skipWhile((canActivate: GuardResult) => canActivate === true),
    endWith(true),
    first()
  );
}
```

Generazione della route figlia — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L98-L120

```ts
  if (
    pageContext.type === PageType.CONTENT_PAGE &&
    pageLabel.startsWith('/') &&
    pageLabel.length > 1
  ) {
    const children = this.wrapCmsGuardsRecursively(
      childRoutesConfig.children ?? []
    );

    const newRoute: CmsRoute = {
      path: pageLabel.substring(1),
      component: PageLayoutComponent,
      children: children,
      data: deepMerge({}, childRoutesConfig?.parent?.data ?? {}, {
        cxCmsRouteContext: {
          type: pageContext.type,
          id: pageLabel,
        },
      }),
    };
```

```
      this.router.resetConfig([newRoute, ...this.router.config]);
      return true;
```

Presenza LCP per componente CMS — core-libs/storefront/cms-structure/cms-lcp-context/provide-lcp-presence-for-cms-component.ts:L20-L36

```ts
export const provideLcpPresenceForCmsComponent = (): FactoryProvider => {
  return {
    provide: LCP_PRESENCE,
    useFactory: (): Observable<LcpPresence> => {
      const cmsComponentData = inject(CmsComponentData);
      const cmsLcpService = inject(CmsLcpService);

      return cmsComponentData.data$.pipe(
        switchMap((componentData) =>
          cmsLcpService.getLcpPresence(componentData)
        ),
        distinctUntilChanged(),
        shareReplay({ bufferSize: 1, refCount: true })
      );
    },
  };
};
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
layoutSlots	LayoutSlotConfig	nessuno nei moduli di questo capitolo	core-libs/storefront/layout/config/layout-config.ts:L102-L102	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L146-L173, :L181-L209, :L88-L104
layoutSlots[template|section].slots	string[]	—	core-libs/storefront/layout/config/layout-config.ts:L36-L46	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L111-L119

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
layoutSlots[template].pageFold	string	—	core-libs/storefront/layout/config/layout-config.ts:L36-L46	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L88-L104
layoutSlots[...][lg|md|sm|xs]	SlotGroup	—	core-libs/storefront/layout/config/layout-config.ts:L48-L57	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243
breakpoints	LayoutBreakPoints	xs: 576, sm: 768, md: 992, lg: 1200, xl: { min: 1200 } (core-libs/storefront/layout/config/default-layout.config.ts:L9-L19, fornito da LayoutModule: core-libs/storefront/layout/layout.module.ts:L22-L22)	core-libs/storefront/layout/config/layout-config.ts:L100-L100	core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L54-L65
deferredLoading.strategy	DeferLoadingStrategy	INSTANT se assente	core-libs/storefront/layout/config/layout-config.ts:L109-L114	core-libs/storefront/layout/loading/defer-loader.service.ts:L31-L31
cmsComponents[type].component	classe / () => import() / 'path#selector'	—	core-libs/core/src/cms/config/cms-config.ts:L60-L61	hasMatch degli handler (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L28-L30, core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L38-L48, core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L38-L43)
cmsComponents[type].providers	StaticProvider[]	[]	core-libs/core/src/cms/config/cms-config.ts:L73-L73	core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L33-L34

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cmsComponents[type].childRoutes	Route[] | CmsComponentChildRoutesConfig	[]	core-libs/core/src/cms/config/cms-config.ts:L74-L74	core-libs/storefront/cms-structure/services/cms-components.service.ts:L205-L214
cmsComponents[type].disableSSR	boolean	falsy	core-libs/core/src/cms/config/cms-config.ts:L75-L75	core-libs/storefront/cms-structure/services/cms-components.service.ts:L187-L190
cmsComponents[type].i18nKeys	string[]	—	core-libs/core/src/cms/config/cms-config.ts:L76-L76	core-libs/storefront/cms-structure/services/cms-components.service.ts:L266-L276
cmsComponents[type].guards	any[]	—	core-libs/core/src/cms/config/cms-config.ts:L77-L77	core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261
cmsComponents[type].data	T	—	core-libs/core/src/cms/config/cms-config.ts:L85-L85	core-libs/storefront/cms-structure/services/cms-components.service.ts:L219-L223 → core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L39-L41
cmsComponents[type].deferLoading	DeferLoadingStrategy	—	core-libs/core/src/cms/config/cms-config.ts:L92-L92	core-libs/storefront/cms-structure/services/cms-components.service.ts:L195-L200 (solo config statica)
featureModules[name]	FeatureModuleConfig | string	{}	core-libs/core/src/cms/config/cms-config.ts:L109-L129	core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
featureModules[name].cmsComponents	string[]	—	core-libs/core/src/cms/config/cms-config.ts:L121-L121	core-libs/storefront/cms-structure/services/cms-features.service.ts:L68-L72, :L177-L181
lcpCmsComponents.idMarker	string	'__cxLCP__'	core-libs/storefront/cms-structure/cms-lcp-context/config/lcp-cms-components.config.ts:L37-L37; default core-libs/storefront/cms-structure/cms-lcp-context/config/default-lcp-cms-components.config.ts:L11-L11	core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L51-L56
lcpCmsComponents.ids	string[]	[]	core-libs/storefront/cms-structure/cms-lcp-context/config/lcp-cms-components.config.ts:L46-L46; default core-libs/storefront/cms-structure/cms-lcp-context/config/default-lcp-cms-components.config.ts:L12-L12	core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L52-L62

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes.product.matchers	token UrlMatcherFactory[]	[PRODUCT_DETAILS_URL_MATCHER]	core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L26-L36	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L94
routing.routes.category.matchers	idem	[PRODUCT_LISTING_URL_MATCHER]	core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L41-L51	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L94
feature toggle a11yB2BRegisterComponent	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L58-L58, :L1021-L1021	core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L55-L57 (via useFeatureStyles)

Valori degli enum usati: BREAKPOINT = xs, sm, md, lg, xl (core-libs/storefront/layout/config/layout-config.ts:L22-L28); DeferLoadingStrategy.DEFER = 'DEFERRED-LOADING', INSTANT = 'INSTANT-LOADING' (core-libs/core/src/cms/config/cms-config.ts:L96-L101).

Esempio di configurazione (illustrativo, costruito con le sole chiavi documentate sopra; i nomi MyBannerComponent e my-banner.component non esistono nel repo):

```ts
provideConfig(<LayoutConfig & CmsConfig>{
  layoutSlots: {
    LandingPage2Template: {
      pageFold: 'Section2B',
      slots: ['Section1', 'Section2A', 'Section2B', 'Section3'],
      xs: { slots: ['Section1', 'Section3'] },
    },
    header: { slots: ['SiteLogo', 'SearchBox', 'MiniCart'] },
  },
  cmsComponents: {
    SimpleResponsiveBannerComponent: {
      component: () =>
        import('./my-banner.component').then((m) => m.MyBannerComponent),
```

```
      deferLoading: DeferLoadingStrategy.DEFER,
      data: { headline: 'default' },
    },
  },
});
```

Con questa config, su `xs` la pagina rende solo `Section1` e `Section3` (se esistono nella pagina CMS); su `sm` e superiori, dove non c'è una voce `slots` specifica, `getResponsiveSlotConfig` cerca fra i breakpoint più piccoli: trova `xs` e quindi usa ancora `['Section1', 'Section3']`. È il comportamento di T5 in §9: una config su `xs` "si propaga" verso l'alto finché non viene ridefinita. La voce `header` globale vale solo sulle pagine il cui template ha una voce in `layoutSlots` (qui solo `LandingPage2Template`); sulle altre pagine la section `header` rende `[]` (T2, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L181-L185).

---

7. Commenti e note notevoli dal sorgente

- "Prints warn messages for missing layout configs. The warnings are only printed once per config to not pollute the console log." — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L48-L50; e "the info log is not printed in production" (L256).
- `PageTemplateDirective`: "Persist the current template for rehydration. Without this, the current template would be unknown and the directive would not be able to remove old style classes." — core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L120-L121. Il JSDoc della stessa direttiva mostra gli esempi `<cx-storefront class="LandingPageTemplate">` e `[cxPageTemplateStyle]="pageTemplateName"` (L31-L44).
- `hasComponents`: "This is no longer used in spartacus, but kept for backwards compatibility." — core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L87-L91.
- "manually set classes for when change detection will not update host binding" — core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L149-L149.
- "instantiate PageSlotService ASAP, so it can examine SSR pre-rendered DOM" — core-libs/storefront/cms-structure/page/slot/page-slot.module.ts:L19-L19.
- `shouldNotDefer`: "It's especially useful when transitioning from SSR to CSR application, where we don't want to apply deferring logic to slots that are visible to avoid unnecessary flickering." — core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L44-L50.
- `cmpRef`: "This property in unsafe" (lazy/defer o web component) — core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L47-L51.
- Il commento SSR/NG0205 del `LazyComponentHandler` — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L61-L65 — e il refuso "We want to mach dynamic import signature () ⇒ import('')" (L35-L37).
- `WebComponentHandler`: "If script is still loading (has onload callback defined) add new callback and chain it with the existing one. Needed to support loading multiple components from one script" — core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L124-L126.
- "Contains already initialized resolvers for specified component typez" (refuso originale) — core-libs/storefront/cms-structure/services/cms-components.service.ts:L49-L49.
- "We treat cms mapping configuration from a feature as a default, that can be overridden by app/static configuration" — core-libs/storefront/cms-structure/services/cms-components.service.ts:L127-L128.

- `// TODO:#checkout - handle breaking changes in schematics` nel costruttore di `CmsGuardsService` — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L25-L25.
- `CmsRoutesService`: "Public injection token for the private implementation … After #7070, this class should be replaced with a real implementation." — core-libs/storefront/cms-structure/services/cms-routes.service.ts:L11-L12; e in `CmsRoutesImplService`: "This service should be exposed in public API only after the refactor planned in https://github.com/SAP/spartacus/issues/7070" (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L31).
- `GuardsComposer`: "Replacement for the Angular's deprecated type `CanActivate`." — core-libs/storefront/cms-structure/services/guards-composer.ts:L16-L18.
- Due `// eslint-disable-next-line @typescript-eslint/no-non-null-assertion` in `CmsFeaturesService` — core-libs/storefront/cms-structure/services/cms-features.service.ts:L161-L161, :L178-L178.
- PLP: "The 'category' route may include a greedy suffix url matcher '**/c/:categoryCode' So not to shadow the specific 'brand' route, the 'category' is the last route in the sequence." — core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L31-L32.
- `// @ts-ignore` su `path: null` — core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L18-L19, core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L17-L18 (e anche :L24-L25, :L33-L34).
- Il JSDoc di `CmsLcpService` cita `config.lcpCmsComponentIdMarker`, chiave che non esiste nel codice (quella reale è `lcpCmsComponents.idMarker`) — core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L21-L24.
- Il JSDoc di `ProvideLcpPresenceDirective` dice che per i componenti CMS la presenza LCP "is already defined by the `[cxComponentWrapper]` directive"; nel codice è fornita dall'injector creato da `CmsInjectorService`, che il wrapper usa — core-libs/storefront/shared/lcp-context/provide-lcp-presence.directive.ts:L17-L23, core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L45-L45.
- Nessun `@deprecated` nei 66 file del capitolo (grep sull'elenco `outline/16.files.txt`); le deprecazioni rilevanti sono lato Angular (`ComponentFactoryResolver`, overload di `createComponent`, §3.8).

---

8. Extension point

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`cmsComponents[type]` (mapping)	sostituire/aggiungere un componente per `flexType`: classe, `() => import()`, con `providers`, `data`, `guards`, `childRoutes`, `i18nKeys`, `disableSSR`, `deferLoading`	core-libs/core/src/cms/config/cms-config.ts:L60-L93	51 — sostituire o creare un componente CMS
`CmsComponentData<T>`	iniettarlo nel proprio componente per leggere `uid` e `data$`	core-libs/storefront/cms-structure/page/model/cms-component-data.ts:L10-L13	51 — estendere un componente CMS
`layoutSlots` / `LayoutConfig`	ordine e visibilità degli slot per template, section, breakpoint; `pageFold`	core-libs/storefront/layout/config/layout-config.ts:L36-L122	51 — page layout

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`PAGE_LAYOUT_HANDLER`	multi-provider: trasformare l'elenco degli slot in base allo stato (es. carrello)	`core-libs/storefront/cms-structure/page/page-layout/page-layout-handler.ts:L11-L22`; esempio `feature-libs/cart/base/components/cart-base-components.module.ts:L42-L46`	51 — page layout
`PageLayoutService`	servizio root con metodi `protected` (`getSlotConfig`, `getSlotConfigForSection`, `getResponsiveSlotConfig`) sostituibile via DI	`core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L146-L243`	50 — override di servizi via DI
`<cx-page-layout section="...">`	rendere una sezione (header, footer, ...) con gli stessi meccanismi	`core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L31-L34`	51
Outlet `layoutName`, `position`, `flexType`	sostituire/affiancare un layout intero, uno slot, un tipo di componente	`core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L1-L10`, `core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L1-L13`	51 — outlet, cap. 17
`ComponentHandler`	multi-provider di un nuovo handler (`hasMatch` + `getPriority` + `launcher`) per nuovi formati di `component`	`core-libs/storefront/cms-structure/page/component/handlers/component-handler.ts:L17-L45`; registrazione `core-libs/storefront/cms-structure/page/component/page-component.module.ts:L23-L40`	50 — multi provider
`ComponentCreateEvent` / `ComponentDestroyEvent`	ascoltare la vita dei componenti CMS con `EventService` (es. tracking)	`core-libs/storefront/cms-structure/page/component/events/component.event.ts:L29-L48`	50 — eventi, Appendice eventi
output `cxComponentRef`	ottenere il `ComponentRef` quando si usa `cxComponentWrapper` a mano	`core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L45-L45`	51
`cxInnerComponentsHost`	rendere componenti annidati da `composition.inner`	`core-libs/storefront/cms-structure/page/component/inner-components-host.directive.ts:L28-L33`	51
`featureModules[name].cmsComponents`	spostare un gruppo di componenti in un feature module lazy	`core-libs/core/src/cms/config/cms-config.ts:L109-L129`	53 — feature lib lazy
`provideCmsStructure`	aggiungere un componente/slot che il backend non fornisce	`core-libs/storefront/cms-structure/utils/cms-structure.util.ts:L31-L38`	51
`childRoutes` + `CmsRoutesService`	sotto-route sotto una content page CMS	`core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L124`	51 — route custom

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`PRODUCT_DETAILS_URL_MATCHER`, `PRODUCT_LISTING_URL_MATCHER`	riusare o sostituire i matcher di PDP/PLP in `routing.routes.*.matchers`	`core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L42-L50`, `core-libs/storefront/cms-pages/product-listing-page/product-listing-url-matcher.ts:L42-L50`	51 — route custom
`CmsLcpService` / `lcpCmsComponents`	marcare componenti LCP per id/marker o con logica custom (estensione documentata nel JSDoc con `CustomCmsLcpService`)	`core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L26-L65`	50 — override di servizi
`LCP_PRESENCE`, `*cxLcpContext`, `[cxProvideLcpPresence]`	leggere/fornire la presenza LCP nei propri template	`core-libs/storefront/shared/lcp-context/lcp-presence.token.ts:L26-L32`	51

Cosa **non** è pubblico (non usarlo in codice applicativo; vedi 54 — API pubblica vs interna): `ComponentDataProvider` e `CxApiService` (`core-libs/storefront/cms-structure/page/component/services/index.ts:L7-L8`), `WebComponentHandler` (`core-libs/storefront/cms-structure/page/component/handlers/index.ts:L7-L9`), `CmsRoutesImplService` e `isCanActivate` (`core-libs/storefront/cms-structure/services/index.ts:L7-L12`), `DEFAULT_LCP_PRESENCE` (`core-libs/storefront/shared/lcp-context/index.ts:L7-L13`), `defaultLcpCmsComponentsConfig` (`core-libs/storefront/cms-structure/cms-lcp-context/config/index.ts:L7-L7`), le factory `get...UrlMatcherFactory` (`core-libs/storefront/cms-pages/product-details-page/index.ts:L7-L8`).

Elenchi trasversali: Appendice A01 — injection token, A02 — config, A03 — componenti CMS, A05 — guard.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Nessuna config = tutti gli slot.** Senza `LayoutConfig` per il template, `resolveSlots` rende *tutti* gli slot della pagina nell'ordine delle chiavi di `page.slots`, con un `info` (elenco degli slot disponibili, una volta per template) e un `warn` (una volta per chiave: la section, o il template se manca la section) solo in dev mode, e nulla in produzione (`core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L120-L122`, `:L251-L272`). Al contrario, uno slot configurato ma assente nella pagina CMS viene scartato in silenzio (L117-L119).
2. **Section globale ignorata.** Una section definita solo globalmente (`layoutSlots.header`) si applica solo se anche il template corrente ha una voce in `layoutSlots`; altrimenti `getSlotConfigForSection` esce subito e la section rende `[]` (`core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L181-L185`, `:L123-L125`).
3. **Possibile `TypeError` nel fallback di section.** `getSlotConfigForSection` legge `pageTemplateConfig[section].hasOwnProperty(...)` senza optional chaining: se la config di section arriva dal fallback globale e la config responsive non ha l'attributo, `pageTemplateConfig[section]` è `undefined` (`core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L189-L209`).
4. **Il breakpoint "più vicino" è solo verso il basso.** `getResponsiveSlotConfig` scorre solo i breakpoint precedenti a quello corrente nell'elenco ordinato: una config `xs` vale anche per `md` se `md` non ha la propria; una config `lg` non vale mai per `md` (`core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L234-L242`).

5. **Handler di layout arrivati tardi.** Gli handler del root injector sono assegnati in modo sincrono nel costruttore (`injectors$` parte dal root injector con `startWith`); quelli forniti da moduli lazy si aggiungono solo quando il modulo viene caricato e valgono per le emissioni successive di `getSlots`. Se il root non fornisce handler, `getMulti` scarta gli array vuoti, `handlers` resta `undefined` fino al primo modulo lazy e `this.handlers || []` salta la catena (core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35, :L82-L91; core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L41-L45, :L63-L63).
6. **Page fold con le section.** `pageFoldSlot$` usa sempre il template, anche quando `PageLayoutComponent` rende una section (core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L47-L53).
7. `cx-pending` **che non sparisce.** `pending` parte dal numero totale di componenti dello slot, ma il template crea l'outlet solo per i componenti con `flexType`: un componente senza `flexType` non emetterà mai `loaded(true)` e lo slot resterà `cx-pending` (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L145-L145; core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L8-L12).
8. `loaded(false)` **anche con INSTANT.** `cxOutletDefer` riceve sempre un oggetto, quindi l'outlet passa sempre da `deferLoading()` ed emette `loaded(false)` prima di `true`, anche quando la strategia è INSTANT (core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L11-L11; core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L85-L86, :L102-L114).
9. `shouldNotDefer` **consuma lo slot.** La prima chiamata per uno slot pre-renderizzato lo rimuove dall'elenco con `splice` (core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L51-L57). Il template rivaluta `getComponentDeferOptions` a ogni change detection, ma `OutletDirective` rifà il render solo quando cambia `cxOutlet` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L92-L100): conta quindi solo la prima valutazione, che è quella che ottiene INSTANT. Attenzione però: lo slot viene consumato dal **primo componente** dello slot; dal secondo in poi `shouldNotDefer` ritorna `false` e vale il `deferLoading` del tipo. Effetto pratico su slot con più componenti differiti `DA VERIFICARE` con un test end-to-end.
10. `deferLoading` **nelle feature lazy non conta.** `getDeferLoadingStrategy` legge solo la config statica: un `deferLoading` dichiarato nella config di un feature module viene ignorato (core-libs/storefront/cms-structure/services/cms-components.service.ts:L195-L200).
11. `isDistinct` **e i campi del componente.** Lo slot riemette solo se cambia la lista degli `uid`; un cambio di altri campi dei componenti dello slot non riemette. Inoltre ritorna `false` se uno dei due slot non ha `components` (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L201-L213).
12. `replace` **sulla classe.** `decorate` rimuove la vecchia position con `cls.replace(this.lastPosition, '')`, che sostituisce la prima sottostringa uguale anche se fa parte di un'altra classe (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L136-L138).
13. **Subscription non conservata nel wrapper.** `ngOnInit` non salva la subscription a `determineMappings`: se il wrapper è distrutto prima che la feature lazy sia risolta, `launchComponent` potrebbe girare dopo il destroy (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L67-L79). `DA VERIFICARE` con un test: dipende da tempi di risoluzione della feature.
14. **Il riconoscimento del lazy si basa su** `toString()`**.** `isNotClass` controlla che i primi 20 caratteri (senza il primo spazio) inizino con `function()` o `()=>` (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L45-L48). Forme diverse (per esempio `async () =>` o una funzione con nome) non fanno match con `LazyComponentHandler` e ricadono sul `DefaultComponentHandler`, che le tratterebbe come classi (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L28-L30). Per un `() => import()` corretto fanno match entrambi e vince il lazy per priorità (`LOW` > `FALLBACK`).

15. **Web component non attivi di default.** Un mapping `'path#selector'` produce "Can't resolve handler for component mapping" finché l'app non registra `WebComponentHandler` come `ComponentHandler`; la classe non è nel public API (core-libs/storefront/cms-structure/page/component/handlers/index.ts:L7-L9; core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L54-L62). DA VERIFICARE come un'app possa importarla senza path interno. Inoltre l'elemento viene appeso al `parentElement` dell'anchor e non inserito nella view Angular, e il launcher ignora `module` (core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L49-L53, :L80-L83).
16. **Inner component senza** `uid`**.** `InnerComponentsHostDirective` usa `uid: ''`: `ComponentDataProvider.get('', type)` restituisce solo i dati statici oppure `EMPTY`, e gli eventi avranno `id: ''` e `typeCode` undefined (core-libs/storefront/cms-structure/page/component/inner-components-host.directive.ts:L73-L73; core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L55-L57; core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L128-L131).
17. **Doppia emissione di** `data$`**.** Con dati statici nel mapping, il componente vede prima i soli dati statici (`startWith`) e poi il merge (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L45-L51).
18. **Guard e** `disableSSR`**.** `getGuards` non filtra con `shouldRender` (a differenza di `getChildRoutes` e `getI18nKeys`): le guard di componenti con `disableSSR` girano anche in SSR (core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261 vs :L205-L214, :L266-L276).
19. **Guard funzionali nei mapping.** `CmsGuardsService` tiene solo oggetti con `canActivate` (`isCanActivate`), quindi una `CanActivateFn` messa in `guards` di un mapping non viene eseguita da questa strada (core-libs/storefront/cms-structure/services/cms-guards.service.ts:L42-L46): viene ignorata in silenzio. `UnifiedInjector.get` passa `NOT_FOUND_SYMBOL` come `notFoundValue` e filtra quel valore (core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67); per una funzione senza `ɵprov`, `R3Injector.get` non trova definizioni, delega al parent e alla fine restituisce il `notFoundValue` (angular:packages/core/src/di/r3_injector.ts:L351-L383), quindi `getLastValueSync` dà `undefined` e `isCanActivate` la scarta. Nelle child route, invece, le `CanActivateFn` sono supportate dal wrapper, che conta proprio su questo comportamento (`classGuard ?? { canActivate: guard as CanActivateFn }`, core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L164-L182, :L174-L179).
20. `resetConfig` **e navigazione annullata.** Al primo accesso a una content page con child route, la guard ritorna `false`, modifica la config globale del router e rinaviga (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L84-L88, :L119-L119). `wrapCmsGuardsRecursively` muta anche `route.children` dell'oggetto di config originale (L139-L141).
21. **Feature come stringa.** Le voci `featureModules` espresse come alias stringa, o senza `module`/`cmsComponents`, non entrano nella mappa tipo → feature (core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L73).
22. **Classe del template e SSR.** `DirectiveStateTransferService.set` scrive l'attributo solo lato server; nel browser la persistenza è un no-op (core-libs/storefront/utils/directive-state-transfer.service.ts:L23-L27). `currentTemplate` è un field initializer che usa `this.host` (che dipende dalle parameter properties del costruttore): funziona perché il `tsconfig.json` root imposta `useDefineForClassFields: false` (tsconfig.json:L799-L799; core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L84-L94).
23. **Marker LCP come sottostringa.** `idMarker` è cercato con `includes`, quindi ovunque nell'`uid`, non solo come suffisso (core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L55-L57).
24. `[cxProvideLcpPresence]` **come attributo nudo.** Anche un attributo statico fa scattare `ngOnChanges` (initial input: `NgOnChangesFeature` imposta `setInput = ngOnChangesSetInput`, che registra un `SimpleChange`, angular:packages/core/src/render3/features/ng_onchanges_feature.ts:L50-L53, :L87-L110), ma con `<div cxProvideLcpPresence>` il valore è la stringa vuota: `this.cxProvideLcpPresence ?? LcpPresence.NO_LCP` non la sostituisce, e la direttiva emette `''` invece di `NO_LCP` (core-libs/storefront/shared/lcp-

context/provide-lcp-presence.directive.ts:L38-L41). Va sempre passato un valore (`[cxProvideLcpPresence]="..."`).

25. **API deprecate di Angular.** `DefaultComponentHandler` usa `ComponentFactoryResolver` e l'overload factory di `createComponent`, entrambi `@deprecated` in Angular 21 (angular:packages/core/src/linker/component_factory_resolver.ts:L30-L41, angular:packages/core/src/linker/view_container_ref.ts:L263-L275): da tenere d'occhio negli upgrade.

---

10. Come fare debug

- **"Il mio slot non compare."** In dev mode cerca in console `Available CMS page slots: '...'` e `No layout config found for <template|section>` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L251-L273). Confronta i nomi con `layoutSlots`; ricorda T2 (section globale) e T4 (breakpoint solo verso il basso). Ispeziona l'attributo `position` dei `cx-page-slot` nel DOM (`@HostBinding('attr.position')`, core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L62-L69) e la classe del template sul tag `cx-page-layout` (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L147-L151).
- **"Il componente non si vede."** Messaggi utili, solo in dev mode: `No component implementation found for the CMS component type '...'` (mapping mancante, core-libs/storefront/cms-structure/services/cms-components.service.ts:L165-L181) e `Can't resolve handler for component mapping:` (formato di `component` non riconosciuto, core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L54-L62). Se in SSR manca ma nel browser c'è, verifica `disableSSR` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L187-L190).
- **Slot bloccato su** `cx-pending`. Controlla che ogni componente dello slot abbia `flexType` (T7) e se la strategia è DEFER: con `deferredLoading.strategy` o `cmsComponents[type].deferLoading = DEFER` il componente aspetta l'intersezione con il viewport (core-libs/storefront/layout/loading/defer-loader.service.ts:L55-L64).
- **Eventi dei componenti.** Abbonati a `ComponentCreateEvent`/`ComponentDestroyEvent` con `EventService.get(...)` per vedere `typeCode`, `id` e `host` di ogni componente creato (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L124-L136).
- **Mapping effettivo di una feature lazy.** Breakpoint in `CmsComponentsService.getFeatureMappingResolver` sul `deepMerge` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L129-L133) e in `CmsFeaturesService.createFeatureInstance` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L157-L183) per vedere cosa arriva dalla config della feature e cosa la sovrascrive.
- **Quale handler viene scelto.** Breakpoint in `ComponentHandlerService.resolve` (core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L49-L52) e in `resolveApplicable` (core-libs/core/src/util/applicable.ts:L63-L87).
- **Guard e child route.** Breakpoint in `CmsPageGuardService.canActivatePage` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L98): i `componentTypes` dopo `determineMappings`, l'esito di `cmsPageCanActivate` e il ramo `handleCmsRoutesInGuard`. Dopo la prima visita a una content page con child route, `router.config` contiene in testa una route con `data.cxCmsRouteContext` (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L107-L119).
- **SSR e lazy component.** Un `NG0205` (destroyed injector) in SSR con componenti lazy indica un `import()` non tracciato: il `LazyComponentHandler` usa `PendingTasks` proprio per evitarlo (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L70); un handler custom che carica codice in modo asincrono dovrebbe fare lo stesso. Vedi cap. 4 e cap. 7.

- **Classe del template dopo l'idratazione.** Nel markup SSR cerca l'attributo `data-cx-state-transfer-page-template` sul tag `cx-page-layout` (core-libs/storefront/utils/directive-state-transfer.service.ts:L23-L41).
- **LCP.** Verifica l'`uid` del componente (marker `__cxLCP__` o lista `ids`) e, nei template che usano `*cxLcpContext`, il valore di `fetchPriority$` (core-libs/storefront/shared/lcp-context/lcp-context.directive.ts:L45-L76).

---

11. Domande di autoverifica

1. **Se per `LandingPage2Template` non c'è alcuna voce in `layoutSlots`, cosa rende `PageLayoutComponent` senza section? E con `section="header"`?** Senza section: tutti gli slot della pagina, nell'ordine delle chiavi di `page.slots`, più un avviso in dev mode. Con section: `[]`, anche se esiste `layoutSlots.header` globale, perché `getSlotConfigForSection` esce subito quando il template non ha config. — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L120-L125, :L181-L185
2. **Su `md`, se la config del template ha solo `slots` piatto e una voce `xs.slots`, quale elenco viene usato?** Quello di `xs`: senza voce per `md`, `getResponsiveSlotConfig` scorre i breakpoint precedenti e tiene l'ultimo che ha `slots`; la config piatta vale solo se nessun breakpoint più piccolo ce l'ha. — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L229-L242
3. **Perché un mapping `component: () => import('./x').then(m => m.X)` non viene trattato da `DefaultComponentHandler`, che pure fa match su ogni funzione?** Fanno match entrambi, ma `LazyComponentHandler` ha priorità `LOW` (-10) e `DefaultComponentHandler` `FALLBACK` (-50); `resolveApplicable` sceglie la priorità più alta. — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L38-L52, core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L28-L34, core-libs/core/src/util/applicable.ts:L76-L86
4. **Perché `LazyComponentHandler` chiama `PendingTasks.add()`?** Per tenere l'app "instabile" finché il chunk non è caricato: altrimenti in SSR Angular serializzerebbe e distruggerebbe l'app durante l'`import()`, con errore NG0205. Il task è rimosso in `finalize`. — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L70, angular:packages/core/src/pending_tasks.ts:L42-L60
5. **Un componente vuole un servizio non singleton, una istanza per ogni componente CMS. Dove si dichiara e chi lo fornisce?** In `cmsComponents[type].providers`; `CmsInjectorService.getInjector` li aggiunge all'injector creato con `Injector.create` per ogni istanza, accanto a `CmsComponentData` e `LCP_PRESENCE`. — core-libs/core/src/cms/config/cms-config.ts:L60-L73, core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50
6. **Un feature module dichiara `cmsComponents.X.component = A` e l'app configura staticamente `cmsComponents.X.component = B`. Quale vince?** `B`: il mapping della feature è il default e viene fuso con `deepMerge({}, featureMapping, staticConfig)`, quindi la config statica sovrascrive. — core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L134
7. **Perché uno slot visibile renderizzato in SSR non applica il defer al primo render nel browser?** `PageSlotModule` istanzia subito `PageSlotService`, che nel browser legge dal DOM SSR le `position` degli slot nel viewport; `getComponentDeferOptions` ritorna `INSTANT` per quello slot (una volta sola, poi lo rimuove dall'elenco). — core-libs/storefront/cms-structure/page/slot/page-slot.module.ts:L18-L23, core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L30-L73
8. **Cosa succede alla prima navigazione verso una content page `/faq` i cui componenti dichiarano `childRoutes`?** La guard chiama `handleCmsRoutesInGuard`: `updateRouting` antepone a `router.config` una route `faq` con `PageLayoutComponent`, le child route e `data.cxCmsRouteContext`, poi rinaviga con `navigateByUrl` e ritorna `false`. Alla nuova navigazione la route ha `cxCmsRouteContext` e il passo viene sal-

tato. — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L124, core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L86-L95

9. **In che ordine vengono eseguite le guard dichiarate dai componenti della pagina, e quando si fermano?** In sequenza (`concat`): i `true` vengono saltati, il primo `false`/`UrlTree` è il risultato; se tutte ritornano `true` si emette `true` (`endWith(true)`). Senza guard: `of(true)`. — core-libs/storefront/cms-structure/services/guards-composer.ts:L39-L51, :L78-L86
10. **Come fa un componente CMS a sapere che contiene l'elemento LCP, senza codice specifico?** L'injector per componente fornisce `LCP_PRESENCE` calcolato da `CmsLcpService` sull'`uid` (marker `__cxLCP__` o lista `lcpCmsComponents.ids`); nel template, `*cxLcpContext` lo trasforma in `fetchPriority$` (`HIGH` se `HAS_LCP`). — core-libs/storefront/cms-structure/cms-lcp-context/provide-lcp-presence-for-cms-component.ts:L20-L36, core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L50-L65, core-libs/storefront/shared/lcp-context/lcp-presence-mapping.service.ts:L14-L22

TAVOLA 16.1 **4.1 Pipeline completa: dalla route al componente**

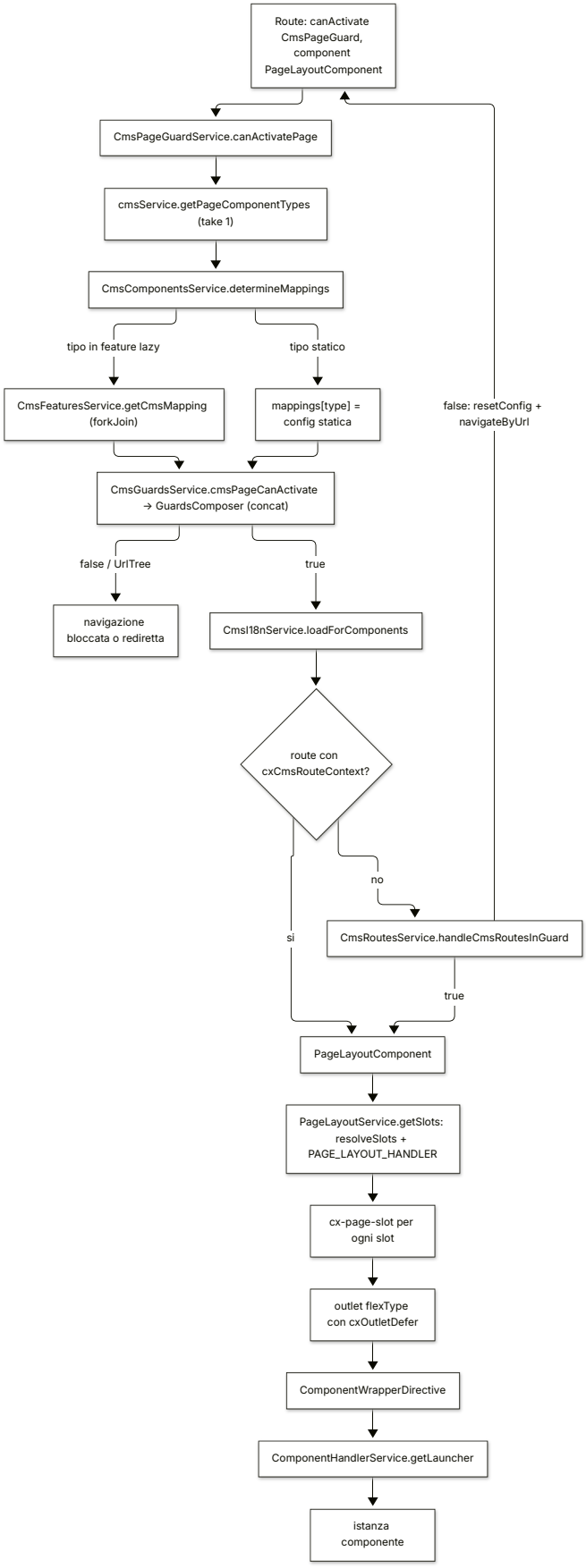


TAVOLA 16.2 **4.2 Lancio di un componente (ComponentWrapperDirective)**

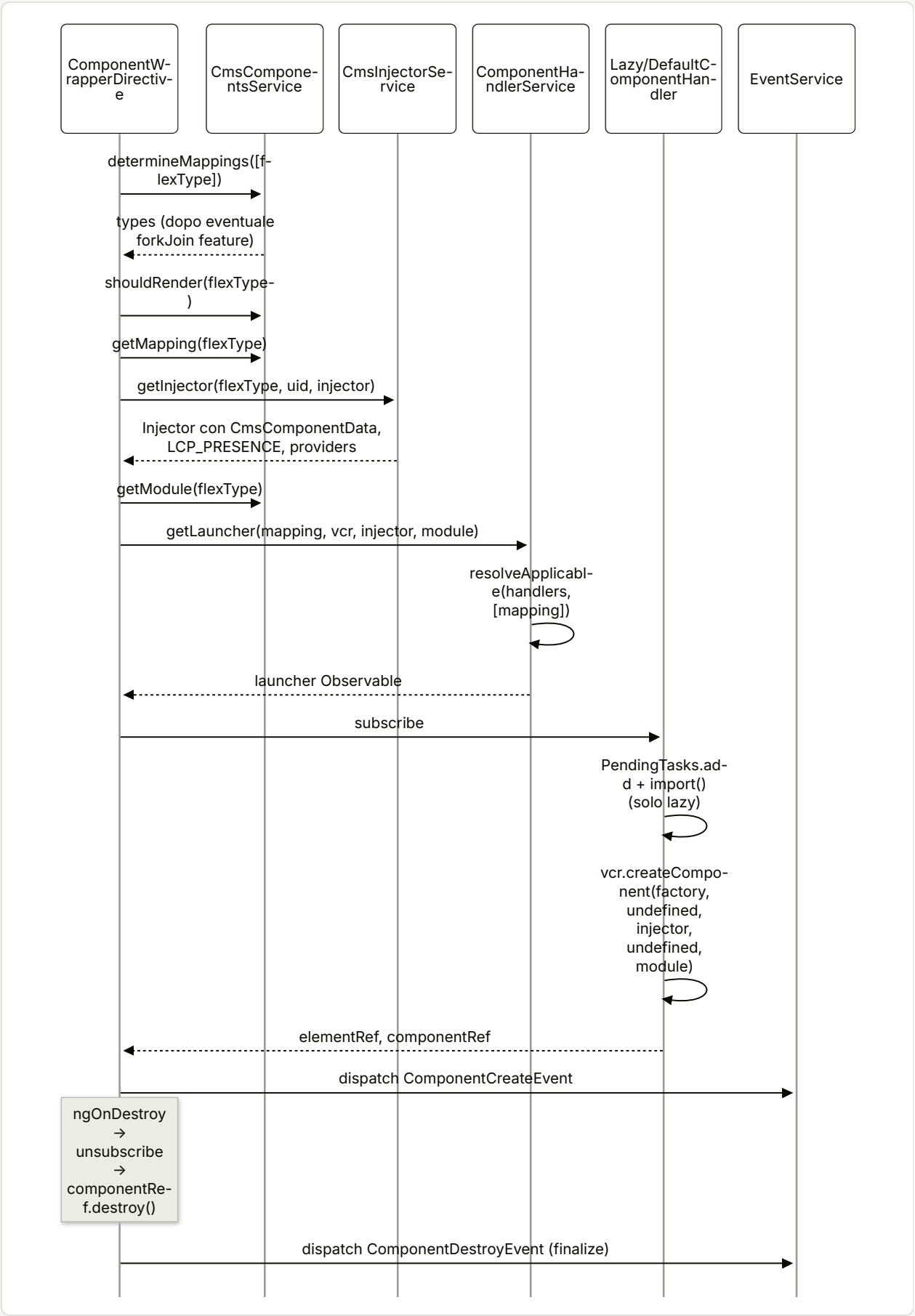
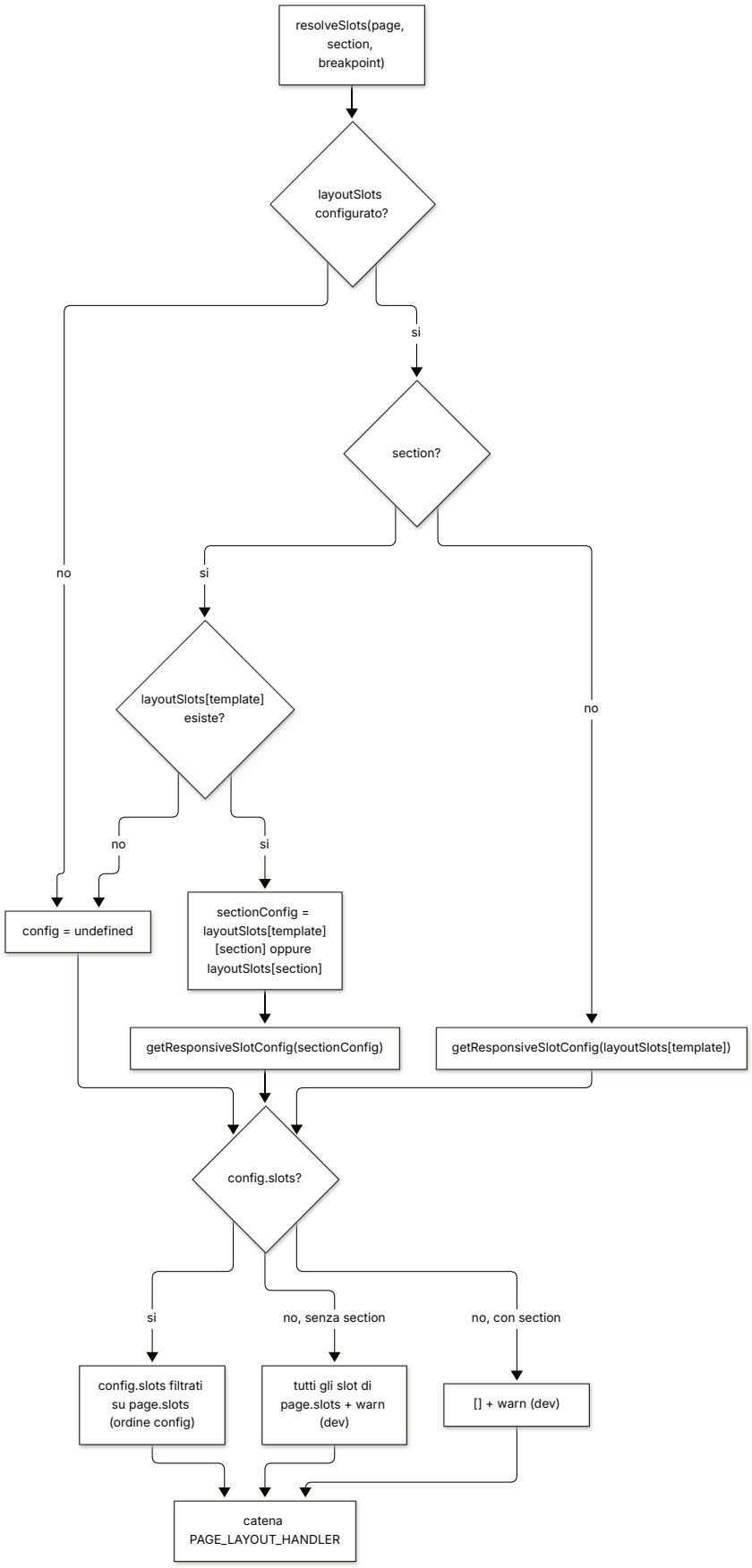


TAVOLA 16.3 **4.3 Risoluzione degli slot (resolveSlots / getSlotConfig)**



17

Outlet

Punti nominati nei template per inserire o sostituire UI senza fork.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 01 — Dependency injection, 12 — Feature module e lazy loading, 16 — CMS (storefront): rendering dinamico. Per l'elenco completo dei nomi di outlet esistenti vedi l'Appendice A08 — Outlet.

1. A cosa serve e perché esiste

Il CMS decide *quali* componenti compaiono in una pagina (cap. 15 e 16), ma molte porzioni di UI non sono componenti CMS: il riepilogo d'ordine dentro il checkout, le righe di una tabella B2B, l'intero layout di una pagina, il guscio dello storefront. Per renderle personalizzabili senza fare fork dei template, Spartacus offre gli **outlet**: punti nominati nei template in cui l'applicazione può inserire contenuto *prima*, *dopo* o *al posto* del contenuto di default.

Il meccanismo, che vive in `core-libs/storefront/cms-structure/outlet/`, ha quattro pezzi:

- un **modello**: l'enum `OutletPosition` con i valori `replace`, `before`, `after` e il token astratto `OutletContextData` con cui un componente inserito legge il contesto dell'outlet (`core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L13`, `core-libs/storefront/cms-structure/outlet/outlet.model.ts:L18-L21`);
- un **registro** globale e sincrono, `OutletService`, che per ogni posizione mantiene una `Map` nome-outlet → array di `TemplateRef` o `ComponentFactory` (`core-libs/storefront/cms-structure/outlet/outlet.service.ts:L10-L18`);
- due **direttive**: `[cxOutlet]` dichiara il punto di inserimento e renderizza ciò che il registro contiene (`core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L33-L34`); `[cxOutletRef]` registra un template HTML in un outlet (`core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L17-L22`);
- un modo **programmatico** di registrare componenti: `provideOutlet()` + `OutletModule.forRoot()/forChild()` (`core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L34-L47`, `core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88`), più un servizio per forzare il ri-render di un outlet già a schermo, `OutletRendererService` (`core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L13-L17`).

La direttiva si appoggia inoltre al **deferred loading** del layout (`DeferLoaderService`, `IntersectionService`) per costruire il contenuto solo quando l'host entra nel viewport (`core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L23-L24`, `core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L102-L116`).

Gli outlet sono ovunque nel layout di base: l'intero storefront è avvolto nell'outlet `cx-storefront` (`core-libs/storefront/layout/main/storefront.component.html:L1-L1`), ogni pagina nell'outlet che porta il nome del suo template di layout (`core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L1-L10`), ogni slot nell'outlet col nome della posizione e ogni componente CMS nell'outlet col suo `flexType` (`core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L1-L17`). È per questo che un'applicazione può sostituire "la pagina con layout X" o "il componente CMS di tipo Y" senza toccare la configurazione CMS.

2. Mappa dei file

PATH	RUOLO
core-libs/storefront/cms-structure/outlet/index.ts	Barrel pubblico: export completi di direttive, model, servizi; da outlet.module solo OutletModule, da outlet.providers solo provideOutlet e ProvideOutletOptions (:L7-L14).
core-libs/storefront/cms-structure/outlet/outlet.model.ts	OutletPosition, costanti AVOID_STACKED_OUTLETS/USE_STACKED_OUTLETS, token astratto OutletContextData (:L9-L44).
core-libs/storefront/cms-structure/outlet/outlet.service.ts	OutletService: registro root, add/get/remove/removeValueOrAll (:L10-L98).
core-libs/storefront/cms-structure/outlet/outlet.directive.ts	OutletDirective ([cxOutlet]): rendering BEFORE/REPLACE/AFTER, contesto, defer, injector con OutletContextData (:L33-L236).
core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts	OutletRendererService: mappa nome → direttiva per il re-render imperativo (:L13-L48).
core-libs/storefront/cms-structure/outlet/outlet.providers.ts	PROVIDE_OUTLET_OPTIONS (privato), ProvideOutletOptions, provideOutlet() (:L10-L47).
core-libs/storefront/cms-structure/outlet/outlet.module.ts	OutletModule con forRoot() (APP_INITIALIZER) e forChild() (MODULE_INITIALIZER); registerOutletsFactory (privato) (:L26-L88).
core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts	OutletRefDirective ([cxOutletRef], cxOutletPos) (:L17-L36).
core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.module.ts	OutletRefModule: importa/esporta la direttiva standalone (:L12-L16).

File di altri capitoli toccati dai flussi (solo riferimento):

PATH	RUOLO NEL FLUSSO
core-libs/storefront/layout/loading/defer-loader.service.ts	Decide se caricare subito o attendere l'intersezione (:L44-L64).
core-libs/storefront/layout/loading/intersection.service.ts	IntersectionObserver nativo, primo true (:L45-L58, :L103-L118).
core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts	Apre dialog/componenti dentro un outlet (:L26-L99). Dettagli nel cap. 18.
core-libs/storefront/cms-structure/page/slot/page-slot.component.html	Consumer principale con cxOutletDefer e (loaded) (:L1-L17).
core-libs/core/src/lazy-loading/tokens.ts, lazy-modules.service.ts, lazy-loading.module.ts	MODULE_INITIALIZER usato da OutletModule.forChild() (cap. 12).

3. Come funziona passo per passo

3.1 Il modello: posizioni e contesto

`OutletPosition` è un enum di stringhe: `REPLACE = 'replace'`, `BEFORE = 'before'`, `AFTER = 'after'` (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L13). Essendo valori stringa, nei template si possono scrivere letteralmente (`cxOutletPos="before"`, come nell'app demo: projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-template.component.html:L3-L3).

Due costanti booleane danno un nome al terzo parametro `stacked` di `OutletService.get`: `AVOID_STACKED_OUTLETS = false` e `USE_STACKED_OUTLETS = true` (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L15-L16).

`OutletContextData<T>` è una **classe astratta usata come token DI** (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L18-L21). Espone `reference` (nome dell'outlet), `position`, `context` e `context$: Observable<T>` (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L25-L43). `context` è il solo valore iniziale, non viene aggiornato ed è `@deprecated since 3.0` a favore di `context$` (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L39).

3.2 Il registro `OutletService`

`OutletService` è `providedIn: 'root'` ed è generico su `T = TemplateRef<any> | ComponentFactory<any>` (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L10-L13). Il suo stato è un oggetto con tre `Map<string, T[]>`, una per posizione (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L14-L18).

1. `add(outlet, templateOrFactory, position = REPLACE)` prende lo store della posizione e accoda il nuovo elemento all'array esistente con `concat`, senza deduplicare (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L42). Se la posizione non corrisponde a nessuno store, la chiamata è ignorata in silenzio (guard `if (store)`, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L36-L41).
2. `get(outlet, position = REPLACE, stacked = AVOID_STACKED_OUTLETS)` ricade sullo store REPLACE se la posizione non esiste (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L53-L60); con `stacked=false` restituisce solo il **primo** elemento, con `stacked=true` l'intero array (o `undefined`) (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L62-L66).
3. `remove(outlet, position = REPLACE, value?)` ricade anch'esso su REPLACE e delega a `removeValueOrAll` (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L69-L79), che è `protected`: senza `value` cancella l'intera chiave, con `value` filtra per identità (`!==`) (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L81-L96).

Il registro **non espone Observable**: aggiungere o togliere una voce non notifica nessuna direttiva (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L13-L98). Questa scelta è all'origine di diverse trappole (sezione 9) e dell'esistenza di `OutletRendererService` (3.7).

3.3 Tre modi di riempire il registro

(a) `provideOutlet` **+** `OutletModule.forRoot()` **(eager).**

1. Un modulo dichiara `provideOutlet({ id, component, position? })`, che restituisce un multi provider `{ provide: PROVIDE_OUTLET_OPTIONS, useValue: options, multi: true }` (core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L47). `ProvideOutletOptions` ha `id: string`, `component:`

Type<any>, position?: OutletPosition (core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L17-L32).

2. BaseStorefrontModule importa OutletModule.forRoot() (core-libs/storefront/base-storefront.module.ts:L43-L43), che registra registerOutletsFactory come APP_INITIALIZER multi con deps [new Optional(), PROVIDE_OUTLET_OPTIONS], ComponentFactoryResolver, OutletService (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L70).
3. Al bootstrap la funzione restituita da registerOutletsFactory itera sulle opzioni, risolve una ComponentFactory per ogni component e la aggiunge a OutletService con posizione options.position ?? OutletPosition.AFTER (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L47).
4. In Angular, resolveComponentFactory crea **una nuova istanza** di ComponentFactory a ogni chiamata, legata all'NgModuleRef del resolver (angular:packages/core/src/render3/component_ref.ts:L97-L101); con bootstrapApplication (come nell'app demo, projects/storefrontapp/src/main.ts:L17-L17) il root environment injector è un EnvironmentNgModuleRefAdapter, che fornisce un proprio ComponentFactoryResolver(this) (angular:packages/core/src/application/create_application.ts:L72-L72, angular:packages/core/src/render3/ng_module_ref.ts:L147-L148, angular:packages/core/src/render3/ng_module_ref.ts:L162-L162); per i moduli lazy è invece l'NgModuleRef del modulo (angular:packages/core/src/render3/ng_module_ref.ts:L64-L65, angular:packages/core/src/render3/ng_module_ref.ts:L88-L89).

(b) provideOutlet **+** OutletModule.forChild() **(moduli lazy).**

1. forChild() registra la stessa factory ma sul token MODULE_INITIALIZER di @spartacus/core (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88). È importato in CartBaseComponentsModule, PickupInStoreComponentsModule, OrderConfirmationModule (feature-libs/cart/base/components/cart-base-components.module.ts:L30-L30, feature-libs/pickup-in-store/components/pickup-in-store-components.module.ts:L28-L28, feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L61-L61).
2. MODULE_INITIALIZER è descritto come "un APP_INITIALIZER per moduli lazy"; se il modulo è eager le funzioni girano all'avvio (core-libs/core/src/lazy-loading/tokens.ts:L9-L20). Il fallback eager è l'APP_INITIALIZER di LazyLoadingModule.forRoot() (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L40-L52), importato da BaseCoreModule (core-libs/core/src/base-core.module.ts:L41-L41).
3. Per un modulo lazy, LazyModulesService.resolveModuleInstance crea il modulo e subito dopo chiama runModuleInitializersForModule(moduleRef) (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L98), che legge MODULE_INITIALIZER dall'injector del modulo con { self: true } (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157) ed esegue tutte le funzioni, attendendo solo i risultati promise-like (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L187-L199). La funzione restituita da registerOutletsFactory ha tipo () => void: non restituisce una promise, quindi runModuleInitializersForModule non attende nulla (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L33-L33, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L194-L197).
4. Le deps vengono risolte dall'injector del modulo lazy: il ComponentFactoryResolver è quello dell'NgModuleRef lazy, quindi la factory è legata a quel modulo (angular:packages/core/src/render3/ng_module_ref.ts:L64-L65, angular:packages/core/src/render3/ng_module_ref.ts:L88-L89).

Esempio concreto nell'app demo: CartBaseModule importa CartBaseComponentsModule (feature-libs/cart/base/cart-base.module.ts:L12-L15), che contiene OutletModule.forChild(), ed è caricato lazy da featureModules[CART_BASE_FEATURE] tramite CartBaseWrapperModule (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L26-L35, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L22-L25).

(c) [cxOutletRef] **da template.**

1. `<ng-template [cxOutletRef]="nome" cxOutletPos="before">` istanzia `OutletRefDirective`, con input `cxOutletRef: string` e `cxOutletPos: OutletPosition` (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L17-L22).
2. In `ngOnInit` registra il proprio `TemplateRef` con `outletService.add(cxOutletRef, tpl, cxOutletPos);` con `cxOutletPos` assente vale il default REPLACE di `add()` (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L29-L31, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L35).
3. In `ngOnDestroy` rimuove esattamente il proprio template (filtro per identità) (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L33-L35, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L88-L95).

L'app demo usa questa forma solo in `TestOutletModule`, a scopo di test e2e (projects/storefrontapp/src/app/private/private.providers.ts:L52-L53).

C'è poi un quarto "scrittore" del registro, `OutletRenderStrategy` del launch dialog, descritto in 3.7.

3.4 Rendering di `[cxOutlet]`

`OutletDirective` ha selettore `[cxOutlet]`, nessun flag `standalone` (quindi standalone per default in Angular 21) e implementa `OnDestroy, OnChanges` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L33-L34). Inietta `ViewContainerRef`, `TemplateRef`, `OutletService`, `DeferLoaderService`, `OutletRendererService` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L67-L73): l'iniezione di `TemplateRef` impone di applicarla a un `<ng-template>` (o in forma strutturale con `*`).

Input e output:

- `cxOutlet: string` e `cxOutletContext: T` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L41-L46);
- `cxOutletDefer: IntersectionOptions` e `loaded: EventEmitter<boolean>` creato con `isAsync = true` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L53-L58);
- la coppia `cxComponentRef` / `cxComponentRefChange` per il binding two-way `[(cxComponentRef)]` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L60-L63).

Stato interno: `renderedTemplate` (privato, elementi già creati) e `renderedComponents` (pubblico, `Map` posizione → array di `ComponentRef`/`EmbeddedViewRef`) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L35-L39); il contesto corrente vive in un `ReplaySubject<T>(1)` privato `outletContext$` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L48-L51).

Il ciclo è:

1. `ngOnChanges`: se cambia `cxOutlet` esegue `render()` e poi registra la direttiva in `OutletRendererService`; se cambia `cxOutletContext` emette il nuovo valore su `outletContext$` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L92-L100). Poiché il primo `ngOnChanges` contiene sia `cxOutlet` sia `cxOutletContext`, `render()` gira prima che `outletContext$` riceva il primo valore; dato che è un `ReplaySubject(1)`, i componenti sottoscritti a `context$` ricevono comunque il valore appena emesso (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L51-L51).
2. `render()` (pubblico): `vcr.clear()`, azzera `renderedTemplate` e `renderedComponents`, ricrea la `Subscription`, poi sceglie `deferLoading()` se `cxOutletDefer` è truthy, altrimenti `build()` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L78-L90).
3. `build()` costruisce le posizioni in ordine BEFORE, REPLACE, AFTER (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L121-L125). Poiché tutte usano lo stesso `ViewContainerRef` e `createComponent`/`createEmbeddedView` sono chiamati senza indice (inserimento in coda:

angular:packages/core/src/linker/view_container_ref.ts:L682-L685), l'ordine di creazione è anche l'ordine nel DOM.

4. `buildOutlet(position)`:
   - legge il registro con `USE_STACKED_OUTLETS` (tutti gli elementi) e scarta quelli già in `renderedTemplate` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L130-L135);
   - se per REPLACE il risultato è `undefined`, usa il **contenuto di default**, cioè il `TemplateRef` dell'`<ng-template>` su cui sta la direttiva (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L137-L139);
   - un risultato non-array (es. `OutletService` esteso) viene avvolto in un array (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L141-L145);
   - chiama `create()` per ciascun elemento e salva i ref non-undefined in `renderedComponents` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L147-L155).
5. `create(obj, position)` aggiunge sempre `obj` a `renderedTemplate` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L165-L165), poi:
   - `ComponentFactory` → `vcr.createComponent(factory, undefined, getComponentInjector(position))` ed emette `cxComponentRefChange` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L167-L174);
   - `TemplateRef` → `vcr.createEmbeddedView(tpl, { $implicit: this.cxOutletContext })`, `markForCheck()` sulla vista ed emissione di `cxComponentRefChange` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L175-L188);
   - qualsiasi altro oggetto non produce nulla: `create()` ritorna implicitamente `undefined` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L167-L190).

3.5 Contesto: `$implicit` per i template, `OutletContextData` per i componenti

Le due forme di contenuto ricevono il contesto in modo diverso:

- un **template** lo riceve come `$implicit` della vista embedded (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L176-L181) e lo legge con `let-model` (projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-template.component.html:L3-L19);
- un **componente** riceve un injector dedicato: `getComponentInjector()` costruisce un `OutletContextData` con `reference`, `position`, `context` (valore corrente) e `context$` (dal `ReplaySubject`) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L196-L202) e lo fornisce via `useValue` in un `Injector.create` con parent `this.vcr.injector` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L204-L212).

Un componente tipico usa `@Optional() OutletContextData` e si sottoscrive a `context$` (feature-libs/cart/base/components/cart-shared/order-summary/order-summary.component.ts:L26-L34); l'iniezione è opzionale perché lo stesso `OrderSummaryComponent` è usato anche come elemento diretto `<cx-order-summary [cart]="cart" />` (feature-libs/cart/base/components/cart-totals/cart-totals.component.html:L2-L2).

Lato Angular, la `ComponentFactory` passata a `createComponent` porta con sé un `ngModule`: in quel caso `ViewContainerRef.createComponent` non cerca un `EnvironmentInjector` nell'albero DI e usa come `contextInjector` l'injector passato (angular:packages/core/src/linker/view_container_ref.ts:L531-L537); l'environment injector diventa l'`ngModule` della factory (angular:packages/core/src/render3/component_ref.ts:L282-L286). Il componente vede quindi: node injector = quello con `OutletContextData` (parent `vcr.injector`), environment injector = il `NgModuleRef` in cui la factory è stata risolta (per `forRoot`, con `bootstrapApplication`, l'`EnvironmentNgModuleRefAdapter` root; per `forChild` l'`NgModuleRef` lazy).

3.6 Deferred loading (`cxOutletDefer`)

Quando `cxOutletDefer` è valorizzato, `render()` chiama `deferLoading()` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L85-L86):

1. emette `loaded(false)` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L103-L103);
2. ricava l'host risalendo i `parentNode` dal nodo del VCR (per un `ng-template` è un commento) fino al primo `HTMLElement` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L104-L104, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L225-L230);
3. si sottoscrive a `deferLoaderService.load(hostElement, cxOutletDefer)` e alla prima emissione esegue `build()` ed emette `loaded(true)` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L108-L115).

`DeferLoaderService.load` restituisce `of(true)` se deve caricare subito, altrimenti `intersectionService.isIntersected(element, options)` (core-libs/storefront/layout/loading/defer-loader.service.ts:L44-L53). Carica subito se: siamo su server, oppure la strategia dell'elemento è INSTANT, oppure non è DEFER e quella globale è INSTANT (core-libs/storefront/layout/loading/defer-loader.service.ts:L55-L64). La strategia globale è `config.deferredLoading?.strategy ?? DeferLoadingStrategy.INSTANT` (core-libs/storefront/layout/loading/defer-loader.service.ts:L30-L31); `DeferLoadingStrategy` vale `DEFER = 'DEFERRED-LOADING'` o `INSTANT = 'INSTANT-LOADING'` (core-libs/core/src/cms/config/cms-config.ts:L96-L101).

`IntersectionService.isIntersected` su server restituisce `of(false)`, in browser emette solo il primo `true` (core-libs/storefront/layout/loading/intersection.service.ts:L45-L58), osservando l'elemento con un `IntersectionObserver` nativo disconnesso al teardown (core-libs/storefront/layout/loading/intersection.service.ts:L103-L118); il `rootMargin` viene da `options.rootMargin` o da `LayoutConfig.deferredLoading.intersectionMargin` (core-libs/storefront/layout/loading/intersection.service.ts:L120-L130).

Il consumer principale è `PageSlotComponent`: per ogni componente CMS crea un outlet `component.flexType` con `[cxOutletDefer]="getComponentDeferOptions(component.flexType)"` e `(loaded)="isLoaded($event)"` (core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L7-L13). `getComponentDeferOptions` restituisce **sempre** un oggetto (INSTANT per slot da non differire, altrimenti la strategia del tipo di componente), quindi il ramo defer è sempre preso (core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L63-L73). Il contatore `pending` parte dal numero di componenti dello slot e scende a ogni `loaded(true)`; a zero la classe host `cx-pending` sparisce (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L145-L145, core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L170-L173, core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L183-L188, core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L85-L85). Il resto del deferred loading è trattato nel cap. 18.

3.7 Re-render imperativo: `OutletRendererService` e launch dialog

`OutletRendererService` (`providedIn: 'root'`) tiene un `BehaviorSubject` di `Map<string, OutletDirective>` (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L13-L17):

- `register(cxOutlet, directive)` muta la Map con `set` e ri-emette la stessa istanza (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L36-L38); è chiamato da ogni `OutletDirective` dopo `render()` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L95-L95);
- `render(outlet)` chiama `render()` sulla direttiva registrata con quel nome, se esiste (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L28);

- `getOutletRef(outlet)` restituisce un Observable della direttiva, filtrando i null con `isNotNullable` (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L43-L48).

Il consumer principale è `OutletRenderStrategy` (launch dialog), che inietta `OutletService<ComponentFactory<any>>`, `ComponentFactoryResolver` e `OutletRendererService` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L26-L36). Nel suo `render`:

1. risolve la factory di `config.component` e la aggiunge all'outlet con posizione `config.position` o, in mancanza, BEFORE (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L49-L57), coerentemente col default documentato in `LaunchOutletDialog` (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L38-L47);
2. forza il ri-render dell'outlet con `outletRendererService.render(config.outlet)` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L58-L58) — necessario perché il registro non è reattivo;
3. ottiene il `ComponentRef` creato leggendo `renderedComponents` della direttiva e cercando, dall'ultimo, quello con lo stesso `componentType` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L61-L72);
4. alla chiusura, `remove` risolve una nuova factory e la passa a `outletService.remove` come `value` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L86-L97).

Esempio reale: ASM si apre nell'outlet `'cx-storefront'` senza `position`, quindi prima dello storefront (feature-libs/asm/components/default-asm-layout.config.ts:L10-L17, core-libs/storefront/layout/main/storefront.component.html:L1-L1).

3.8 Teardown

`OutletDirective.ngOnDestroy` chiude la subscription del defer e completa `outletContext$`, ma non deregistra la direttiva da `OutletRendererService` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L232-L235). `OutletRefDirective.ngOnDestroy` invece rimuove il proprio template dal registro (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L33-L35). Le registrazioni fatte via `provideOutlet` non vengono mai rimosse: nessun codice in `outlet.module.ts` chiama `remove` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L47).

---

4. Diagrammi

4.1 Chi scrive nel registro

TAVOLA 17.1 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: `provideOutlet` (core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L47), `forRoot`/`forChild` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88), factory e default AFTER (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L47), `self: true` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157), `cxOutletRef` (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L29-L35), launch dialog (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L49-L58).

4.2 Rendering di un outlet (con e senza defer)

■ **TAVOLA 17.2** diagramma a tutta pagina, in fondo al capitolo

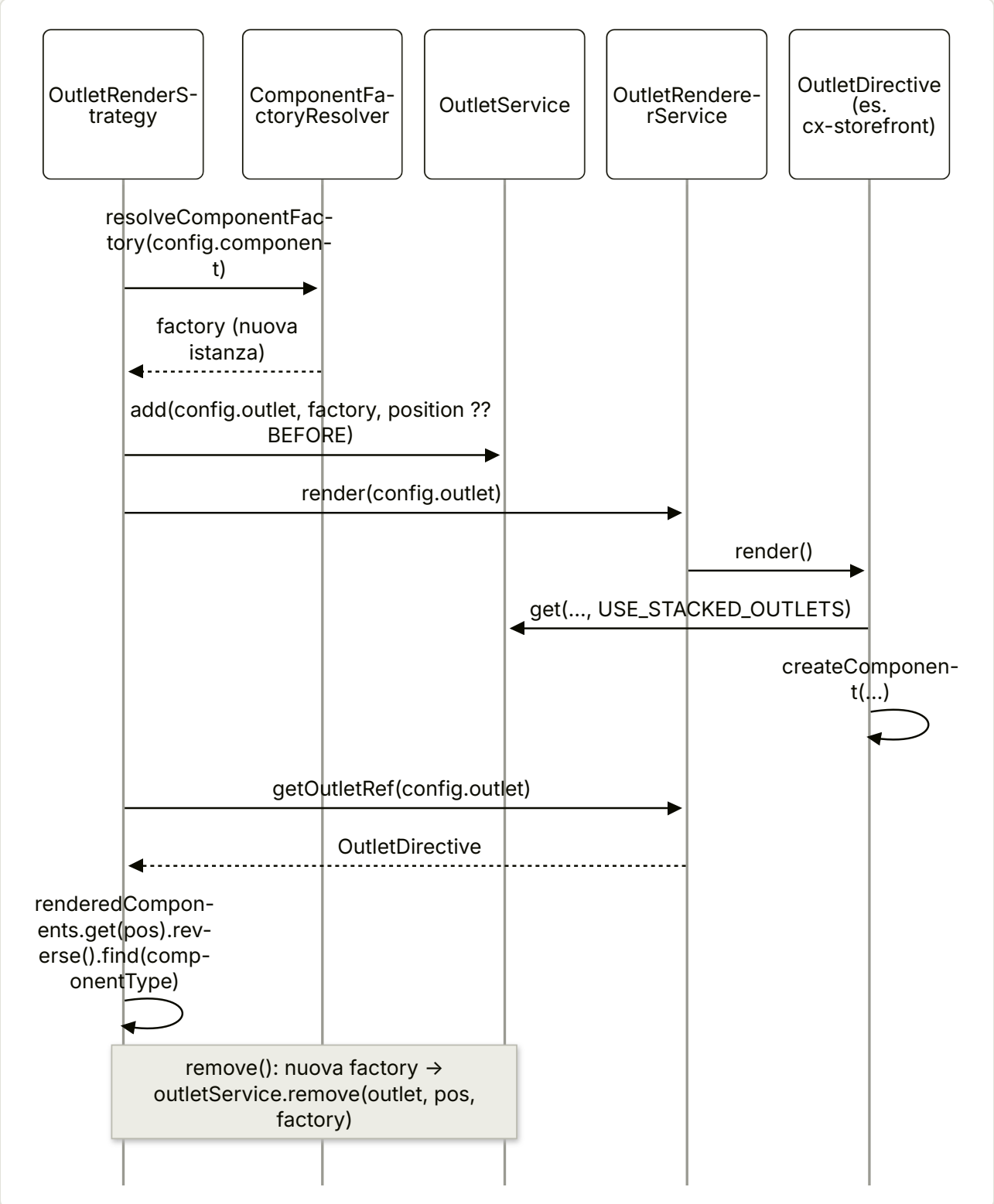
Riferimenti: `ngOnChanges` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L92-L100), `render` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L78-L90), `deferLoading` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L102-L116; nel ramo DEFER `build()` gira nella subscribe, quindi dopo `register`: core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L93-L96, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L108-L115), `DeferLoaderService` (core-libs/storefront/layout/loading/defer-loader.service.ts:L44-L64), `IntersectionService` (core-libs/storefront/layout/loading/intersection.service.ts:L45-L58), `buildOutlet`/`create` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L130-L190).

4.3 Decisione di `buildOutlet` per una posizione

■ **TAVOLA 17.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L130-L156, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L161-L190.

4.4 Launch dialog in un outlet



Riferimenti: `core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L45-L99`, `core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L48`.

5. Estratti di codice reale

Il registro: `add` **e** `get`**.**

TS

```ts
// core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L66
  add(
    outlet: string,
    templateOrFactory: T,
    position: OutletPosition = OutletPosition.REPLACE
  ): void {
    const store = this.templatesRefs[position];
    if (store) {
      const existing = store.get(outlet) || [];
      const newValue: T[] = existing.concat([templateOrFactory]);
      store.set(outlet, newValue);
    }
  }
  // ...
  get(
    outlet: string,
    position: OutletPosition = OutletPosition.REPLACE,
    stacked = AVOID_STACKED_OUTLETS
  ): T[] | T | undefined {
    const store =
      this.templatesRefs[position] ||
      this.templatesRefs[OutletPosition.REPLACE];

    const templateRef: T[] | undefined = store.get(outlet);
    if (templateRef && !stacked) {
      return templateRef[0];
    }
    return templateRef;
  }
```

Il cuore della direttiva: `render`**,** `ngOnChanges`**,** `buildOutlet`**.**

TS

```ts
// core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L78-L100
  public render(): void {
    this.vcr.clear();
    this.renderedTemplate = [];
    this.renderedComponents.clear();
    this.subscription.unsubscribe();
    this.subscription = new Subscription();

    if (this.cxOutletDefer) {
      this.deferLoading();
    } else {
      this.build();
```

```
    }
  }

  ngOnChanges(changes: SimpleChanges): void {
    if (changes.cxOutlet) {
      this.render();
      this.outletRendererService.register(this.cxOutlet, this);
    }
    if (changes.cxOutletContext) {
      this.outletContext$.next(this.cxOutletContext);
    }
  }
```

TS

```
// core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L130-L156
  private buildOutlet(position: OutletPosition): void {
    let templates: any[] = <any[]>(
      this.outletService.get(this.cxOutlet, position, USE_STACKED_OUTLETS)
    );

    templates = templates?.filter((el) => !this.renderedTemplate.includes(el));

    if (!templates && position === OutletPosition.REPLACE) {
      templates = [this.templateRef];
    }

    // Just in case someone extended the `OutletService` and
    // returns a singular object.
    if (!Array.isArray(templates)) {
      templates = [templates];
    }

    const components: (ComponentRef<any> | EmbeddedViewRef<any>)[] = [];
    templates.forEach((obj) => {
      const component = this.create(obj, position);
      if (component) {
        components.push(component);
      }
    });

    this.renderedComponents.set(position, components);
  }
```

L'injector che porta `OutletContextData`**.**

TS

```
// core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L196-L213
  private getComponentInjector(position: OutletPosition): Injector {
    const contextData: OutletContextData<T> = {
```

```
      reference: this.cxOutlet,
      position,
      context: this.cxOutletContext,
      context$: this.outletContext$.asObservable(),
    };

    return Injector.create({
      providers: [
        {
          provide: OutletContextData,
          useValue: contextData,
        },
      ],
      parent: this.vcr.injector,
    });
  }
```

Registrazione programmatica: `provideOutlet` **e** `registerOutletsFactory`**.**

TS

```
// core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L34-L47
/**
 * Helper function to register a component for an outlet.
 *
 * @param options.id unique id of the outlet
 * @param options.component Component to be registered for the outlet
 * @param options.position Component's position in the outlet (default: `OutletPosition.AFTER`)
 */
export function provideOutlet(options: ProvideOutletOptions): StaticProvider {
  return {
    provide: PROVIDE_OUTLET_OPTIONS,
    useValue: options,
    multi: true,
  };
}
```

TS

```
// core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L47
export function registerOutletsFactory(
  providedOutletOptions: ProvideOutletOptions[],
  componentFactoryResolver: ComponentFactoryResolver,
  outletService: OutletService<ComponentFactory<Type<any>>>
): () => void {
  const result = () => {
    (providedOutletOptions ?? []).forEach((options) => {
      const factory = componentFactoryResolver.resolveComponentFactory(
        options.component
      );
      outletService.add(
```

```
        options.id,
        factory,
        options.position ?? OutletPosition.AFTER
      );
    });
  };
  return result;
}
```

Uso reale: registrazione e "ancora vuota". Il cart registra `OrderSummaryComponent` in AFTER (default):

```ts
// feature-libs/cart/base/components/cart-shared/cart-shared.module.ts:L55-L58
    provideOutlet({
      id: CartOutlets.ORDER_SUMMARY,
      component: OrderSummaryComponent,
    }),
```

Il checkout dichiara l'outlet su un `<ng-template />` vuoto: il REPLACE di default non produce nulla e il componente registrato compare dopo (feature-libs/checkout/base/components/checkout-order-summary/checkout-order-summary.component.html:L1-L4):

```html
<!-- feature-libs/checkout/base/components/checkout-order-summary/checkout-order-summary.component.html:L1-L4 -->
<ng-template
  [cxOutlet]="cartOutlets.ORDER_SUMMARY"
  [cxOutletContext]="cart$ | async"
/>
```

E il componente legge il contesto:

```ts
// feature-libs/cart/base/components/cart-shared/order-summary/order-summary.component.ts:L26-L34
  constructor(@Optional() protected outlet?: OutletContextData<any>) {}

  ngOnInit(): void {
    if (this.outlet?.context$) {
      this.subscription.add(
        this.outlet.context$.subscribe((context) => (this.cart = context))
      );
    }
  }
```

Outlet annidati con defer in `PageSlotComponent`.

```html
<!-- core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L1-L17 -->
<ng-template
  *ngIf="position"
  [cxOutlet]="position"
```

```
  [cxOutletContext]="{ components$: components$ }"
>
  <ng-container *ngFor="let component of components">
    <ng-template
      *ngIf="component.flexType"
      [cxOutlet]="component.flexType"
      [cxOutletContext]="{ component: component }"
      [cxOutletDefer]="getComponentDeferOptions(component.flexType)"
      (loaded)="isLoaded($event)"
    >
      <ng-container [cxComponentWrapper]="component" />
    </ng-template>
  </ng-container>
</ng-template>
```

`cxOutletRef` **nell'app demo** (posizione `before`, contesto via `let-model`; nel file seguono `after` e REPLACE senza `cxOutletPos`, `projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-template.component.html:L11-L11` e `projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-template.component.html:L19-L19`):

HTML

```
<!-- projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-
template.component.html:L3-L9 -->
<ng-template [cxOutletRef]="testTemplate" cxOutletPos="before" let-model>
  <p>
    Before slot 1, slots: {{ (model.slots$ | async)?.length }}, template:
    {{ model.templateName$ | async }}, sections:
    {{ model.sections$ | async | json }}
  </p>
</ng-template>
```

6. Configurazione

Gli outlet non hanno una chiave di configurazione globale propria: `PROVIDE_OUTLET_OPTIONS` è un token marcato `@private` con il commento "We plan to drive the outlets by standard configuration" (`core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L10-L15`). I "parametri" sono le opzioni di `provideOutlet`, gli input delle direttive e la config di deferred loading.

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`provideOutlet({ id })`	`string`	— (obbligatoria)	`core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L21-L21`	`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L39-L40`
`provideOutlet({ component })`	`Type<any>`	— (obbligatoria)	`core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L26-L26`	`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L36-L38`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
provideOutlet({ position })	OutletPosition?	AFTER	core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L31-L31	core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42
[cxOutlet]	string	—	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L41-L41	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L93-L95, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L132-L132, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L198-L198
[cxOutletContext]	T	undefined	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L46-L46	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L97-L99, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L179-L179, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L200-L200
[cxOutletDefer]	IntersectionOptions	undefined (nessun defer)	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L56-L56	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L85-L85, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L110-L110
[(cxComponentRef)]	ComponentRef | EmbeddedViewRef	—	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L60-L63	mai letto; solo emesso via cxComponentRefChange (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L173-L173, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L187-L187)
[cxOutletRef]	string	—	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L19-L20	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L30-L30, core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L34-L34
cxOutletPos	OutletPosition	undefined → REPLACE in add()	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L21-L22	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L30-L30; default in core-libs/storefront/cms-structure/outlet/outlet.service.ts:L34-L34
OutletService.get(..., stacked)	boolean	AVOID_STACKED_OUTLETS (false)	core-libs/storefront/cms-structure/outlet/outlet.service.ts:L56-L56	core-libs/storefront/cms-structure/outlet/outlet.service.ts:L63-L63
layout.launch[X].outlet / .position (LaunchOutletDialog)	string / OutletPosition?	position BEFORE	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L38-L47	core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L53-L57
deferredLoading.strategy (LayoutConfig)	DeferLoadingStrategy?	INSTANT	core-libs/storefront/layout/config/layout-config.ts:L114-L114	core-libs/storefront/layout/loading/defer-loader.service.ts:L30-L31

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`deferredLoading.intersectionMargin` (`LayoutConfig`)	`string?`	`undefined`	`core-libs/storefront/layout/config/layout-config.ts:L121-L121`	`core-libs/storefront/layout/loading/intersection.service.ts:L124-L129`
`IntersectionOptions.deferLoading` / `rootMargin` / `threshold`	vari	—	`core-libs/storefront/layout/loading/intersection.model.ts:L9-L21`	`core-libs/storefront/layout/loading/defer-loader.service.ts:L48-L48`, `core-libs/storefront/layout/loading/intersection.service.ts:L108-L109`, `core-libs/storefront/layout/loading/intersection.service.ts:L121-L122`

Default di posizione a confronto (fonte frequente di sorprese, vedi T1):

VIA DI REGISTRAZIONE	DEFAULT	RIFERIMENTO
`provideOutlet` (via `registerOutletsFactory`)	`AFTER`	`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42`
`OutletService.add` diretto	`REPLACE`	`core-libs/storefront/cms-structure/outlet/outlet.service.ts:L34-L34`
`[cxOutletRef]` senza `cxOutletPos`	`REPLACE`	`core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L30-L30`
Launch dialog (`OutletRenderStrategy`)	`BEFORE`	`core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L56-L56`

I nomi degli outlet sono stringhe, raccolte per convenzione in enum per feature: es. `CartOutlets.ORDER_SUMMARY = 'cx-order-summary'`, `StorefrontOutlets.STOREFRONT = 'cx-storefront'` (`feature-libs/cart/base/root/models/cart-outlets.model.ts:L18-L18`, `core-libs/storefront/layout/main/storefront-outlets.model.ts:L7-L9`). L'elenco completo (9 enum, 38 membri, 85 usi nei template, 35 chiamate `provideOutlet`) è nell'Appendice A08. Oltre agli outlet "dichiarati" in enum, sono outlet anche i nomi di layout, le posizioni di slot e i `flexType` dei componenti CMS (`core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L4-L4`, `core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L3-L3`, `core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L9-L9`).

7. Commenti e note notevoli dal sorgente

- `OutletContextData`: "Token for injecting outlet related context to the component rendered in the outlet" (`core-libs/storefront/cms-structure/outlet/outlet.model.ts:L18-L20`).
- `context`: "WARNING: Avoid using it if you need to read the latest value of the context. This property is only the initial context value and won't be updated when the new context appears." + `@deprecated since 3.0` (`core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L37`).
- `PROVIDE_OUTLET_OPTIONS`: "@private We plan to drive the outlets by standard configuration" — dichiara l'intenzione di passare a una configurazione standard; allo SHA analizzato non esiste (`core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L10-L12`).
- `registerOutletsFactory` è marcata `@private` ed è esclusa dal barrel (`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L26-L28`, `core-libs/storefront/cms-structure/outlet/index.ts:L12-L12`).

- Il JSDoc di `OutletService.get` cita `OutletPosition.before` (minuscolo), membro che non esiste nell'enum (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L50-L50).
- `deferLoading`: "Although the deferLoaderService might emit only once, as long as the hostElement isn't being loaded, there's no value being emitted. Therefore we need to clean up the subscription on destroy." (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L105-L107).
- `buildOutlet`: "Just in case someone extended the `OutletService` and returns a singular object." — tolleranza esplicita verso override del servizio (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L141-L142).
- `create`: "we do not know if content is created dynamically or not so we apply change detection anyway" (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L183-L184).
- `getHostElement`: si evita di attraversare i parent *elements* perché bloccante su ie11 (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L215-L224).
- `OutletRendererService.render`: JSDoc "Dynamically render the templates in the specified array", ma il parametro è una stringa (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L19-L24).
- `getOutletRef`: JSDoc "Returns map of outlets", ma restituisce una singola direttiva (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L39-L43).
- `TestOutletModule` è importato "Custom usages of cxOutletRef only for e2e testing" (projects/storefrontapp/src/app/private/private.providers.ts:L52-L53).
- Esiste un uso strutturale commentato `*cxOutlet` nel product summary; nessun uso attivo di quella forma (core-libs/storefront/cms-components/product/product-summary/product-summary.component.html:L26-L26).
- Nessun TODO/FIXME nei 9 file della cartella outlet (verificato allo SHA con `grep -rn "TODO\|FIXME" core-libs/storefront/cms-structure/outlet/`: nessun risultato).

8. Extension point

Tutto ciò che segue è esportato da `@spartacus/storefront` tramite la catena core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L9-L9 → core-libs/storefront/cms-structure/outlet/index.ts:L7-L14.

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`provideOutlet()` + `ProvideOutletOptions`	Registrare un componente in un outlet (default AFTER); serve `OutletModule.forRoot()` (già in `BaseStorefrontModule`) o `forChild()` nei moduli lazy	core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L17-L47	51 — outlet
`OutletModule.forRoot()` / `forChild()`	`forChild()` nei moduli lazy che dichiarano `provideOutlet`	core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88	51, 53 — feature lazy
`[cxOutletRef]` + `cxOutletPos` (`OutletRefModule` o direttiva standalone)	Sostituire/arricchire da template, senza scrivere un componente	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L17-L35	51
`[cxOutlet]` nei propri template	Aprire nuovi punti di estensione nei propri componenti; `cxOutletContext`, `cxOutletDefer`, `(loaded)`, `[(cxComponentRef)]`	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L33-L63	51

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`OutletContextData<T>`	Iniettarlo (meglio `@Optional()`) nel componente registrato e leggere `context$`	core-libs/storefront/cms-structure/outlet/outlet.model.ts:L21-L44	51
`OutletService`	Registrazione imperativa (`add`/`remove`) o sostituzione via DI; `removeValueOrAll` è `protected`; la direttiva tollera `get` che ritorna un singolo oggetto	core-libs/storefront/cms-structure/outlet/outlet.service.ts:L10-L98, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L141-L145	50 — override DI
`OutletRendererService`	Forzare il ri-render di un outlet dopo un `add` tardivo	core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L48	51
`LayoutConfig.launch` con `outlet`	Aprire un componente in un outlet tramite launch dialog	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L38-L47	51, cap. 18
`LayoutConfig.deferredLoading`	Strategia globale e margine dell'intersezione	core-libs/storefront/layout/config/layout-config.ts:L109-L121	cap. 18
Enum `*Outlets` (es. `CartOutlets`)	Nomi stabili da usare invece di stringhe	feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L10	Appendice A08

Non pubblici (non usarli, vedi 54 — upgrade-safe): `PROVIDE_OUTLET_OPTIONS` (core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L13-L15) e `registerOutletsFactory` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L47); nessun file fuori dalla cartella outlet li referenzia (verificato allo SHA con `grep -rln "PROVIDE_OUTLET_OPTIONS\|registerOutletsFactory"`: compaiono solo in `outlet.providers.ts` e `outlet.module.ts`, core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L29).

Esempio minimale, ricostruito solo con simboli pubblici (`provideOutlet`, `OutletPosition`, `CartOutlets` e il pattern di cart-item-validation-quantity-hint.module.ts:L16-L20):

```ts
// Ispirato a feature-libs/cart/base/components/validation/cart-item-quantity-hint/cart-item-validation-quantity-hint.module.ts:L16-L20
providers: [
  provideOutlet({
    id: CartOutlets.ITEM_VALIDATION_QUANTITY_HINT,
    position: OutletPosition.REPLACE,
    component: CartItemValidationQuantityHintComponent,
  }),
],
```

9. Trappole, casi limite, comportamenti controintuitivi

- **T1 — Default di posizione incoerenti.** `provideOutlet` → AFTER (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42); `OutletService.add` e `cxOutletRef` senza `cxOutletPos` → REPLACE (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L34-L34); launch dialog → BEFORE (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L56-L56). Chi si

aspetta di "sostituire" con `provideOutlet` senza `position` ottiene un'aggiunta in coda al contenuto di default.

- **T2 — REPLACE multipli si sommano.** La direttiva legge con `USE_STACKED_OUTLETS`, quindi due registrazioni in REPLACE vengono renderizzate entrambe (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L131-L133); solo `get()` con lo `stacked` di default ritorna il primo (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L62-L65). "Replace" significa "sostituisci il default", non "l'ultimo vince".
- **T3 — Default perso dopo una rimozione.** Il contenuto di default compare solo se lo store REPLACE restituisce `undefined`; se la chiave esiste ma l'array è vuoto (dopo `remove` con `value` che filtra tutto), `templates` è `[]` e non si vede niente, nemmeno il default (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L135-L139, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L91-L94). Per ripristinare il default serve `remove` senza `value`, che cancella la chiave (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L86-L87). Nota: `OutletRefDirective` rimuove sempre con `value` (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L34-L34).
- **T4 — `[undefined]` innocuo.** Per BEFORE/AFTER vuoti `templates` diventa `[undefined]`; `create(undefined)` lo aggiunge a `renderedTemplate` e non crea nulla (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L143-L145, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L165-L165).
- **T5 — Stessa voce in due posizioni.** `renderedTemplate` è condiviso fra le tre posizioni dello stesso `build()`: lo stesso `TemplateRef`/factory registrato in due posizioni viene creato solo nella prima secondo l'ordine BEFORE → REPLACE → AFTER (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L121-L135).
- **T6 — Il registro non è reattivo.** Un `add` successivo al render di un outlet non appare finché la direttiva non ri-esegue `render()` (cambio di `cxOutlet` o `OutletRendererService.render`) (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L30-L42, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L92-L96). Per `forChild` nei moduli lazy il problema non si pone quando l'outlet sta in un componente dello stesso modulo (gli initializer girano subito dopo la creazione del modulo, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L98), ma un outlet già renderizzato da un altro modulo (es. `ORDER_SUMMARY` nei dettagli d'ordine, feature-libs/order/components/order-details/order-detail-totals/order-detail-totals.component.html:L4-L7, riempito dal modulo cart, feature-libs/cart/base/components/cart-shared/cart-shared.module.ts:L55-L58) dipende dall'ordine di caricamento. DA VERIFICARE a runtime: l'ordine fra caricamento delle feature lazy e primo rendering degli outlet non è determinabile dalla sola lettura del codice.
- **T7 — Il contesto non aggiorna i template.** Cambiare `cxOutletContext` emette su `outletContext$` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L97-L99), ma le viste da `TemplateRef` hanno il `$implicit` fissato alla creazione (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L176-L181); solo i componenti, via `context$`, vedono i nuovi valori. Anche `OutletContextData.context` resta il valore iniziale (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L39). Per contesti che cambiano, passare Observable nel contesto (come fanno `PageLayoutComponent` e `PageSlotComponent` con `slots$`, `components$`: core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L5-L9, core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L4-L4).
- **T8 — `cxOutletDefer` non è "reattivo".** Cambiare solo `cxOutletDefer` non provoca re-render (solo `changes.cxOutlet` lo fa) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L93-L96).
- **T9 — `loaded` solo col defer.** `loaded` è emesso solo nel ramo defer (`false` poi `true`); senza `cxOutletDefer` non arriva nulla (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L102-L116, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L85-L89). `PageSlotComponent` funziona perché `getComponentDeferOptions` restituisce sempre un oggetto truthy (core-libs/storefront/cms-

structure/page/slot/page-slot.service.ts:L63-L73): un override che restituisse `undefined` lascerebbe lo slot `cx-pending` per sempre.

- **T10 — `loaded` è asincrono.** `EventEmitter(true)` emette in un tick successivo, quindi `loaded(true)` arriva dopo `build()` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L58-L58).
- **T11 — Il defer osserva l'host, non il contenuto.** L'elemento osservato è il primo `HTMLElement` antenato del commento dell'`ng-template` (es. l'host `cx-page-slot`) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L104-L104, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L225-L230).
- **T12 — In SSR tutto è istantaneo.** `DeferLoaderService` carica subito su server (core-libs/storefront/layout/loading/defer-loader.service.ts:L58-L59); `IntersectionService` su server ritornerebbe `of(false)` (core-libs/storefront/layout/loading/intersection.service.ts:L50-L53), ma non viene interpellato.
- **T13 — Direttive distrutte nel renderer.** `ngOnDestroy` non deregistra da `OutletRendererService` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L232-L235): la Map conserva direttive distrutte e `render(name)` potrebbe invocare `render()` su una di esse (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L28). DA VERIFICARE: l'effetto di `vcr.clear()`/`createComponent` su un `ViewContainerRef` già distrutto non è stato provato a runtime.
- **T14 — Nomi duplicati nel renderer.** La chiave di `OutletRendererService` è il nome: più `[cxOutlet]` con lo stesso nome (es. lo stesso `flexType` ripetuto in più slot) si sovrascrivono e vince l'ultimo registrato (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L36-L38, core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L9-L9). `render(name)` rigenera quindi solo una delle istanze.
- **T15 — `getOutletRef` emette troppo.** `register` muta la Map in place e la ri-emette; `getOutletRef` non ha `distinctUntilChanged` ed emette a ogni registrazione di *qualunque* outlet (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L37-L37, core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L43-L48).
- **T16 — `reverse()` muta lo stato della direttiva.** In `OutletRenderStrategy.render`, `components.reverse()` inverte in place l'array di `renderedComponents` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L67-L68). Se l'outlet fosse differito e non ancora costruito, `renderedComponents.get(position)` sarebbe `undefined` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L81-L81) e `.reverse()` lancerebbe. DA VERIFICARE: non è stato cercato se esistano launch config su outlet con `cxOutletDefer` (l'esempio ASM usa `cx-storefront`, senza defer: core-libs/storefront/layout/main/storefront.component.html:L1-L1).
- **T17 — `remove` del launch dialog con una factory nuova.** `OutletRenderStrategy.remove` risolve una nuova factory e la passa come `value` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L87-L95); dato che `resolveComponentFactory` crea un'istanza nuova a ogni chiamata (angular:packages/core/src/render3/component_ref.ts:L97-L101), il filtro per identità (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L91-L91) non dovrebbe trovare la factory aggiunta in `render()`. Lo spec usa un resolver mock che restituisce sempre lo stesso oggetto e non testa `remove` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.spec.ts:L40-L55, core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.spec.ts:L57-L133). Coerente con questo, il JSDoc di `LaunchOutletDialog` avverte: "Outlet rendering should only be used for elements that open once and do not close." (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L34-L37). DA VERIFICARE con un test reale: è una deduzione dal codice, gli spec non sono eseguibili (node_modules assente).
- **T18 — Posizioni non valide.** `add` ignora in silenzio posizioni sconosciute, `get`/`remove` ricadono su REPLACE (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L36-L37, core-libs/storefront/cms-

structure/outlet/outlet.service.ts:L58-L60, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L74-L76).

- **T19 — Nessuna deduplica.** La stessa opzione registrata due volte produce due rendering (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L38-L40).
- **T20 — Possibili doppie registrazioni con** `forChild`**.** Le deps di `registerOutletsFactory` sono risolte senza `Self` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L79-L83); un injector di modulo senza propri `PROVIDE_OUTLET_OPTIONS` delega al parent (angular:packages/core/src/di/r3_injector.ts:L345-L347, angular:packages/core/src/di/r3_injector.ts:L374-L376) e ri-registrerebbe le opzioni del parent. Inoltre un `forChild` finito nell'injector root fa girare la factory sia via `MODULE_INITIALIZER` (core-libs/core/src/lazy-loading/lazy-loading.module.ts:L44-L49) sia accanto a `forRoot` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L58-L66). DA VERIFICARE a runtime: dipende da come i feature module sono caricati (eager o lazy) nell'app.
- **T21 — Environment injector del componente.** Con `forRoot` la factory è legata al `NgModuleRef` root (con `bootstrapApplication`, come nell'app demo, è un `EnvironmentNgModuleRefAdapter`: angular:packages/core/src/application/create_application.ts:L72-L72, angular:packages/core/src/render3/ng_module_ref.ts:L147-L148), con `forChild` all'`NgModuleRef` lazy; quel `NgModuleRef` diventa l'environment injector del componente, indipendentemente da dove sta l'outlet (angular:packages/core/src/render3/component_ref.ts:L282-L286, angular:packages/core/src/linker/view_container_ref.ts:L536-L537). Un componente che dipende da servizi forniti da un modulo lazy deve quindi essere registrato con `forChild` in quel modulo.
- **T22 — API Angular deprecate.** `ComponentFactoryResolver`, `ComponentFactory` (angular:packages/core/src/linker/component_factory_resolver.ts:L38-L41, angular:packages/core/src/linker/component_factory.ts:L91-L94), l'overload factory di `createComponent` (angular:packages/core/src/linker/view_container_ref.ts:L263-L270) e `APP_INITIALIZER` (angular:packages/core/src/application/application_init.ts:L38-L38) sono deprecati; il registro è tipato su `ComponentFactory` (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L13-L13).
- **T23 — Classi componente ignorate.** Passare a `OutletService.add` una classe componente invece di una factory non produce nulla (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L167-L190).
- **T24 — Esempio demo incoerente.** Il template demo legge `model.sections$` ma `PageLayoutComponent` fornisce `section$` (projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-template.component.html:L7-L7, core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L8-L8).
- **T25 —** `cxComponentRef` **non è un input "vero".** Non viene mai letto dalla direttiva; serve solo come metà del binding two-way (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L60-L63). Uso reale in `AddToCartComponent` (feature-libs/cart/base/components/add-to-cart/add-to-cart.component.html:L24-L28).
- **T26 —** `cxComponentRefChange` **anche per il default.** È emesso per ogni vista creata, incluso il template di default REPLACE (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L137-L139, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L187-L187): chi ascolta l'evento deve distinguere i ref (es. con `instanceof ComponentRef`).

---

10. Come fare debug

1. **Il mio componente non compare.** Verificare in ordine:

- che il modulo con `provideOutlet` importi `OutletModule.forChild()` se è lazy, oppure che le opzioni finiscano nel root injector dove opera `forRoot()` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88);
- che `id` coincida esattamente con il valore stringa usato in `[cxOutlet]` (gli enum hanno valori come `'cx-order-summary'`, feature-libs/cart/base/root/models/cart-outlets.model.ts:L18-L18);
- lo stato del registro: da un componente (injection context), `inject(OutletService).get('nome', OutletPosition.AFTER, true)` mostra l'array (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L53-L66). Dalla console `inject()` non funziona: in dev mode si può usare `ng.getInjector(el).get(...)` passando la stringa `'after'`, perché i valori dell'enum sono stringhe (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L13). DA VERIFICARE: l'uso di `ng.getInjector` in console dipende dalla build in dev mode, non verificato nel repo;
- se l'aggiunta è avvenuta dopo il render, forzarlo con `OutletRendererService.render('nome')` (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L28) e vedere se compare (conferma T6).

2. **Il contenuto di default è sparito.** Controllare se la chiave REPLACE esiste con array vuoto (T3): `get(nome, OutletPosition.REPLACE, true)` restituisce `[]` invece di `undefined`.
3. **Vedo il mio componente due volte.** Cercare doppie `provideOutlet` (T19) o doppie esecuzioni della factory (T20); un breakpoint in `OutletService.add` (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L36-L41) mostra chiamante e stack.
4. **Il componente non vede il contesto aggiornato.** Usare `context$`, non `context` (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L43); per template, passare Observable (T7).
5. **Slot sempre** `cx-pending`**.** Verificare che arrivi `loaded(true)`: breakpoint in `deferLoading()` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L111-L114) e in `PageSlotComponent.isLoaded` (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L183-L188); controllare la strategia globale `deferredLoading.strategy` (core-libs/storefront/layout/loading/defer-loader.service.ts:L30-L31) e che l'host sia davvero visibile (`IntersectionObserver`, core-libs/storefront/layout/loading/intersection.service.ts:L103-L118).
6. **Il componente lanciato in outlet non si chiude.** Vedi T17: ispezionare l'array dopo `remove` con `get(outlet, OutletPosition.BEFORE, true)`.
7. **Ispezionare i ref creati.** `renderedComponents` è pubblico sulla direttiva (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L36-L39); `OutletRendererService.getOutletRef('nome')` restituisce la direttiva registrata (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L43-L48).
8. **Test unitari.** Il registro è una classe senza dipendenze (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L13-L18): si può istanziare con `new OutletService()` e verificare `add`/`get`/`remove` in isolamento.

---

11. Domande di autoverifica

1. **Registro con** `provideOutlet({ id: 'x', component: C })` **senza** `position`**: dove compare** `C`**?** Dopo il contenuto di default, perché `registerOutletsFactory` usa `options.position ?? OutletPosition.AFTER` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42) e il default REPLACE resta il `templateRef` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L137-L139).
2. **Due moduli registrano due componenti diversi in REPLACE sullo stesso outlet. Cosa si vede?** Entrambi, nell'ordine di registrazione: la direttiva legge con `USE_STACKED_OUTLETS` (core-libs/storefront/cms-

structure/outlet/outlet.directive.ts:L131-L133) e add accoda (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L38-L40).

3. **Perché in un modulo lazy serve** `OutletModule.forChild()` **e non basta** `provideOutlet`**?** Perché le opzioni vengono lette da una funzione registrata su `MODULE_INITIALIZER` che `LazyModulesService` esegue leggendo l'injector del modulo con `{ self: true }` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157); l'`APP_INITIALIZER` di `forRoot` è già stato eseguito al bootstrap e vede solo il root injector (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L70).
4. **Un componente inserito via outlet come ottiene i dati? E un template?** Il componente inietta `OutletContextData` dall'injector creato in `getComponentInjector` e legge `context$` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L196-L212); il template riceve il contesto come `$implicit` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L176-L181).
5. **Se il contesto cambia, chi vede il nuovo valore?** Solo chi è sottoscritto a `context$` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L97-L99); `$implicit` dei template e `OutletContextData.context` restano al valore iniziale (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L179-L179, core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L39).
6. **Aggiungo un template con** `OutletService.add` **dopo che l'outlet è già a schermo. Perché non appare e come lo faccio apparire?** Il registro è una `Map` senza notifiche (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L13-L42); bisogna chiamare `OutletRendererService.render(nome)`, che invoca `render()` sulla direttiva registrata (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L28), come fa `OutletRenderStrategy` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L58-L58).
7. **In SSR un outlet con** `cxOutletDefer` **viene renderizzato?** Sì: `DeferLoaderService.shouldLoadInstantly` ritorna `true` su piattaforma server e `load` restituisce `of(true)` (core-libs/storefront/layout/loading/defer-loader.service.ts:L44-L64).
8. **Perché un** `cxOutletRef` **senza** `cxOutletPos` **sostituisce il contenuto, mentre** `provideOutlet` **senza** `position` **lo affianca?** `OutletRefDirective` passa `undefined` ad `add`, che applica il default REPLACE (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L30-L30, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L34-L34); `registerOutletsFactory` invece forza AFTER (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42).
9. **Dopo che tutti i template REPLACE registrati via** `cxOutletRef` **sono stati distrutti, torna il contenuto di default?** No: `remove` con `value` lascia la chiave con array vuoto (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L88-L95) e la direttiva usa il default solo se il risultato è `undefined` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L137-L139). Inoltre, senza re-render, la vista resta quella già creata (T6).
10. **Quale environment injector usa un componente registrato con** `provideOutlet` **in un modulo lazy con** `forChild()`**?** L'`NgModuleRef` lazy, perché la factory è risolta con il `ComponentFactoryResolver` di quel modulo e `createComponent` usa l'`ngModule` della factory (angular:packages/core/src/render3/ng_module_ref.ts:L64-L65, angular:packages/core/src/render3/component_ref.ts:L282-L286, angular:packages/core/src/linker/view_container_ref.ts:L531-L537).

TAVOLA 17.1 **4.1 Chi scrive nel registro**

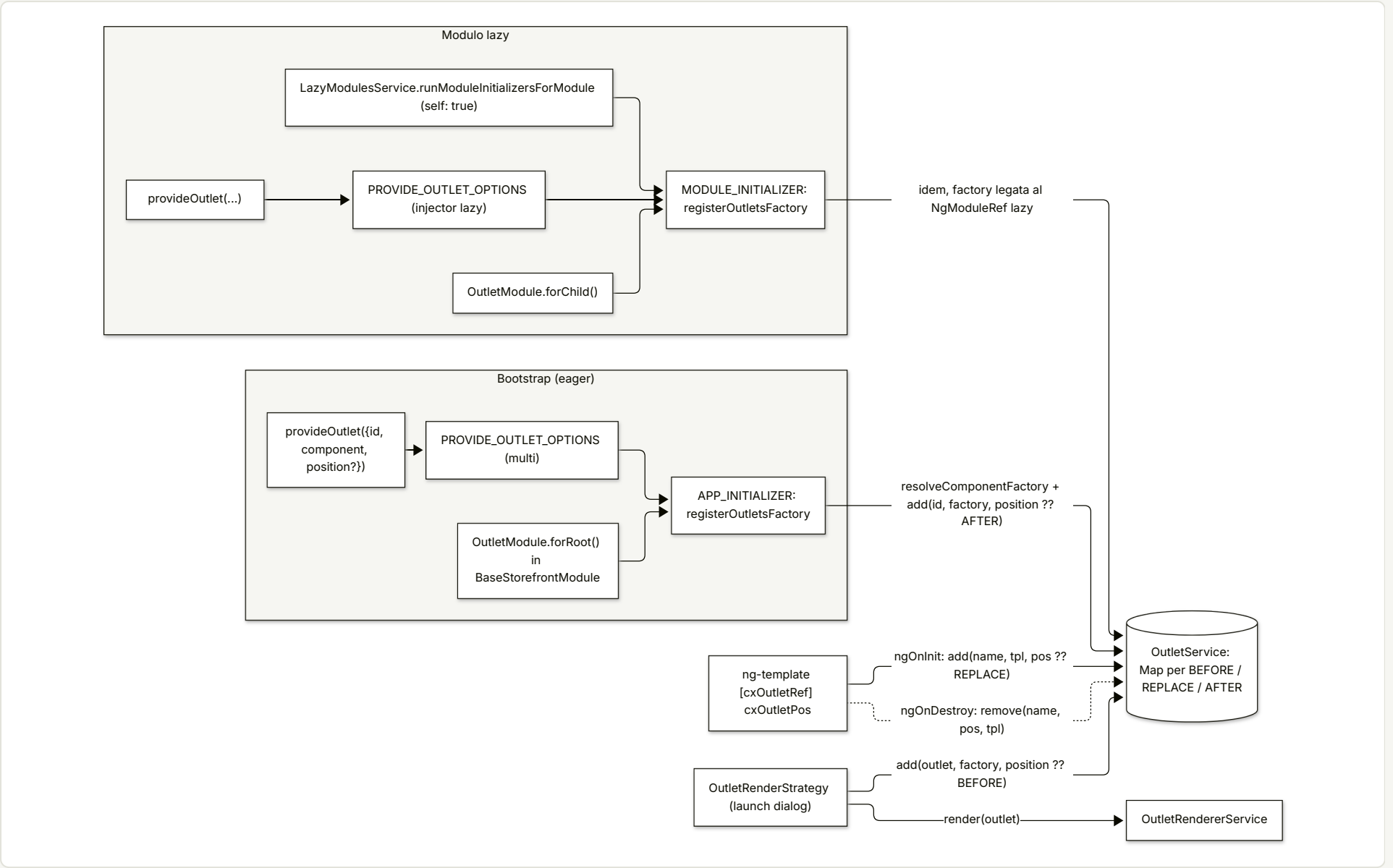


TAVOLA 17.2 **4.2 Rendering di un outlet (con e senza defer)**

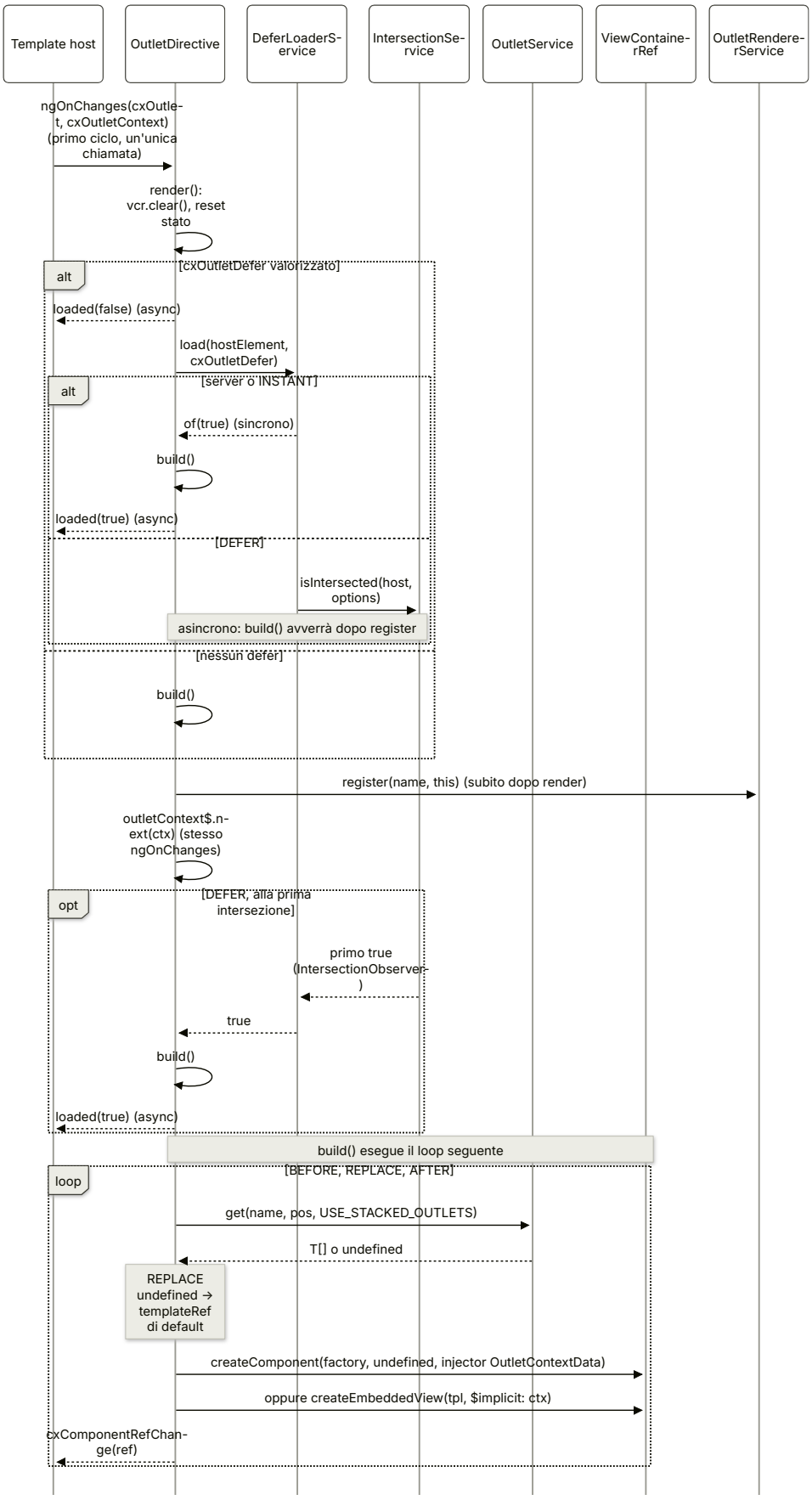
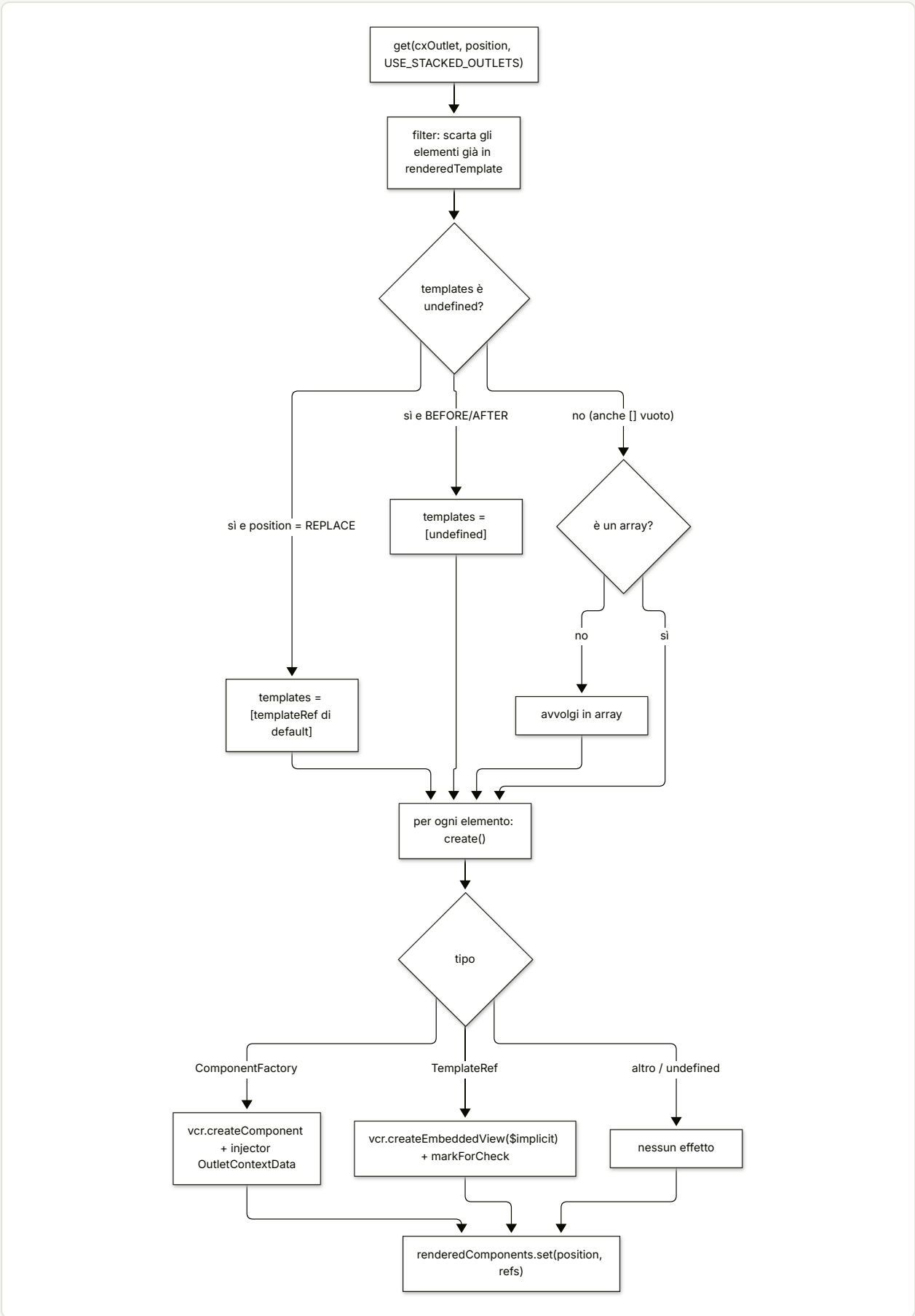


TAVOLA 17.3 **4.3 Decisione di buildOutlet per una posizione**



18

Layout, breakpoint, launch dialog e accessibilità

Il "guscio" dello storefront in `core-libs/storefront/layout/`: breakpoint, dialog, a11y, direction, header.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 11 — Configurazione, 16 — CMS (storefront): rendering dinamico, 17 — Outlet. Per i feature toggle citati vedi 42 — Feature toggle e deprecazioni e l'Appendice A09.

1. A cosa serve e perché esiste

La cartella `core-libs/storefront/layout/` contiene il "guscio" dello storefront: tutto ciò che non è un componente CMS ma serve a disporre, aprire e rendere navigabili i componenti CMS. Il barrel `core-libs/storefront/layout/index.ts` riesporta a11y, breakpoint, config, direction, header, launch-dialog, `layout.module`, loading, main e theme (`core-libs/storefront/layout/index.ts:L7-L16`), e `@spartacus/storefront` lo riesporta per intero (`core-libs/storefront/public_api.ts:L26-L26`).

Il capitolo copre cinque sottosistemi che collaborano:

1. `LayoutConfig`: la configurazione che dice quali slot CMS mostrare in ogni sezione (header, footer, ...) o page template, eventualmente diversi per breakpoint, più i breakpoint stessi, la strategia di deferred loading e la configurazione dei dialog (`launch`) (`core-libs/storefront/layout/config/layout-config.ts:L90-L125`). Il JSDoc dichiara esplicitamente che il design adattivo è "per breakpoint (not per device type)": il DOM viene (ri)renderizzato per breakpoint, non nascosto via CSS (`core-libs/storefront/layout/config/layout-config.ts:L84-L89`).
2. `BreakpointService`: trasforma la larghezza della finestra in un valore dell'enum `BREAKPOINT` (`xs`...`xl`) osservabile, con fallback "mobile first" in SSR (`core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L40-L46`, `core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L123-L131`).
3. **Launch dialog**: `LaunchDialogService` apre componenti (dialog, popover, sidebar, ma anche navigazioni) identificati da una chiave `LAUNCH_CALLER`, scegliendo una di quattro strategie di rendering in base alla configurazione (`core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L78-L97`, `core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L18-L41`).
4. **La shell** `cx-storefront` (`StorefrontComponent`): il template radice con header, main, footer, outlet, skip link, gestione del focus dopo la navigazione e del menu hamburger (`core-libs/storefront/layout/main/storefront.component.html:L1-L28`, `core-libs/storefront/layout/main/storefront.component.ts:L51-L67`), più il **deferred loading** (`DeferLoaderService`, `IntersectionService`) usato dagli outlet.
5. **Accessibilità da tastiera**: la directive `[cxFocus]` (una catena di nove classi: Base → Visible → Block → Persist → Escape → AutoFocus → Tab → Trap → Lock), `[cxSkipFocus]`, le skip link e alcune directive di supporto (`core-libs/storefront/layout/a11y/keyboard-focus/focus.directive.ts:L12-L25`, `core-libs/storefront/layout/a11y/index.ts:L7-L13`).

Il tutto entra nell'applicazione tramite `BaseStorefrontModule`, che importa `SkipLinkModule`, `KeyboardFocusModule`, `LayoutModule` e `StorefrontComponentModule` e riesporta gli ultimi due (`core-libs/storefront/base-storefront.module.ts:L26-L49`). `LayoutModule` a sua volta importa `OutletRefModule`, `LaunchDialogModule.forRoot()`, `DirectionModule`, `ThemeModule` e registra i breakpoint di default con `provideDefaultConfig(defaultLayoutConfig)` (`core-libs/storefront/layout/layout.module.ts:L15-L25`).

2. Mappa dei file

PATH	RUOLO
core-libs/storefront/layout/index.ts	Barrel pubblico del layout (:L7-L16).
core-libs/storefront/layout/layout.module.ts	LayoutModule: default breakpoints, LaunchDialogModule.forRoot(), Direction/Theme (:L15-L25).
core-libs/storefront/layout/config/layout-config.ts	BREAKPOINT, LayoutSections, SlotConfig, SlotGroup, LayoutSlotConfig, BreakPoint, LayoutBreakPoints, LayoutConfig + augmentation di Config (:L11-L129).
core-libs/storefront/layout/config/default-layout.config.ts	defaultLayoutConfig — solo breakpoints (:L9-L19).
core-libs/storefront/recipes/config/layout-config.ts	Recipe layoutConfig (deprecata) con i layoutSlots standard e layoutConfigFactory() (:L25-L198).
core-libs/storefront/layout/breakpoint/breakpoint.service.ts	BreakpointService: breakpoint$, isUp/isDown/isEqual, getSize (:L19-L207).
core-libs/storefront/layout/launch-dialog/config/launch-config.ts	LaunchConfig, LaunchOptions e sottotipi, DIALOG_TYPE, LAUNCH_CALLER (:L9-L103).
core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts	Multi-provider delle 4 strategie; forRoot() alias LayoutConfig (:L18-L49).
core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts	LaunchDialogService: openDialog, launch, clear, closeDialog, data$ (:L24-L186).
core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts	Classe astratta LaunchRenderStrategy (shouldRender, applyClasses, remove, priorità LOW) (:L24-L158).
core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts	Render dentro un outlet via OutletService (:L45-L98).
core-libs/storefront/layout/launch-dialog/services/inline-render.strategy.ts	Render nel ViewContainerRef del chiamante (:L42-L75).
core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts	Render nel root component (ApplicationRef.components[0]) (:L21-L70).
core-libs/storefront/layout/launch-dialog/services/routing-render.strategy.ts	"Apertura" come navigazione RoutingService.go (:L22-L32).
core-libs/storefront/layout/main/storefront.component.ts / .html	StorefrontComponent (cx-storefront): shell, focus, Escape, hamburger, navigazione.
core-libs/storefront/layout/main/storefront-component.module.ts	StorefrontComponentModule (:L19-L34).
core-libs/storefront/layout/main/storefront-outlets.model.ts	Enum StorefrontOutlets (:L7-L9).
core-libs/storefront/layout/header/hamburger-menu/*	HamburgerMenuService (stato isExpanded), HamburgerMenuComponent, HamburgerMenuModule (mapping CMS).
core-libs/storefront/layout/loading/defer-loader.service.ts	DeferLoaderService: caricamento immediato o al viewport (:L22-L64).

PATH	RUOLO
core-libs/storefront/layout/loading/intersection.service.ts / intersection.model.ts	IntersectionService (IntersectionObserver), IntersectionOptions.
core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts	Attributi data-cx-focus*, MOVE_FOCUS, gerarchia di config, TrapFocus (:L7-L149).
core-libs/storefront/layout/a11y/keyboard-focus/focus.directive.ts	FocusDirective [cxFocus], foglia della catena (:L12-L25).
core-libs/storefront/layout/a11y/keyboard-focus/{base,visible,block,persist,escape,autofocus,tab,trap,lock}/	Le nove directive astratte della catena e i servizi paralleli.
core-libs/storefront/layout/a11y/keyboard-focus/services/keyboard-focus.service.ts / select-focus.util.ts	KeyboardFocusService (foglia dei servizi), SelectFocusUtility (query dei focusable).
core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts	SkipFocusDirective [cxSkipFocus] (:L17-L78).
core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.module.ts	KeyboardFocusModule: solo FocusDirective + SkipFocusDirective (:L12-L29).
core-libs/storefront/layout/a11y/keyboard-focus/focus-testing.module.ts	KeyboardFocusTestingModule per gli unit test (:L10-L19).
core-libs/storefront/layout/a11y/skip-link/*	Config, directive [cxSkipLink], SkipLinkService, SkipLinkComponent, SkipLinkModule + skipLinkFactory.
core-libs/storefront/layout/a11y/{btn-like-link,focus-first-invalid-field,native-select-space,on-dom-change,truncation-tooltip}/	Directive a11y di supporto.

File di altri capitoli toccati dai flussi:

PATH	RUOLO NEL FLUSSO
core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts	Combina pagina e breakpoint per risolvere gli slot (:L55-L59, :L187-L191, :L217-L243). Dettagli nel cap. 16.
core-libs/storefront/cms-structure/outlet/outlet.directive.ts	Consumer del deferred loading (:L102-L116). Vedi cap. 17.
core-libs/core/src/window/window-ref.ts	resize$ con debounce e startWith (:L91-L107).
core-libs/core/src/util/applicable.ts	resolveApplicable, usato per scegliere la strategia (:L47-L74).

3. Come funziona passo per passo

3.1 LayoutConfig: tipi e default

- BREAKPOINT è un enum di stringhe xs, sm, md, lg, xl; il JSDoc dice che il tipo "can be extended to allow for custom screens, such as XLL or tablet" (core-libs/storefront/layout/config/layout-config.ts:L11-L28) e il public API lo marca fra i tipi augmentabili (blocco AUGMENTABLE_TYPES_START … AUGMENTABLE_TYPES_END) (core-libs/storefront/public_api.ts:L32-L36).
- LayoutSections è 'header' | 'footer' | 'LandingPage2Template' | string, cioè qualunque stringa (core-libs/storefront/layout/config/layout-config.ts:L30-L34): sia una sezione (header, navigation, footer) sia un nome di page template.

- `SlotConfig` ha `slots?: string[]` (posizioni degli slot CMS, mappate su `slot.position`) e `pageFold?: string` (ultimo slot atteso above-the-fold) (core-libs/storefront/layout/config/layout-config.ts:L36-L46).
- `SlotGroup` permette config per `lg`, `md`, `sm`, `xs` — **non** `xl` (core-libs/storefront/layout/config/layout-config.ts:L48-L57).
- `LayoutSlotConfig` è ricorsivo: ogni sezione mappa a `SlotConfig | SlotGroup | LayoutSlotConfig` (core-libs/storefront/layout/config/layout-config.ts:L59-L61), così un page template può contenere a sua volta config per sezioni.
- `BreakPoint` ha `min?`/`max?` e `LayoutBreakPoints` mappa ogni `BREAKPOINT` a `number | BreakPoint` (core-libs/storefront/layout/config/layout-config.ts:L63-L82).
- `LayoutConfig` è una classe astratta `@Injectable({ providedIn: 'root', useExisting: Config })` con `breakpoints?`, `layoutSlots?`, `deferredLoading?: { strategy?, intersectionMargin? }` e `launch?: LaunchConfig` (core-libs/storefront/layout/config/layout-config.ts:L90-L125), fusa nell'interfaccia `Config` di `@spartacus/core` con module augmentation (core-libs/storefront/layout/config/layout-config.ts:L127-L129). È il pattern standard descritto nel cap. 11.

`defaultLayoutConfig` contiene **solo** i breakpoint: `xs: 576, sm: 768, md: 992, lg: 1200, xl: { min: 1200 }` (core-libs/storefront/layout/config/default-layout.config.ts:L9-L19). I `layoutSlots` standard vivono altrove, nella recipe `layoutConfig`, marcata `@deprecated Use layoutConfigFactory and provideConfigFactory(layoutConfigFactory) instead` (core-libs/storefront/recipes/config/layout-config.ts:L25-L51). Nella recipe l'header ha una config `lg` con 8 slot e una di default di 4 (`'PreHeader','SiteLogo','SearchBox','MiniCart'`), mentre `navigation` ha `lg: { slots: [] }` (core-libs/storefront/recipes/config/layout-config.ts:L25-L51); `LandingPage2Template` ha `pageFold: 'Section2B'` e `CategoryPageTemplate` `pageFold: 'Section2'` (core-libs/storefront/recipes/config/layout-config.ts:L55-L73), e `ProductDetailsPageTemplate` ha `pageFold: 'Summary'` (`lg`: `'UpSelling'`, unico `pageFold` responsivo della recipe) (core-libs/storefront/recipes/config/layout-config.ts:L96-L100); la factory li rimuove tutti e tre (core-libs/storefront/recipes/config/layout-config.ts:L172-L185).

`layoutConfigFactory()` clona la recipe via JSON, applica `applyUnifiedHeaderSlots` (rimuove `header.lg` e mette tutti gli 8 slot nel default) e `applyWithoutPageFold` (rimuove i `pageFold`) (core-libs/storefront/recipes/config/layout-config.ts:L190-L198). La rimozione del `pageFold` è motivata da regressioni di CLS: il `pageFold` faceva comparire i componenti CMS "only after a small delay even in SSR pages" (core-libs/storefront/recipes/config/layout-config.ts:L158-L171).

3.2 `BreakpointService`: dalla larghezza al breakpoint

1. Il servizio è `providedIn: 'root'` e il JSDoc dichiara la base Bootstrap: xs < 576, sm 576–767, md 768–991, lg 992–1199, xl ≥ 1200 (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L19-L37).
2. Il costruttore dichiara `winRef`, `layoutConfig` e `platform` come parameter properties `protected` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L48-L52): per questo il field initializer `breakpoint$` può già usarli.
3. In browser `breakpoint$` = `winRef.resize$` → `getBreakpoint(innerWidth)` → `distinctUntilChanged()` → `shareReplay({ refCount: true, bufferSize: 1 })`; fuori dal browser è `of(this.fallbackBreakpoint)` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L40-L46).
4. `WindowRef.resize$` ascolta `resize` con `debounceTime(300)`, simula un primo evento con `startWith({ target: this.nativeWindow })` e, senza window, restituisce `of(null)` (core-libs/core/src/window/window-ref.ts:L91-L107). Grazie a `startWith` il breakpoint è disponibile subito, senza aspettare un resize.

5. La lista ordinata dei breakpoint è calcolata e memoizzata in `_breakpoints` alla prima lettura del getter `breakpoints` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L60-L65). L'ordinamento usa un comparatore che, per ogni breakpoint, considera `max(min+1, max)` e restituisce `-1` o `0` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L138-L153). La sorgente è `layoutConfig?.breakpoints || {}` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L204-L206).
6. `getMaxSize` tratta un numero come max, altrimenti usa `.max`, altrimenti `null` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L161-L176); `getMinSize` legge `.min` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L178-L180).
7. `getSize(bp)` restituisce il max del breakpoint o, in mancanza, il `min` del breakpoint successivo (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L71-L80). Con il default, `xl` non ha né max né successivo: `getSize(xl)` è falsy, come conferma lo spec (core-libs/storefront/layout/breakpoint/breakpoint.service.spec.ts:L185-L188).
8. `getBreakpoint(width)` restituisce il primo breakpoint per cui `width < getSize(br)`, altrimenti l'ultimo della lista (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L192-L199). Con il default: 575 → `xs`, 576 → `sm`, 991 → `md`, 1199 → `lg`, 1200 → nessun match → `xl`.
9. In SSR `fallbackBreakpoint` è il primo della lista ordinata ("mobile first"), cioè `xs` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L123-L131).
10. Le API derivate: `isDown(bp)` è vero se il breakpoint corrente sta fra l'indice 0 e `bp` incluso (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L89-L97); `isUp(bp)` se è `bp` o successivo (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L106-L114); `isEqual(bp)` confronta in modo stretto (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L119-L121). Esempi: `NavigationUIComponent.isDesktop$ = isUp(BREAKPOINT.lg)` (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L107-L107), `SearchBoxComponent` usa `isDown(BREAKPOINT.sm)` (core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L222-L222).

3.3 Dagli slot configurati agli slot renderizzati

Il consumatore principale di `breakpoint$` è `PageLayoutService` (cap. 16), che fa `combineLatest([this.page$, this.breakpointService.breakpoint$])` e risolve gli slot per sezione/template (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L55-L59):

1. per una sezione dentro un page template, se il template ha una voce in `layoutSlots` ma non per quella sezione si usa la config globale `layoutSlots[section]`; se il template non ha alcuna voce in `layoutSlots`, la sezione non riceve slot, anche se esiste `layoutSlots[section]` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L181-L191);
2. `getResponsiveSlotConfig` usa la config del breakpoint esatto se contiene l'attributo cercato (`slots` o `pageFold`); altrimenti scorre i breakpoint **più piccoli** di quello corrente e tiene l'ultimo che lo definisce; altrimenti usa la config di default (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243).

Conseguenza: una config `lg` vale anche per `xl` (che `SlotGroup` non prevede), ma non per `md`.

3.4 Launch dialog: configurazione

`LaunchConfig` è un dizionario `[key: string]: LaunchOptions` e `LaunchOptions` è l'unione di quattro forme (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L9-L17):

FORMA	DISCRIMINANTE	NOTE	REF
`LaunchOutletDialog`	`outlet: string` (+ `position?`, default BEFORE)	"should only be used for elements that open once and do not close"	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L34-L47
`LaunchInlineDialog`	`inline: boolean`	accanto al trigger (serve un `ViewContainerRef`)	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L49-L54
`LaunchInlineRootDialog`	`inlineRoot: boolean`	"directly inside the cx-storefront", consigliato per dialog globali	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L56-L62
`LaunchRoute`	`cxRoute: string` (+ `params?`)	navigazione, nessun componente	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L64-L76

Le prime tre estendono `LaunchDialog` con `component`, `multi?` (può essere renderizzato più volte) e `dialogType?: DIALOG_TYPE` (per le classi CSS) (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L19-L32). `DIALOG_TYPE` ha sei valori: `POPOVER`, `POPOVER_CENTER`, `POPOVER_CENTER_BACKDROP`, `DIALOG`, `SIDEBAR_START`, `SIDEBAR_END` (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L78-L88).

Le chiavi sono valori di `LAUNCH_CALLER`; lo storefront ne definisce nove (`ASM`, `SKIP_LINKS`, `ANONYMOUS_CONSENT`, `REPLENISHMENT_ORDER`, `PLACE_ORDER_SPINNER`, `SUGGESTED_ADDRESSES`, `COUPON`, `CLAIM_DIALOG`, `STOCK_NOTIFICATION`) (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L90-L103) e le feature lib lo estendono con module augmentation **più** assegnazione runtime, perché un enum TypeScript augmentato non crea il valore a runtime (feature-libs/cart/base/root/models/augmented.model.ts:L7-L16). Esempi di config: `ADDED_TO_CART` è `inlineRoot` con `DIALOG_TYPE.DIALOG` (feature-libs/cart/base/components/added-to-cart-dialog/default-added-to-cart-layout.config.ts:L10-L18); `ASM` è renderizzato nell'outlet `'cx-storefront'` (feature-libs/asm/components/default-asm-layout.config.ts:L10-L17).

3.5 Launch dialog: scelta della strategia

1. `LaunchDialogModule` registra quattro provider multi `LaunchRenderStrategy` con `useExisting`, nell'ordine Outlet, Inline, Routing, InlineRoot (core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L18-L41); `forRoot()` aggiunge `{ provide: LayoutConfig, useExisting: Config }` (core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L42-L49).
2. `LaunchDialogService` riceve l'array delle strategie e `LayoutConfig` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L36-L42).
3. `findConfiguration(caller)` legge `layoutConfig.launch[caller]`; `getStrategy(config)` chiama `resolveApplicable(this.renderStrategies, [config])` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L167-L185).
4. `resolveApplicable` tiene gli applicable con `hasMatch` vero; fra più match vince la priorità più alta e, a parità, "the applicable that was provided later wins" (core-libs/core/src/util/applicable.ts:L47-L74). Tutte le strategie restituiscono `Priority.LOW` (core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L155-L157).
5. I `hasMatch`: Outlet → `Boolean(config.outlet)` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L82-L84); Inline → `Boolean(config.inline)` (core-libs/storefront/layout/launch-dialog/services/inline-render.strategy.ts:L73-L75); InlineRoot → `Boolean(config.inlineRoot)` (core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L67-L70); Routing → `Boolean(config.cxRoute)` (core-libs/storefront/layout/launch-dialog/services/routing-render.strategy.ts:L22-L32).

3.6 Launch dialog: apertura e chiusura

1. `launch(caller, vcr?, data?)`: se c'è config e strategia, resetta `_dialogClose` a `undefined`, emette `data` su `_dataSubject` e restituisce `renderer.render(config, caller, vcr)`; senza config, in dev mode, logga `No configuration provided for caller ...` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L78-L97).
2. Ogni strategia controlla `shouldRender`: serve `config.component` e, se il caller è già in `renderedCallers`, un nuovo render è permesso solo con `multi: true` (core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L92-L102).
3. Rendering per strategia:
   - **Outlet**: risolve una `ComponentFactory`, la aggiunge a `OutletService` (posizione default `BEFORE`), chiama `outletRendererService.render(outlet)` e registra il caller (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L45-L59); il `ComponentRef` viene poi recuperato dall'`OutletDirective` registrata, cercando nei `renderedComponents` invertiti il componente con lo stesso `componentType` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L61-L78). Il meccanismo outlet è descritto nel cap. 17.
   - **Inline**: crea il componente solo se è fornito un `vcr` (`vcr.createComponent(factory)`), applica le classi e restituisce `of(component)`; altrimenti, in dev mode, logga un warning (core-libs/storefront/layout/launch-dialog/services/inline-render.strategy.ts:L42-L71).
   - **InlineRoot**: `hostComponent` è `ApplicationRef.components[0]` (core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L32-L34); il componente viene creato con un `Injector.create({ providers: [] })` vuoto, la sua `hostView` viene attaccata all'`ApplicationRef` e il suo elemento appeso a quello del root component (core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L36-L64). `ApplicationRef.attachView` mette la view sotto dirty checking e la stacca automaticamente quando viene distrutta (angular:packages/core/src/application/application_ref.ts:L753-L763).
   - **Routing**: `routingService.go(config)` e nessun valore di ritorno (core-libs/storefront/layout/launch-dialog/services/routing-render.strategy.ts:L22-L32).
4. `applyClasses` aggiunge all'host del componente le classi del `dialogType` (hard-coded: DIALOG → `d-block fade modal show`, POPOVER → `cx-dialog-popover`, ecc.) e, per `DIALOG`, `modal-open` sul `body` (core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L32-L57, core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L104-L136).
5. Il componente aperto legge i dati con `launchDialogService.data$` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L32-L34).
6. `openDialog(caller, openElement?, vcr?, data?)` combina il `ComponentRef$` con `dialogClose`; alla prima chiusura non-`undefined` rimette il focus su `openElement`, chiama `clear(caller)` e distrugge il componente (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L52-L71).
7. La chiusura parte da `closeDialog(reason)`, che emette su `_dialogClose` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L134-L140). `clear(caller)` delega a `strategy.remove(caller, config)` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L121-L132), che toglie il caller da `renderedCallers` e `modal-open` dal body (core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L145-L153); la variante Outlet rimuove prima la factory da `OutletService` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L86-L98).
8. `focusElement` focalizza il primo focusable dentro il trigger (via `AutoFocusService.findFirstFocusable`) o il trigger stesso, con un `tabindex="-1"` temporaneo se assente, rimosso subito dopo (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L142-L160).

9. `openDialogAndSubscribe` sottoscrive internamente con `take(1)` ed è pensato per i trigger che possono essere distrutti mentre il dialog è aperto (`core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L99-L114`).

Esempio reale: `AddedToCartDialogEventListener` chiama `openDialog(LAUNCH_CALLER.ADDED_TO_CART, event?.triggerElementRef, undefined, addToCartData)` e sottoscrive con `take(1)` (`feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog-event.listener.ts:L57-L67`), poi chiude con `closeDialog(reason)` (`feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog-event.listener.ts:L84-L85`).

3.7 La shell `cx-storefront`

`StorefrontComponent` è standalone e importa, fra gli altri, `PageLayoutComponent`, `PageSlotComponent`, `FocusDirective`, `SkipFocusDirective`, `SkipLinkDirective`, `GlobalMessageComponent`, `OutletDirective`, `PageTemplateDirective`; non usa OnPush (`core-libs/storefront/layout/main/storefront.component.ts:L51-L67`). Nella demo app il root `app-root` ha come template solo `<cx-storefront />` (`projects/storefrontapp/src/app/app.component.ts:L15-L16`).

Struttura del template (`core-libs/storefront/layout/main/storefront.component.html:L1-L28`):

- tutto è dentro l'outlet `StorefrontOutlets.STOREFRONT` (`'cx-storefront'`, `core-libs/storefront/layout/main/storefront-outlets.model.ts:L7-L9`) con `cxPageTemplateStyle`;
- l'header è dentro l'outlet `cx-header`: `<header id="cx-header" cxSkipLink="cx-header" [cxFocus]="focusConfig">` con `keydown.escape` → `collapseMenu()` e `click` → `collapseMenuIfClickOutside` (`core-libs/storefront/layout/main/storefront.component.html:L1-L10`); contiene `<cx-page-layout section="header" [cxSkipFocus]="skipFocusConfig" />` e `<cx-page-layout section="navigation" />`, seguiti dallo slot `BottomHeaderSlot` e da `<cx-global-message aria-atomic="true" aria-live="assertive" />` (`core-libs/storefront/layout/main/storefront.component.html:L11-L16`);
- `<main cxSkipLink="cx-main" [cxFocus]="{ disableMouseFocus: true }">` contiene il `<router-outlet />`; il footer è nell'outlet `cx-footer` (`core-libs/storefront/layout/main/storefront.component.html:L19-L27`).

Comportamenti del componente:

1. Stato iniziale: `focusConfig = { disableMouseFocus: true, trap: false }`, `skipFocusConfig = { isEnabled: false, activeElementSelectors: ['button.cx-hamburger'] }` (`core-libs/storefront/layout/main/storefront.component.ts:L68-L74`).
2. Host binding `class.start-navigating`, `class.stop-navigating` e `tabindex = '-1'` ("required by esc focus") (`core-libs/storefront/layout/main/storefront.component.ts:L86-L90`).
3. `keydown.escape` sull'host delega a `KeyboardFocusService.handleEscape` con `{ focusOnEscape: true, focusOnDoubleEscape: true }` (`core-libs/storefront/layout/main/storefront.component.ts:L94-L106`).
4. Il costruttore registra sei feature styles con `useFeatureStyles(...)` (`core-libs/storefront/layout/main/storefront.component.ts:L112-L124`); `useFeatureStyles` richiede un injection context, registra l'uso nel `FeatureStylesService` e lo deregistra su `DestroyRef.onDestroy` (`core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40`).
5. `ngOnInit` sottoscrive `routingService.isNavigating()` (che seleziona `RoutingSelector.isNavigating` dallo store, `core-libs/core/src/routing/facade/routing.service.ts:L79-L81`), ridefinisce `isExpanded$` con un `tap` di focus, chiama `trapFocusOnMenuIfExpanded()` (`core-libs/storefront/layout/main/storefront.component.ts:L126-L139`) e, con `a11yPreventWindowsHighContrastOverride`, aggiunge una classe a `documentElement` (`core-libs/storefront/layout/main/storefront.component.ts:L141-L147`).
6. **Focus dopo la navigazione** (`onNavigation`): a fine navigazione, se l'elemento attivo non è il `body`, sposta il focus sul contenuto di `cx-main` con `SkipLinkService.scrollToTarget`; con

a11yFocusBreadcrumbOnNavigation e una CATEGORY_PAGE prova prima cx-breadcrumb nav a (core-libs/storefront/layout/main/storefront.component.ts:L194-L228, core-libs/storefront/layout/main/storefront.component.ts:L110-L110). Il JSDoc del toggle conferma il fallback su cx-main (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L838-L844).

3.8 Menu hamburger e focus trap su mobile

1. HamburgerMenuService (root) espone isExpanded: BehaviorSubject<boolean> (iniziale false) e collassa il menu a ogni NavigationStart (core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.service.ts:L12-L24); toggle(forceCollapse?) emette false o inverte (core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.service.ts:L29-L35).
2. HamburgerMenuComponent (OnPush) delega al service (core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.component.ts:L13-L29); il template è un <button class="cx-hamburger"> con aria-expanded, aria-label tradotto e aria-controls="cx-header" (core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.component.html:L1-L18). Viene mappato come componente CMS HamburgerMenuComponent (core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.module.ts:L12-L25).
3. Quando il menu si espande, focusOnFirstNavigationItem cerca header cx-navigation-ui e focalizza, in setTimeout, il primo li:not(.back) button, [tabindex="0"] (core-libs/storefront/layout/main/storefront.component.ts:L176-L186).
4. trapFocusOnMenuIfExpanded ricrea focusConfig con trap: isExpanded e skipFocusConfig con isEnabled: isExpanded (core-libs/storefront/layout/main/storefront.component.ts:L164-L174). Sull'header questo attiva il trap di [cxFocus]; sulla sezione header SkipFocusDirective mette tabindex=-1 su tutti i focusabili visibili tranne button.cx-hamburger (core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L38-L67): restano tabbabili il bottone e la navigazione.
5. Chiusura: Escape sull'header, click sull'area header.is-expanded (core-libs/storefront/layout/main/storefront.component.ts:L150-L162) o una nuova navigazione.

3.9 Deferred loading

1. DeferLoaderService calcola globalLoadStrategy = config.deferredLoading?.strategy ?? DeferLoadingStrategy.INSTANT (core-libs/storefront/layout/loading/defer-loader.service.ts:L22-L32). I valori sono DEFER = 'DEFERRED-LOADING' e INSTANT = 'INSTANT-LOADING' (core-libs/core/src/cms/config/cms-config.ts:L96-L100).
2. load(element, options) restituisce of(true) se va caricato subito, altrimenti intersectionService.isIntersected(element, options) (core-libs/storefront/layout/loading/defer-loader.service.ts:L44-L53). "Subito" significa: server, OR strategia dell'elemento INSTANT, OR (elemento non DEFER AND globale INSTANT) (core-libs/storefront/layout/loading/defer-loader.service.ts:L55-L64).
3. IntersectionService.isIntersected in SSR restituisce of(false); in browser emette solo il primo true (core-libs/storefront/layout/loading/intersection.service.ts:L45-L58). L'observable crea un IntersectionObserver nativo per sottoscrizione e lo disconnette al teardown (core-libs/storefront/layout/loading/intersection.service.ts:L103-L118); il rootMargin viene da options.rootMargin, poi da deferredLoading.intersectionMargin (core-libs/storefront/layout/loading/intersection.service.ts:L120-L131).
4. Chi lo usa: OutletDirective.deferLoading() emette loaded(false), attende deferLoaderService.load(...) e poi costruisce l'outlet ed emette loaded(true) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L102-L116); PageSlotService.getComponentDeferOptions forza INSTANT per gli slot che non vanno differiti, altrimenti usa la strategia del componente CMS (core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L63-L73).

3.10 Keyboard focus: la catena di [cxFocus]

Modello. Gli attributi DOM sono data-cx-focus e data-cx-focus-group; MOVE_FOCUS ha NEXT = 1, PREV = -1 (core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L7-L15). Le interfacce di config formano una catena: VisibleFocusConfig (disableMouseFocus) → BlockFocusConfig (block) → PersistFocusConfig (key, group, focusTargetSelector, clearOnRestore) → EscapeFocusConfig (focusOnEscape, focusOnDoubleEscape) → AutoFocusConfig (autofocus, refreshFocus) → TabFocusConfig (tab) → TrapFocusConfig (trap, trapTabOnly) → LockFocusConfig (lock) → FocusConfig (core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L19-L149).

Directive e servizi. FocusDirective ([cxFocus]) estende LockFocusDirective, con defaultConfig = {} e @Input('cxFocus') config, e inietta KeyboardFocusService (core-libs/storefront/layout/a11y/keyboard-focus/focus.directive.ts:L12-L25). Sotto c'è la catena Lock → Trap → Tab → AutoFocus → Escape → Persist → Block → Visible → Base, tutte @Directive() senza selettore (core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L32-L36, core-libs/storefront/layout/a11y/keyboard-focus/visible/visible-focus.directive.ts:L24-L25). In parallelo, KeyboardFocusService extends LockFocusService extends TrapFocusService extends TabFocusService extends AutoFocusService extends EscapeFocusService extends PersistFocusService extends BaseFocusService, tutti providedIn: 'root' (core-libs/storefront/layout/a11y/keyboard-focus/services/keyboard-focus.service.ts:L10-L13, core-libs/storefront/layout/a11y/keyboard-focus/persist/persist-focus.service.ts:L19-L22). Solo FocusDirective, SkipFocusDirective, alcuni tipi, i servizi di services/ e le utils sono pubblici; le directive intermedie sono volutamente non esportate (core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L7-L27) e KeyboardFocusModule dichiara solo le due directive pubbliche (core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.module.ts:L12-L29).

Cosa fa ogni anello:

ANELLO	COMPORTAMENTO	REF
Base	@HostBinding('attr.tabindex'); applica defaultConfig se config vuota; tabindex=-1 solo se l'host non è nativamente focusabile e non ha tabindex	core-libs/storefront/layout/a11y/keyboard-focus/base/base-focus.directive.ts:L43-L53, :L67-L71, :L88-L112
Visible	classe mouse-focus su mousedown, rimossa su keydown di navigazione, solo con disableMouseFocus	core-libs/storefront/layout/a11y/keyboard-focus/visible/visible-focus.directive.ts:L35-L86
Block	tabindex=-1 se block	core-libs/storefront/layout/a11y/keyboard-focus/block/block-focus.directive.ts:L30-L35
Persist	data-cx-focus=key; su focus salva la chiave per gruppo; una config stringa diventa { key }; in ngAfterViewInit ripristina il focus persistito	core-libs/storefront/layout/a11y/keyboard-focus/persist/persist-focus.directive.ts:L69-L131
Escape	su keydown.escape focalizza l'host; al secondo Escape (già sull'host) con focusOnDoubleEscape il primo focusable	core-libs/storefront/layout/a11y/keyboard-focus/escape/escape-focus.directive.ts:L36-L48, core-libs/storefront/layout/a11y/keyboard-focus/escape/escape-focus.service.ts:L24-L42
AutoFocus	in ngAfterViewInit focalizza il primo focusable (persistito → selettore → [autofocus] → primo focusable → host con :host); refreshFocus in ngOnChanges lo rilancia	core-libs/storefront/layout/a11y/keyboard-focus/autofocus/auto-focus.directive.ts:L18-L39, :L61-L95; core-libs/storefront/layout/a11y/keyboard-focus/autofocus/auto-focus.service.ts:L22-L33
Tab	frecce sinistra/destra spostano il focus fra i figli (tab: true, selettore CSS o 'scroll')	core-libs/storefront/layout/a11y/keyboard-focus/tab/tab-focus.directive.ts:L25-L37, core-libs/storefront/layout/a11y/keyboard-focus/tab/tab-focus.service.ts:L18-L35

ANELLO	COMPORTAMENTO	REF
Trap	Tab/freccia giù e Shift+Tab/freccia su ciclano dentro l'host (`trap`), o solo Tab (`trapTabOnly`)	core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.directive.ts:L23-L39, core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.service.ts:L34-L73
Lock	l'host è un unico stop di tabulazione; Enter/Space/click sbloccano i figli; Escape ri-blocca	core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L42-L221

La ricerca dei focusable è in `SelectFocusUtility`: selettori `a[href], button, [tabindex], input, select, textarea` con suffisso `:not([disabled]):not([hidden]):not([aria-hidden])` (core-libs/storefront/layout/a11y/keyboard-focus/services/select-focus.util.ts:L18-L33); senza `locked` esclude `tabindex='-1'` e filtra gli elementi nascosti (core-libs/storefront/layout/a11y/keyboard-focus/services/select-focus.util.ts:L65-L78), dove "nascosto" significa `offsetParent === null` (core-libs/storefront/layout/a11y/keyboard-focus/services/select-focus.util.ts:L80-L92).

Esempio di configurazione reale per un modale: `AddedToCartDialogComponent.focusConfig = { trap: true, block: true, autofocus: 'button', focusOnEscape: true }` (feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog.component.ts:L93-L98).

3.11 Skip link

1. `SkipLinkConfig` (`useExisting: Config`) ha `skipLinks?: SkipLink[]`, dove `SkipLink` ha `key`, `i18nKey`, `target?`, `position?` (core-libs/storefront/layout/a11y/skip-link/config/skip-link.config.ts:L10-L32). Il default definisce `cx-header`, `cx-main`, `cx-footer` (core-libs/storefront/layout/a11y/skip-link/config/default-skip-link.config.ts:L9-L21).
2. `SkipLinkModule` registra `defaultSkipLinkConfig` sia come config normale (`ConfigModule.withConfig`, che equivale a `provideConfig`) sia come default (`provideDefaultConfig`), quindi per personalizzare `skipLinks` serve `provideConfig` (core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L29-L36, core-libs/core/src/config/config.module.ts:L28-L33); registra inoltre un `APP_INITIALIZER` con `skipLinkFactory` (core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L25-L45), che aggiunge la factory di `SkipLinkComponent` all'outlet `'cx-storefront'` in posizione `BEFORE` (core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L47-L60): così la skip link è il primo contenuto renderizzato prima della shell.
3. `[cxSkipLink]` registra l'elemento host nel service in `ngOnInit` e lo rimuove in `ngOnDestroy` (core-libs/storefront/layout/a11y/skip-link/directive/skip-link.directive.ts:L10-L26). `SkipLinkService.add` accetta solo chiavi presenti in config e le inserisce in ordine di config (core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L27-L42, core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L93-L111).
4. `SkipLinkComponent` (OnPush) rende un `<button>` per link dentro un `div [cxFocus]="{ tab: true }"`, quindi navigabile con le frecce (core-libs/storefront/layout/a11y/skip-link/component/skip-link.component.ts:L15-L29, core-libs/storefront/layout/a11y/skip-link/component/skip-link.component.html:L1-L6).
5. Il click chiama `scrollToTarget`, che focalizza il primo focusable del target (via `KeyboardFocusService.findFirstFocusable`) o il target, con `tabindex="-1"` temporaneo (core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L56-L87).

3.12 Altre directive a11y

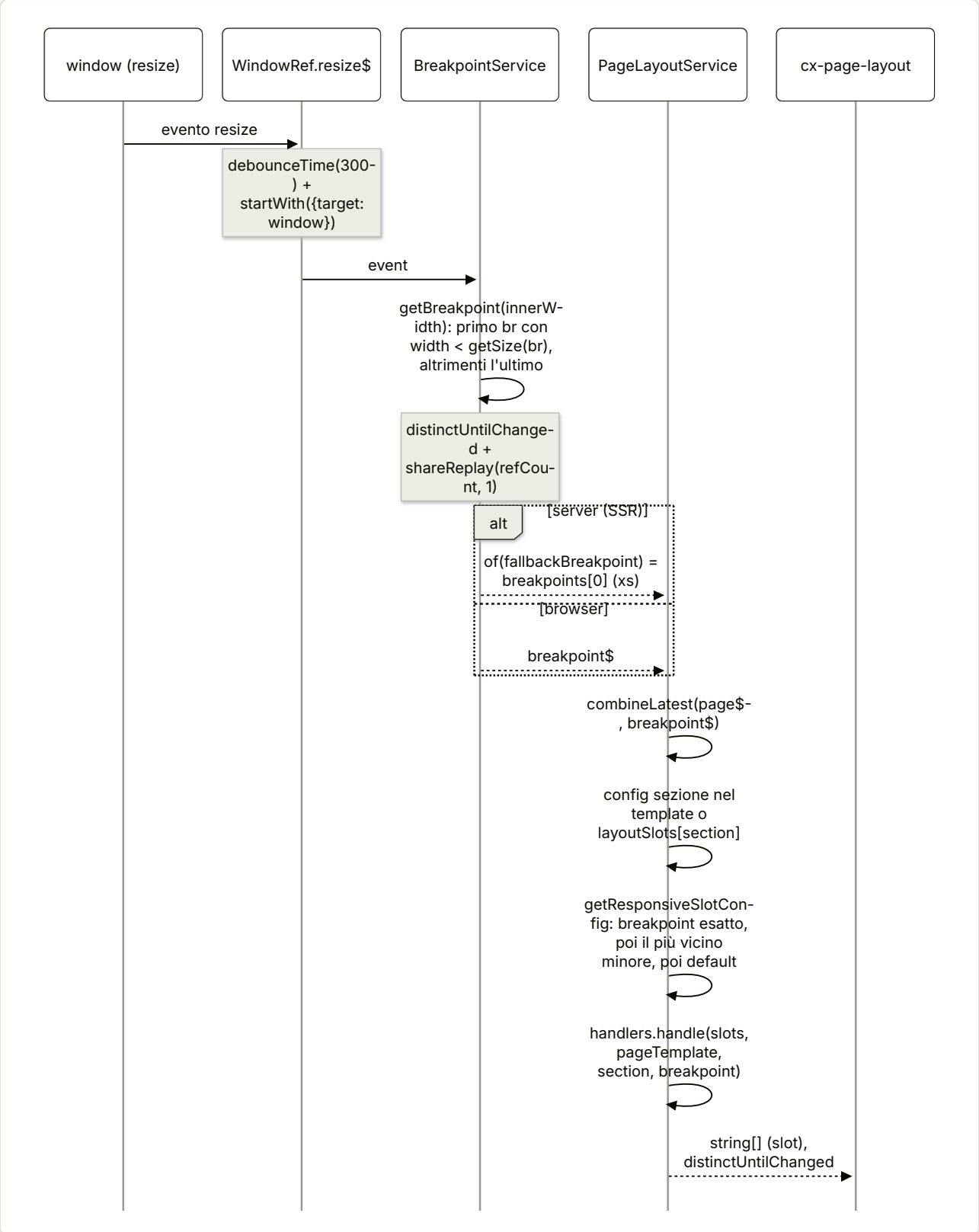
- `BtnLikeLinkDirective` (`a[cxBtnLikeLink].btn`): host `tabindex="0"`, `role="button"`, Enter/Space → `click()` (core-libs/storefront/layout/a11y/btn-like-link/btn-like-link.directive.ts:L18-L36); il JSDoc la de-

finisce un "code smell" da usare solo per vecchi componenti (core-libs/storefront/layout/a11y/btn-like-link/btn-like-link.directive.ts:L9-L17).

- FocusFirstInvalidFieldDirective ([cxFocusFirstInvalidField]): focusFirstInvalidField() focalizza in setTimeout il primo campo .ng-invalid (core-libs/storefront/layout/a11y/focus-first-invalid-field/focus-first-invalid-field.directive.ts:L30-L53); il timeout serve a passare dopo l'autofocus di un host cxFocus (caso Safari) (core-libs/storefront/layout/a11y/focus-first-invalid-field/focus-first-invalid-field.directive.ts:L18-L23).
- NativeSelectSpaceDirective (select[cxNativeSelectSpace]), con a11yNavigationSpaceKeyOnKeyUp, sposta l'azione della barra spaziatrice sul keyup (WCAG 2.5.2) (core-libs/storefront/layout/a11y/native-select-space/native-select-space.directive.ts:L22-L49).
- DomChangeDirective ([cxDomChange]) emette le mutation di un MutationObserver su subtree/childList (core-libs/storefront/layout/a11y/on-dom-change/dom-change.directive.ts:L28-L66).
- TruncationTooltipDirective ([cxTruncationTooltip]) mostra uno span aria-hidden appeso al body se il testo è troncato (core-libs/storefront/layout/a11y/truncation-tooltip/truncation-tooltip.directive.ts:L17-L56).

4. Diagrammi

4.1 Risoluzione del breakpoint e degli slot



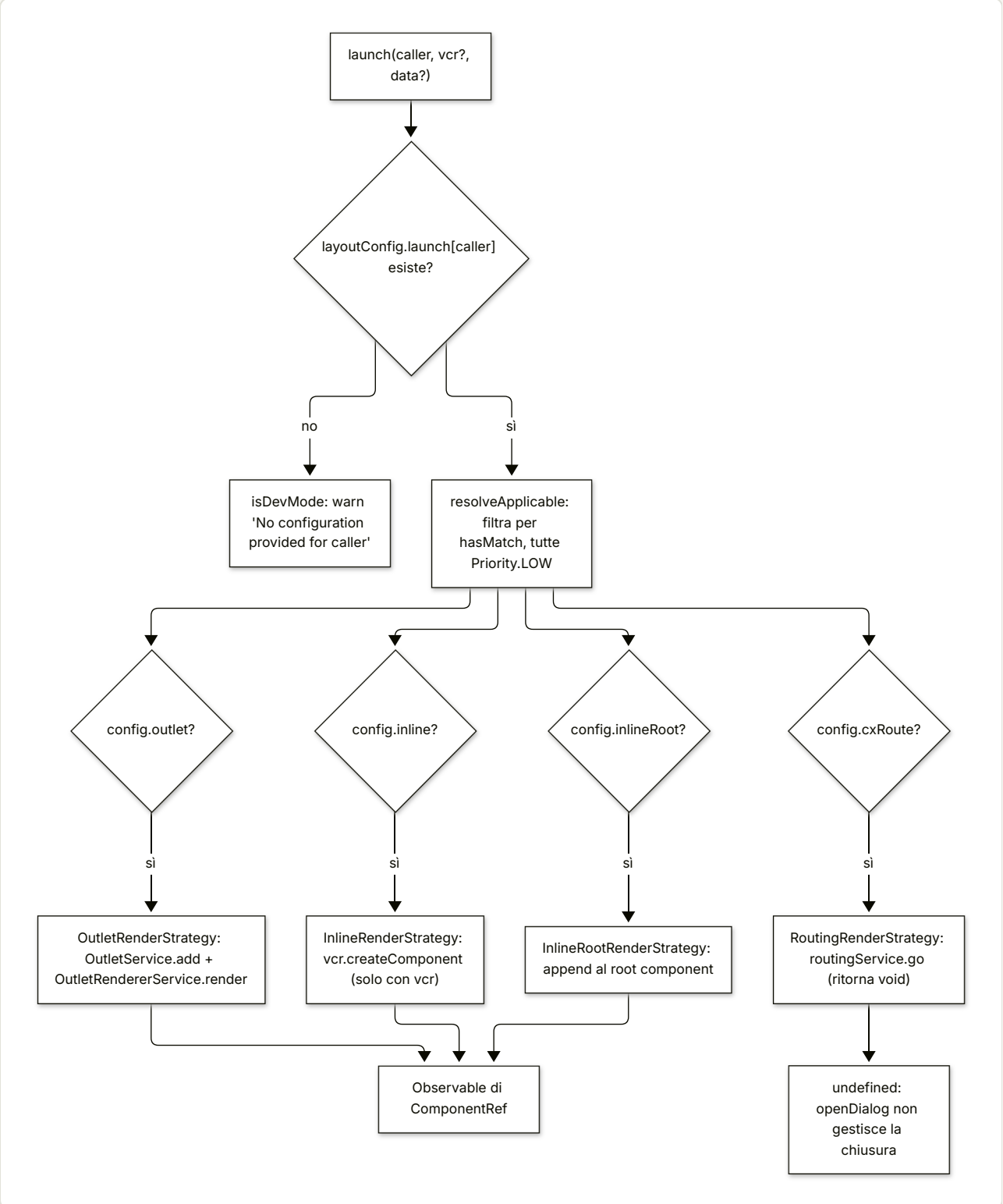
Riferimenti: core-libs/core/src/window/window-ref.ts:L91-L107, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L40-L46, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L192-L199, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L123-L131, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L55-L59, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L187-L191, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243.

4.2 Ciclo di vita di un launch dialog (inlineRoot)

■ **TAVOLA 18.1** diagramma a tutta pagina, in fondo al capitolo

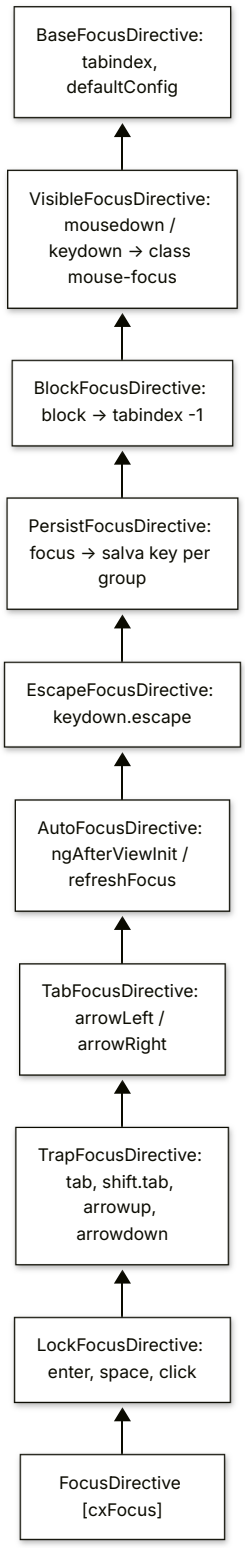
Riferimenti: feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog-event.listener.ts:L57-L67, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L52-L97, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L121-L185, core-libs/core/src/util/applicable.ts:L47-L74, core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L36-L64, core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L92-L153.

4.3 Scelta della strategia di rendering



Se più `hasMatch` sono veri vince il provider registrato per ultimo (ordine: Outlet, Inline, Routing, InlineRoot). Riferimenti: `core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L18-L41`, `core-libs/core/src/util/applicable.ts:L47-L74`, `core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L45-L84`, `core-libs/storefront/layout/launch-dialog/services/inline-render.strategy.ts:L42-L75`, `core-libs/storefront/layout/launch-dialog/services/routing-render.strategy.ts:L22-L32`, `core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L36-L70`.

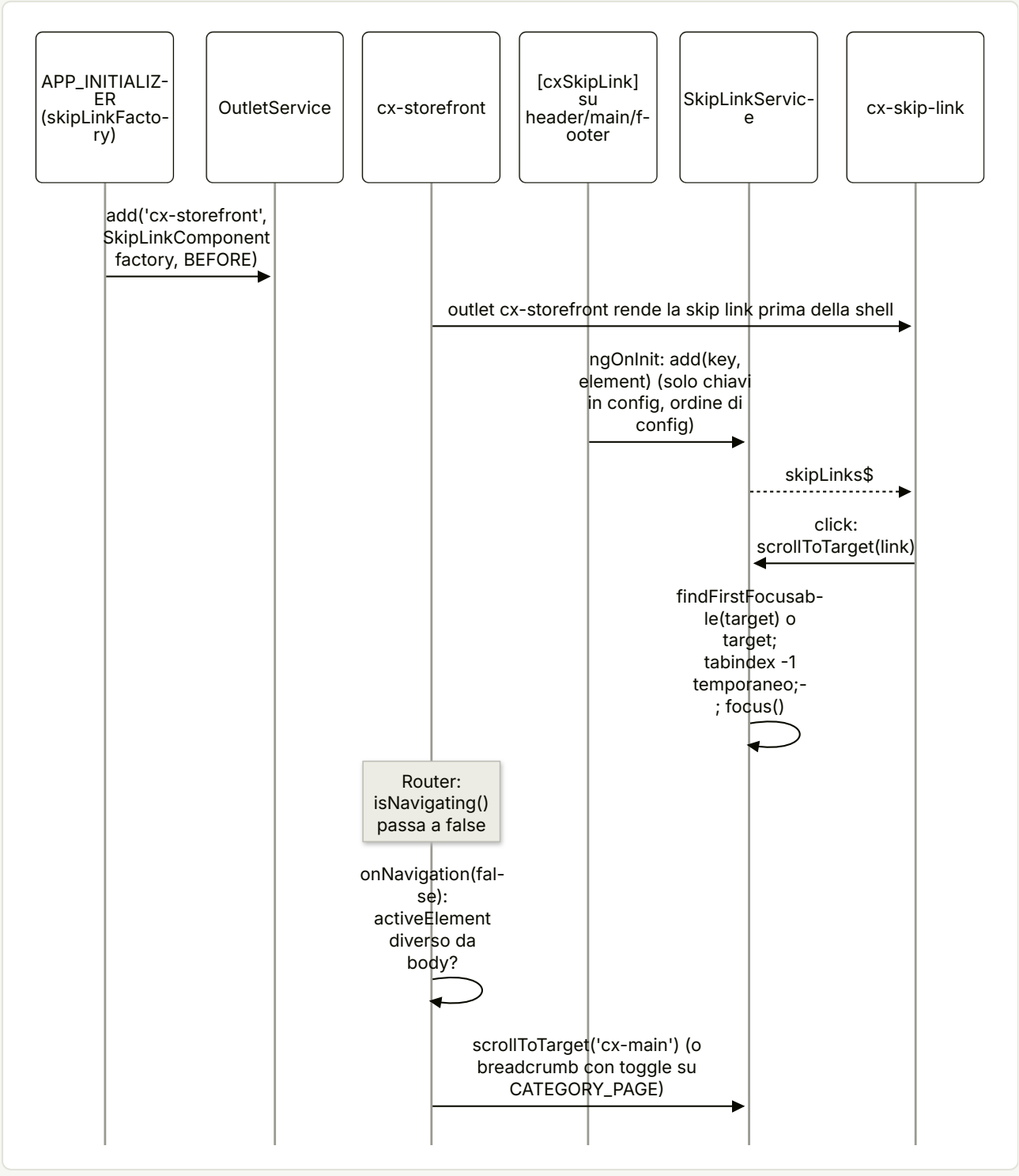
4.4 Gerarchia di `[cxFocus]` ed eventi gestiti



Riferimenti: core-libs/storefront/layout/a11y/keyboard-focus/focus.directive.ts:L12-L25, core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L32-L36, core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.directive.ts:L16-L17, core-libs/storefront/layout/a11y/keyboard-focus/tab/tab-focus.directive.ts:L17-L18, core-libs/storefront/layout/a11y/keyboard-focus/

autofocus/auto-focus.directive.ts:L40-L44, core-libs/storefront/layout/a11y/keyboard-focus/escape/escape-focus.directive.ts:L26-L30, core-libs/storefront/layout/a11y/keyboard-focus/persist/persist-focus.directive.ts:L47-L51, core-libs/storefront/layout/a11y/keyboard-focus/block/block-focus.directive.ts:L12-L17, core-libs/storefront/layout/a11y/keyboard-focus/visible/visible-focus.directive.ts:L24-L25.

4.5 Skip link e focus dopo la navigazione



Riferimenti: core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L25-L60, core-libs/storefront/layout/main/storefront.component.html:L1-L1, core-libs/storefront/layout/a11y/skip-link/directive/skip-link.directive.ts:L10-L26, core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L27-L111, core-libs/storefront/layout/a11y/skip-link/component/skip-link.component.html:L1-L6, core-libs/storefront/layout/main/storefront.component.ts:L194-L228.

---

5. Estratti di codice reale

breakpoint$ — core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L40-L46

```ts
breakpoint$: Observable<BREAKPOINT> = isPlatformBrowser(this.platform)
  ? this.winRef.resize$.pipe(
      map((event) => this.getBreakpoint((<Window>event.target).innerWidth)),
      distinctUntilChanged(),
      shareReplay({ refCount: true, bufferSize: 1 })
    )
  : of(this.fallbackBreakpoint);
```

Ordinamento dei breakpoint — core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L138-L153

```ts
protected resolveBreakpointsFromConfig(): BREAKPOINT[] {
  const sortByScreenSize = (next: BREAKPOINT, prev: BREAKPOINT): number => {
    const nextMinSize = this.getMinSize(next);
    const maxNext = Math.max(
      nextMinSize ? nextMinSize + 1 : 0,
      this.getMaxSize(next) || 0
    );
    const preMinSize = this.getMinSize(prev);
    const maxPrev = Math.max(
      preMinSize ? preMinSize + 1 : 0,
      this.getMaxSize(prev) || 0
    );
    return maxNext < maxPrev ? -1 : 0;
  };
  return (Object.keys(this.config) as BREAKPOINT[]).sort(sortByScreenSize);
}
```

Config responsiva degli slot — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243

```ts
protected getResponsiveSlotConfig(
  layoutSlotConfig: LayoutSlotConfig,
  configAttribute: string,
  breakpoint?: BREAKPOINT
): SlotConfig {
  let slotConfig = <SlotConfig>layoutSlotConfig;
```

```
  // fallback to default slot config
  if (!layoutSlotConfig || !breakpoint) {
    return slotConfig;
  }

  // we have a config for the specific breakpoint
  if (layoutSlotConfig[breakpoint]?.hasOwnProperty(configAttribute)) {
    return <SlotConfig>layoutSlotConfig[breakpoint];
  }

  // find closest config
  const all = this.breakpointService.breakpoints;

  for (const br of all.slice(0, all.indexOf(breakpoint))) {
    if (layoutSlotConfig[br]?.hasOwnProperty(configAttribute)) {
      slotConfig = <SlotConfig>layoutSlotConfig[br];
    }
  }
  return slotConfig;
}
```

openDialog **e** launch — core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L52-L97

```ts
openDialog(
  caller: LAUNCH_CALLER | string,
  openElement?: ElementRef,
  vcr?: ViewContainerRef,
  data?: any
): Observable<any> | undefined {
  const component = this.launch(caller, vcr, data);

  if (component) {
    return combineLatest([component, this.dialogClose]).pipe(
      filter(([, close]) => close !== undefined),
      tap(([comp]) => {
        this.focusElement(openElement?.nativeElement);
        this.clear(caller);
        comp?.destroy();
      }),
      map(([comp]) => comp)
    );
  }
}
// ...
launch(
  caller: LAUNCH_CALLER | string,
```

```
    vcr?: ViewContainerRef,
    data?: any
  ): void | Observable<ComponentRef<any> | undefined> {
    const config = this.findConfiguration(caller);
    if (config) {
      const renderer = this.getStrategy(config);

      // Render if the strategy exists
      if (renderer) {
        this._dialogClose.next(undefined);
        this._dataSubject.next(data);

        return renderer.render(config, caller, vcr);
      }
    } else if (isDevMode()) {
      this.logger.warn('No configuration provided for caller ' + caller);
    }
  }
```

shouldRender — core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L92-L102

```ts
  protected shouldRender(
    caller: LAUNCH_CALLER | string,
    config: LaunchDialog
  ): boolean {
    return (
      Boolean(config.component) &&
      (this.renderedCallers.some((el) => el.caller === caller)
        ? !!config.multi
        : true)
    );
  }
```

Render inline-root — core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L40-L64

```ts
    if (this.shouldRender(caller, config)) {
      const componentFactory =
        this.componentFactoryResolver.resolveComponentFactory(config.component);

      const contentInjector = Injector.create({
        providers: [],
      });

      const componentRef = componentFactory.create(contentInjector);

      this.injector.get(ApplicationRef)?.attachView(componentRef.hostView);
```

```
      this.renderer.appendChild(
        this.hostComponent?.location.nativeElement,
        componentRef.location.nativeElement
      );

      if (config?.dialogType) {
        this.applyClasses(componentRef, config?.dialogType);
      }

      this.renderedCallers.push({ caller, component: componentRef });

      return of(componentRef);
    }
```

Shell dello storefront — core-libs/storefront/layout/main/storefront.component.html:L1-L28

```HTML
<ng-template [cxOutlet]="StorefrontOutlets.STOREFRONT" cxPageTemplateStyle>
  <ng-template cxOutlet="cx-header">
    <header
      id="cx-header"
      cxSkipLink="cx-header"
      [cxFocus]="focusConfig"
      [class.is-expanded]="isExpanded$ | async"
      (keydown.escape)="collapseMenu()"
      (click)="collapseMenuIfClickOutside($event)"
    >
      <cx-page-layout section="header" [cxSkipFocus]="skipFocusConfig" />

      <cx-page-layout section="navigation" />
    </header>
    <cx-page-slot position="BottomHeaderSlot" class="cx-bottom-header-slot" />
    <cx-global-message aria-atomic="true" aria-live="assertive" />
  </ng-template>

  <main cxSkipLink="cx-main" [cxFocus]="{ disableMouseFocus: true }">
    <router-outlet />
  </main>

  <ng-template cxOutlet="cx-footer">
    <footer cxSkipLink="cx-footer" [cxFocus]="{ disableMouseFocus: true }">
      <cx-page-layout section="footer" />
    </footer>
  </ng-template>
</ng-template>
```

Focus dopo la navigazione — core-libs/storefront/layout/main/storefront.component.ts:L194-L202, :L224-L228

```ts
protected onNavigation(isNavigating: boolean): void {
  this.startNavigating = isNavigating === true;
  this.stopNavigating = isNavigating === false;

  // After clicking a link the focus should move to the first available item in the main content area.
  if (
    this.stopNavigating &&
    this.document?.activeElement !== this.document?.body
  ) {
    // ... ramo a11yFocusBreadcrumbOnNavigation
    } else {
      this.skipLinkService?.scrollToTarget('cx-main');
    }
  }
}
```

Focus trap — core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.service.ts:L34-L73

```ts
moveFocus(
  host: HTMLElement,
  config: TrapFocusConfig,
  increment: MOVE_FOCUS,
  event: UIEvent
): void {
  const focusable: HTMLElement[] = this.findFocusable(host);

  let index = focusable.findIndex((v) => v === event.target) + increment;
  const trap = config.trap || config.trapTabOnly;

  const shouldMoveFocus =
    (index >= 0 && index < focusable.length) ||
    (index < 0 && this.getTrapStart(trap)) ||
    (index >= focusable.length && this.getTrapEnd(trap));

  if (shouldMoveFocus) {
    if (index >= focusable.length) {
      index = 0;
    }
    if (index < 0) {
      index = focusable.length - 1;
    }

    event.preventDefault();
    event.stopPropagation();

    const el = focusable[index];

    el.focus();
```

```
    }
  }

  protected getTrapStart(trap: TrapFocusType | undefined): boolean {
    return trap === true || trap === TrapFocus.start;
  }

  protected getTrapEnd(trap: TrapFocusType | undefined): boolean {
    return trap === true || trap === TrapFocus.end;
  }
```

Lock: default impliciti — core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L109-L129

```ts
  ngOnInit() {
    super.ngOnInit();

    this.shouldLock = this.config?.lock;

    if (this.shouldLock) {
      this.tabindex = 0;

      // Locked elements will be set to `autofocus` by default if it's not
      // been configured. This will ensure that autofocus kicks in upon unlock.
      if (!this.config.hasOwnProperty('autofocus')) {
        this.config.autofocus = true;
      }
      // Locked elements will be set to `focusOnEscape` by default if it's not
      // been configured. This will ensure that  the host gets locked again when
      // `escape` is pressed.
      if (!this.config.hasOwnProperty('focusOnEscape')) {
        this.config.focusOnEscape = !(this.config?.focusOnEscape === false);
      }
    }
  }
```

Skip link nell'outlet — core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L47-L60

```ts
/**
 * Adds the skip link component before the cx-storefront.
 */
export function skipLinkFactory(
  componentFactoryResolver: ComponentFactoryResolver,
  outletService: OutletService
): () => void {
  const isReady = () => {
    const factory =
      componentFactoryResolver.resolveComponentFactory(SkipLinkComponent);
```

```
    outletService.add('cx-storefront', <any>factory, OutletPosition.BEFORE);
  };
  return isReady;
}
```

Decisione di caricamento immediato — core-libs/storefront/layout/loading/defer-loader.service.ts:L55-L64

```ts
  private shouldLoadInstantly(
    elementLoadingStrategy: DeferLoadingStrategy | undefined
  ): boolean {
    return (
      isPlatformServer(this.platformId) ||
      elementLoadingStrategy === DeferLoadingStrategy.INSTANT ||
      (elementLoadingStrategy !== DeferLoadingStrategy.DEFER &&
        this.globalLoadStrategy === DeferLoadingStrategy.INSTANT)
    );
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
breakpoints	LayoutBreakPoints (number = max, oppure {min?, max?})	xs:576, sm:768, md:992, lg:1200, xl:{min:1200}	core-libs/storefront/layout/config/layout-config.ts:L100-L100; default core-libs/storefront/layout/config/default-layout.config.ts:L10-L18	core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L204-L206
layoutSlots	LayoutSlotConfig	nessuno in defaultLayoutConfig; recipe layoutConfig / layoutConfigFactory	core-libs/storefront/layout/config/layout-config.ts:L102-L102; recipe core-libs/storefront/recipes/config/layout-config.ts:L27-L134	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L146-L210

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
layoutSlots.<x>.slots	string[]	—	core-libs/storefront/layout/config/layout-config.ts:L38-L38	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243
layoutSlots.<x>.pageFold	string	recipe Section2B/Section2/Summary (lg: UpSelling), rimossi dalla factory (core-libs/storefront/recipes/config/layout-config.ts:L172-L185)	core-libs/storefront/layout/config/layout-config.ts:L45-L45	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L89-L101
deferredLoading.strategy	DeferLoadingStrategy	INSTANT (fallback nel service)	core-libs/storefront/layout/config/layout-config.ts:L114-L114	core-libs/storefront/layout/loading/defer-loader.service.ts:L30-L31
deferredLoading.intersectionMargin	string	undefined	core-libs/storefront/layout/config/layout-config.ts:L121-L121	core-libs/storefront/layout/loading/intersection.service.ts:L124-L130
launch	LaunchConfig ({[caller]: LaunchOptions})	nessuno nel layout core; contributi delle feature (es. ADDED_TO_CART, ASM)	core-libs/storefront/layout/config/layout-config.ts:L124-L124; tipi core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L9-L76	core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L167-L174
launch.<c>.component / multi / dialogType	any / boolean / DIALOG_TYPE	multi falsy	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L22-L32	core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L92-L136

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
launch.<c>.outlet / position	string / OutletPosition	position BEFORE	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L38-L47	core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L53-L57
launch.<c>.inline	boolean	—	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L52-L54	core-libs/storefront/layout/launch-dialog/services/inline-render.strategy.ts:L73-L75
launch.<c>.inlineRoot	boolean	—	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L60-L62	core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L67-L69
launch.<c>.cxRoute / params	string / object	—	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L67-L76	core-libs/storefront/layout/launch-dialog/services/routing-render.strategy.ts:L25-L31
skipLinks	SkipLink[]	header / main / footer	core-libs/storefront/layout/a11y/skip-link/config/skip-link.config.ts:L14-L16; default core-libs/storefront/layout/a11y/skip-link/config/default-skip-link.config.ts:L9-L21	core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L28-L30, core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L94-L99
FocusConfig (input [cxFocus])	oggetto o stringa (= key)	{}	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L19-L149	catena di directive (sez. 3.10)

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
SkipFocusConfig (input [cxSkipFocus])	{isEnabled, activeElementSelectors?}	{ isEnabled: false }	core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L17-L24, core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L32-L32	core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L38-L43
toggle a11yPreventWindowsHighContrastOverride	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1026-L1026	core-libs/storefront/layout/main/storefront.component.ts:L143-L147
toggle a11yFocusBreadcrumbOnNavigation	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1105-L1105	core-libs/storefront/layout/main/storefront.component.ts:L203-L203
toggle a11yConsentManagementFocusPreservation	boolean	true	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1059-L1059	core-libs/storefront/layout/a11y/keyboard-focus/visible/visible-focus.directive.ts:L74-L74
toggle a11yRestoreFocusOnNgSelect	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1065-L1065	core-libs/storefront/layout/a11y/keyboard-focus/persist/persist-focus.directive.ts:L115-L115

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `a11yNavigationSpaceKeyOnKeyUp`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1093-L1093`	`core-libs/storefront/layout/a11y/native-select-space/native-select-space.directive.ts:L32-L32`, `core-libs/storefront/layout/a11y/native-select-space/native-select-space.directive.ts:L43-L43`

7. Commenti e note notevoli dal sorgente

- `// TODO: make classes configurable` — le classi CSS dei dialog sono hard-coded (`core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L110-L110`).
- `// this is going to fail as we have sub services. They will al have their own map. // We must bring this to a singleton map.` — gli autori sanno che ogni servizio della catena ha la propria `Map` di focus persistito (`core-libs/storefront/layout/a11y/keyboard-focus/persist/persist-focus.service.ts:L23-L24`).
- `// given that we're likely going to refactor the directives, we're // NOT exposing all it to the public API.` (`core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L7-L8`).
- La recipe `layoutConfig` è `@deprecated` a favore di `layoutConfigFactory` + `provideConfigFactory(layoutConfigFactory)` (`core-libs/storefront/recipes/config/layout-config.ts:L25-L25`), e la factory rimuove `pageFold` "to avoid CLS (Cumulative Layout Shift) regressions" (`core-libs/storefront/recipes/config/layout-config.ts:L158-L171`).
- JSDoc di `LaunchOutletDialog`: "Outlet rendering should only be used for elements that open once and do not close." (`core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L34-L37`).
- JSDoc di `BREAKPOINT`: cambiando dimensioni o nomi dei breakpoint "you might loose out on these configurations" (page layout e tabelle preconfigurate) (`core-libs/storefront/layout/config/layout-config.ts:L17-L20`).
- `// required by esc focus` sul `tabindex = '-1'` di `cx-storefront` (`core-libs/storefront/layout/main/storefront.component.ts:L89-L90`).
- `// TODO: Required to use feature flag with root styles.` sul blocco `a11yPreventWindowsHighContrastOverride` (`core-libs/storefront/layout/main/storefront.component.ts:L141-L142`).
- `// we wait a few milliseconds, mainly because firefox will otherwise apply the mouse event on the new focused child element` (`core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L101-L102`) e `// Delay unlocking in case the host is using ChangeDetectionStrategy.Default` (`core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L156-L156`).
- JSDoc di `disableTabbingForTick`: `requestAnimationFrame` "may seem like a bad code smell", giustificato dall'interazione con `TrapFocusDirective` (`core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.utils.ts:L7-L32`).

- `SelectFocusUtility` esclude volutamente `[contentEditable=true]`, `iframe`, `area[href]` (core-libs/storefront/layout/a11y/keyboard-focus/services/select-focus.util.ts:L27-L30) e per performance non usa computed style in `isHidden` (core-libs/storefront/layout/a11y/keyboard-focus/services/select-focus.util.ts:L80-L92).
- JSDoc di `VisibleFocusDirective`: il bordo di focus "can be considered annoying by keyboard users" — testuale; il senso inteso sembra riferito agli utenti mouse (core-libs/storefront/layout/a11y/keyboard-focus/visible/visible-focus.directive.ts:L19-L22).
- JSDoc di `NativeSelectSpaceDirective`: "Fixes WCAG 2.5.2 (Level A)" (core-libs/storefront/layout/a11y/native-select-space/native-select-space.directive.ts:L10-L21).
- "Early return for SSR - element is never intersected on server" (core-libs/storefront/layout/loading/intersection.service.ts:L50-L50).
- Angular 21 depreca `ComponentFactoryResolver` (angular:packages/core/src/linker/component_factory_resolver.ts:L38-L41), la firma `createComponent(componentFactory)` (angular:packages/core/src/linker/view_container_ref.ts:L263-L267) e `APP_INITIALIZER` ("use provideAppInitializer instead", angular:packages/core/src/application/application_init.ts:L38-L38): tutte usate in questo capitolo.

8. Extension point

EXTENSION POINT	COME SI USA	DEFINIZIONE	EXPORT PUBBLICO
`LayoutConfig.breakpoints`	`provideConfig({ breakpoints: {...} })` per cambiare soglie o aggiungere breakpoint	core-libs/storefront/layout/config/layout-config.ts:L100-L100	core-libs/storefront/layout/config/index.ts:L8-L8
`LayoutConfig.layoutSlots`	slot per sezione / page template / breakpoint; base consigliata `provideConfigFactory(layoutConfigFactory)`	core-libs/storefront/layout/config/layout-config.ts:L102-L102; core-libs/storefront/recipes/config/layout-config.ts:L190-L198	core-libs/storefront/public_api.ts:L27-L27
`BREAKPOINT` augmentabile	module augmentation + valore runtime (come `LAUNCH_CALLER`)	core-libs/storefront/layout/config/layout-config.ts:L22-L28	core-libs/storefront/public_api.ts:L34-L34
`BreakpointService`	sostituzione via DI; metodi `protected` (`resolveBreakpointsFromConfig`, `getBreakpoint`, `fallbackBreakpoint`)	core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L37-L207	core-libs/storefront/layout/breakpoint/index.ts:L7-L7
`LAUNCH_CALLER` + `LayoutConfig.launch`	nuovo caller (augmentation + valore runtime) e config `{ inlineRoot / inline / outlet / cxRoute, component, dialogType }`	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L9-L103; esempio feature-libs/cart/base/root/models/augmented.model.ts:L7-L16	core-libs/storefront/public_api.ts:L35-L35

EXTENSION POINT	COME SI USA	DEFINIZIONE	EXPORT PUBBLICO
`LaunchRenderStrategy`	nuova strategia: estendere la classe astratta, implementare `render`/`hasMatch` (eventualmente `getPriority` > LOW) e registrarla come provider multi	core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L24-L158; core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L18-L41	core-libs/storefront/layout/launch-dialog/services/index.ts:L10-L10
`LaunchDialogService`	API per aprire/chiudere dialog e passare dati	core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L24-L186	core-libs/storefront/layout/launch-dialog/services/index.ts:L9-L9
Outlet `cx-storefront`, `cx-header`, `cx-footer`	sostituire o arricchire la shell (cap. 17)	core-libs/storefront/layout/main/storefront.component.html:L1-L2, core-libs/storefront/layout/main/storefront.component.html:L23-L23	core-libs/storefront/layout/main/index.ts:L7-L7
`SkipLinkConfig.skipLinks` + `[cxSkipLink]`	nuove skip link: aggiungere la chiave in config e la directive sul target	core-libs/storefront/layout/a11y/skip-link/config/skip-link.config.ts:L10-L32; core-libs/storefront/layout/a11y/skip-link/directive/skip-link.directive.ts:L10-L26	core-libs/storefront/layout/a11y/skip-link/index.ts:L7-L11
`[cxFocus]` / `FocusConfig`	comportamento da tastiera di dialog, menu, liste	core-libs/storefront/layout/a11y/keyboard-focus/focus.directive.ts:L12-L25	core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L9-L10
`KeyboardFocusService` / `SelectFocusUtility`	focus programmatico (`findFirstFocusable`, `findFocusable`)	core-libs/storefront/layout/a11y/keyboard-focus/services/keyboard-focus.service.ts:L10-L13	core-libs/storefront/layout/a11y/keyboard-focus/services/index.ts:L7-L8
`KeyboardFocusTestingModule`	mock di `[cxFocus]` negli unit test	core-libs/storefront/layout/a11y/keyboard-focus/focus-testing.module.ts:L10-L19	core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L13-L13
`LayoutConfig.deferredLoading` / `DeferLoaderService`	strategia globale e margine di intersezione	core-libs/storefront/layout/config/layout-config.ts:L114-L121	core-libs/storefront/layout/loading/index.ts:L7-L9

Non pubblici (non esportati da `core-libs/storefront/layout/a11y/keyboard-focus/index.ts`): `AutoFocusService`, `PersistFocusService`, `EscapeFocusService`, `TabFocusService`, `TrapFocusService`, `LockFocusService`, `BaseFocusService` e le directive intermedie (core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L17-L27). Non vanno estese né importate da path interni (vedi 54 — API pubblica vs interna).

Ricette pertinenti in Parte III:

- 51 — Ricette: componenti CMS, outlet, route e layout: page layout custom con `LayoutConfig.layoutSlots`, uso degli outlet della shell.
- 50 — Ricette: configurazione, DI, interceptor, eventi: override di config (`provideConfig`, `provideConfigFactory`) e sostituzione di servizi come `BreakpointService` o di una strategia.
- 54 — API pubblica vs interna, upgrade, anti-pattern, checklist: perché non estendere le directive di focus non esportate e cosa cambia con le deprecazioni Angular citate.
- 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni: il dialog "aggiunto al carrello" come esempio di launch dialog di una feature lib.

Schema illustrativo (non presente nel repo; usa solo simboli esportati) di un nuovo dialog globale, sul modello di `ADDED_TO_CART` (feature-libs/cart/base/root/models/augmented.model.ts:L7-L16, feature-libs/cart/base/components/added-to-cart-dialog/default-added-to-cart-layout.config.ts:L10-L18):

```ts
// esempio illustrativo
declare module '@spartacus/storefront' {
  enum LAUNCH_CALLER {
    MY_DIALOG = 'MY_DIALOG',
  }
}
(LAUNCH_CALLER as any)['MY_DIALOG'] = 'MY_DIALOG';

provideDefaultConfig(<LayoutConfig>{
  launch: {
    MY_DIALOG: {
      inlineRoot: true,
      component: MyDialogComponent,
      dialogType: DIALOG_TYPE.DIALOG,
    },
  },
});

// apertura e chiusura
launchDialogService.openDialogAndSubscribe(LAUNCH_CALLER.MY_DIALOG, triggerRef, data);
launchDialogService.closeDialog('done');
```

9. Trappole, casi limite, comportamenti controintuitivi

1. `defaultLayoutConfig` **non ha slot.** Senza la recipe (`layoutConfig` o `layoutConfigFactory`) le sezioni (`header`, `navigation`, `footer`) restano vuote, mentre il corpo della pagina mostra tutti gli slot CMS della pagina, nell'ordine delle chiavi di `page.slots`, con un warning di config mancante (core-libs/

storefront/layout/config/default-layout.config.ts:L9-L19, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L117-L126).

2. **SSR è sempre** `xs`. Lato server `breakpoint$` è il breakpoint più piccolo, quindi l'HTML server usa la config "mobile" (con la recipe deprecata, header a 4 slot) e il client può cambiare layout dopo l'hydration (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L46-L46, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L129-L131, core-libs/storefront/recipes/config/layout-config.ts:L33-L47). La factory (`applyUnifiedHeaderSlots`) toglie la config `lg` dell'header e usa gli 8 slot per tutti i breakpoint, quindi HTML SSR e client hanno lo stesso header (core-libs/storefront/recipes/config/layout-config.ts:L136-L145; cap. 39).
3. `xl` **non esiste in** `SlotGroup`, ma su `xl` vale la config del breakpoint minore più vicino (es. `lg`) (core-libs/storefront/layout/config/layout-config.ts:L48-L57, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L234-L241).
4. **Comparatore non coerente.** `resolveBreakpointsFromConfig` restituisce solo `-1` o `0`, mai `1` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L139-L152). **DA VERIFICARE**: l'ordine risultante su configurazioni arbitrarie dipende dall'algoritmo di sort del motore JS; lo spec copre alcuni casi (core-libs/storefront/layout/breakpoint/breakpoint.service.spec.ts:L48-L142) ma non è stato possibile eseguire test (node_modules assente). Dichiarare le chiavi in ordine crescente è la scelta prudente.
5. **Numero = max.** `xs: 576` significa "fino a 575": un valore numerico è sempre un max (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L161-L176). Il JSDoc di `BreakPoint.max` dice erroneamente "The minimum screen width" (core-libs/storefront/layout/config/layout-config.ts:L69-L72).
6. **Field initializer e costruttore.** `breakpoint$` legge `this.platform`/`this.winRef` durante l'inizializzazione dei campi: una sottoclasse che cambia il costruttore deve preservare quelle parameter properties (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L40-L52).
7. **Un solo** `_dialogClose` **per tutti.** `closeDialog` chiude logicamente tutti i dialog aperti in parallelo con `openDialog`, e ogni `launch` resetta lo stato a `undefined` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L26-L26, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L89-L89, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L138-L140).
8. **Senza subscribe niente cleanup.** Se nessuno sottoscrive l'observable di `openDialog`, focus, `clear` e `destroy` non avvengono mai (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L61-L69); il caller resta in `renderedCallers` e, senza `multi`, il dialog non si riapre (core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L92-L102, core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L145-L148).
9. **Routing restituisce** `void`: con una config `cxRoute`, `openDialog` restituisce `undefined` (core-libs/storefront/layout/launch-dialog/services/routing-render.strategy.ts:L25-L27, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L60-L70).
10. `openDialogAndSubscribe` **non funziona con** `inline`: passa sempre `vcr = undefined`, e la strategia inline non renderizza senza `ViewContainerRef` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L111-L111, core-libs/storefront/layout/launch-dialog/services/inline-render.strategy.ts:L47-L48).
11. **"inside the cx-storefront" non è letterale.** `InlineRootRenderStrategy` appende al primo componente bootstrappato, che nella demo è `app-root`, genitore di `cx-storefront` (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L56-L59, core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L32-L34, projects/storefrontapp/src/app/app.component.ts:L15-L16).
12. **Injector vuoto, ma i servizi root funzionano.** Il componente inline-root è creato con `Injector.create({ providers: [] })` (core-libs/storefront/layout/launch-dialog/services/inline-root-render.

strategy.ts:L44-L48). La strategia è `providedIn: 'root'` e riceve un `ComponentFactoryResolver` legato all'injector root; con `bootstrapApplication` (projects/storefrontapp/src/main.ts:L17-L17) questo è un `EnvironmentNgModuleRefAdapter`, che crea il proprio `ComponentFactoryResolver` e lo registra come provider (core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L21-L30, angular:packages/core/src/application/create_application.ts:L72-L72, angular:packages/core/src/render3/ng_module_ref.ts:L145-L162); la factory risultante è legata a quel modulo (angular:packages/core/src/render3/component_ref.ts:L97-L101) e `create` concatena l'injector vuoto con l'injector di modulo tramite `ChainedInjector` (angular:packages/core/src/render3/component_ref.ts:L140-L158, angular:packages/core/src/render3/component_ref.ts:L268-L285). Ne segue che il componente **non** vede provider di componenti o di route lazy del trigger: solo l'ambiente root (più l'eventuale standalone injector del componente stesso, angular:packages/core/src/render3/component_ref.ts:L150-L153).

13. **Ordine dei provider decide le config ambigue.** Tutte le strategie sono `Priority.LOW`: una config con sia `outlet` sia `inlineRoot` va a InlineRoot, registrato per ultimo (core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L18-L41, core-libs/core/src/util/applicable.ts:L57-L57).
14. `reverse()` **muta in place** l'array `renderedComponents` dell'`OutletDirective` (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L63-L71).
15. **Focus persistito non condiviso.** Ogni servizio della catena è un'istanza root distinta con la propria `Map`: `LaunchDialogService` (che usa `AutoFocusService`) non vede il focus persistito tramite `FocusDirective` (che usa `KeyboardFocusService`) (core-libs/storefront/layout/a11y/keyboard-focus/persist/persist-focus.service.ts:L23-L25, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L30-L30, core-libs/storefront/layout/a11y/keyboard-focus/focus.directive.ts:L20-L20). In più `AutoFocusService` non è pubblico (core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L17-L17).
16. `TrapFocus.both` **non intrappola**: né start né end, permette solo movimento interno con le frecce (core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.service.ts:L67-L73, core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L111-L116).
17. `lock` **letto solo all'init,** `trap` **a ogni tasto.** Cambiare `lock` dopo `ngOnInit` non ha effetto; `trap` è letto a ogni keydown, ed è per questo che `StorefrontComponent` può attivare il trap riassegnando `focusConfig` (core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L112-L112, core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.directive.ts:L27-L27, core-libs/storefront/layout/main/storefront.component.ts:L168-L168).
18. **Config mutata.** `LockFocusDirective` scrive `autofocus` e `focusOnEscape` nell'oggetto ricevuto e `AutoFocusDirective.ngOnChanges` scrive `autofocus = true`: un literal condiviso fra istanze verrebbe modificato (core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L119-L127, core-libs/storefront/layout/a11y/keyboard-focus/autofocus/auto-focus.directive.ts:L74-L76).
19. `ng-reflect-router-link` **non generato per default.** `requiresExplicitTabIndex` lo controlla per riconoscere i link focusabili (core-libs/storefront/layout/a11y/keyboard-focus/base/base-focus.directive.ts:L105-L110), ma Angular 21 non produce più gli attributi `ng-reflect-*` per default: solo in dev mode e con `provideNgReflectAttributes()`, che Spartacus non usa (angular:packages/core/src/ng_reflect.ts:L11-L12, angular:packages/core/src/ng_reflect.ts:L25-L29). Il ramo è di fatto inattivo.
20. `isHidden` **=** `offsetParent === null`**.** DA VERIFICARE: secondo il comportamento DOM standard anche gli elementi `position: fixed` hanno `offsetParent` null e risulterebbero "nascosti"; non verificato nel repo (core-libs/storefront/layout/a11y/keyboard-focus/services/select-focus.util.ts:L90-L92).
21. **Selettore malformato in** `SkipFocusDirective`: `[tabindex]:not([tabindex="-1"]` manca di una parentesi (core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L53-L54). DA VERIFICARE come i browser lo interpretino (probabile chiusura implicita a fine input).
22. `SkipFocusDirective` **in disattivazione** scrive `tabindex="0"` su tutti gli elementi trovati, anche quelli nativamente focusabili che prima non lo avevano (core-libs/storefront/layout/a11y/keyboard-focus/skip-

focus.directive.ts:L52-L66).

23. `tap` **di focus duplicato.** `isExpanded$` è riassegnato in `ngOnInit` e sottoscritto sia dall'`async` del template sia da `trapFocusOnMenuIfExpanded`: `focusOnFirstNavigationItem` può girare una volta per sottoscrittore (core-libs/storefront/layout/main/storefront.component.ts:L131-L139, core-libs/storefront/layout/main/storefront.component.html:L7-L7).
24. **Primo caricamento senza focus automatico.** `onNavigation` sposta il focus solo se `activeElement !== body` (core-libs/storefront/layout/main/storefront.component.ts:L199-L202).
25. `@ViewChild(SkipLinkComponent) child` non ha letture di `.child` in `core-libs/storefront/layout/main/` e la skip link non è nel template ma nell'outlet (core-libs/storefront/layout/main/storefront.component.ts:L92-L92). `DA VERIFICARE` se sia mai valorizzato: non eseguibile senza runtime.
26. `SkipLink.position` **inutilizzato**: copiato in `add` ma non letto da `scrollToTarget` (core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L37-L37, core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L56-L87).
27. **Array mutato nel** `BehaviorSubject`**.** `SkipLinkService.add` fa `splice` sull'array corrente e ri-emette lo stesso riferimento (core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L33-L40). La lista viene comunque ri-renderizzata: `AsyncPipe` chiama `markForCheck()` a ogni emissione, anche con lo stesso riferimento (angular:packages/common/src/pipes/async_pipe.ts:L147-L147, angular:packages/common/src/pipes/async_pipe.ts:L235-L241), e `*ngFor` confronta il contenuto dell'array con un differ in `ngDoCheck` (angular:packages/common/src/directives/ng_for_of.ts:L253-L281, core-libs/storefront/layout/a11y/skip-link/component/skip-link.component.html:L1-L2).
28. `NativeSelectSpaceDirective`: il JSDoc dice che il keydown Space è soppresso, ma il codice sopprime solo i keydown con `event.repeat` (core-libs/storefront/layout/a11y/native-select-space/native-select-space.directive.ts:L14-L18, core-libs/storefront/layout/a11y/native-select-space/native-select-space.directive.ts:L31-L37).
29. `DomChangeDirective` **in SSR.** Il `MutationObserver` è creato nel costruttore senza guard di piattaforma (core-libs/storefront/layout/a11y/on-dom-change/dom-change.directive.ts:L42-L61). `DA VERIFICARE` se sia definito nel runtime SSR (cap. 39).
30. `IntersectionService` **in SSR non emette mai** `true`; è `DeferLoaderService` a caricare subito sul server (core-libs/storefront/layout/loading/intersection.service.ts:L51-L53, core-libs/storefront/layout/loading/defer-loader.service.ts:L59-L59).
31. **API Angular deprecate**: `ComponentFactoryResolver`, `createComponent(factory)` e `APP_INITIALIZER` (sez. 7). Un aggiornamento futuro di Angular che le rimuova toccherà Outlet/Inline/InlineRoot strategy e `skipLinkFactory`.

---

10. Come fare debug

- **Quale breakpoint è attivo?** Iniettare `BreakpointService` e sottoscrivere `breakpoint$`; `breakpoints` restituisce l'ordine effettivo usato da `getBreakpoint` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L60-L65). Ricordare il debounce di 300 ms del resize (core-libs/core/src/window/window-ref.ts:L91-L107).
- **Slot mancanti in una sezione?** Verificare che la recipe sia fornita (trappola 1) e quale config vince per il breakpoint corrente (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243); confrontare HTML SSR (sempre `xs`) e client.
- **Il dialog non si apre.** In dev mode cercare in console `No configuration provided for caller` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L93-L95) o `No view container`

`ref provided for ...` (core-libs/storefront/layout/launch-dialog/services/inline-render.strategy.ts:L42-L71). Se si apre la prima volta ma non la seconda, controllare che l'observable di `openDialog` sia sottoscritto e che `closeDialog` venga chiamato (trappola 8).

- **Il dialog si apre nel posto sbagliato.** Controllare quali discriminanti (`outlet`, `inline`, `inlineRoot`, `cxRoute`) sono presenti nella config finale, ricordando l'ordine dei provider (trappola 13). La config finale è leggibile iniettando `LayoutConfig` e guardando `launch`.
- **Il `body` resta con `modal-open`.** Significa che `remove` non è stato chiamato (manca `clear`) (core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L145-L153).
- **Focus da tastiera.** Ispezionare nel DOM gli attributi `tabindex`, `data-cx-focus`, `data-cx-focus-group` e le classi `mouse-focus`, `focus-lock`, `is-locked` impostati dalla catena (core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L7-L10, core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L42-L56). In DevTools `document.activeElement` mostra dove è finito il focus dopo una navigazione (core-libs/storefront/layout/main/storefront.component.ts:L194-L228).
- **Skip link mancante.** La chiave deve esistere in `skipLinks` e il target avere `[cxSkipLink]` (core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L27-L42).
- **Unit test.** Usare `KeyboardFocusTestingModule` per sostituire `[cxFocus]` con un mock senza logica (core-libs/storefront/layout/a11y/keyboard-focus/focus-testing.module.ts:L10-L19).

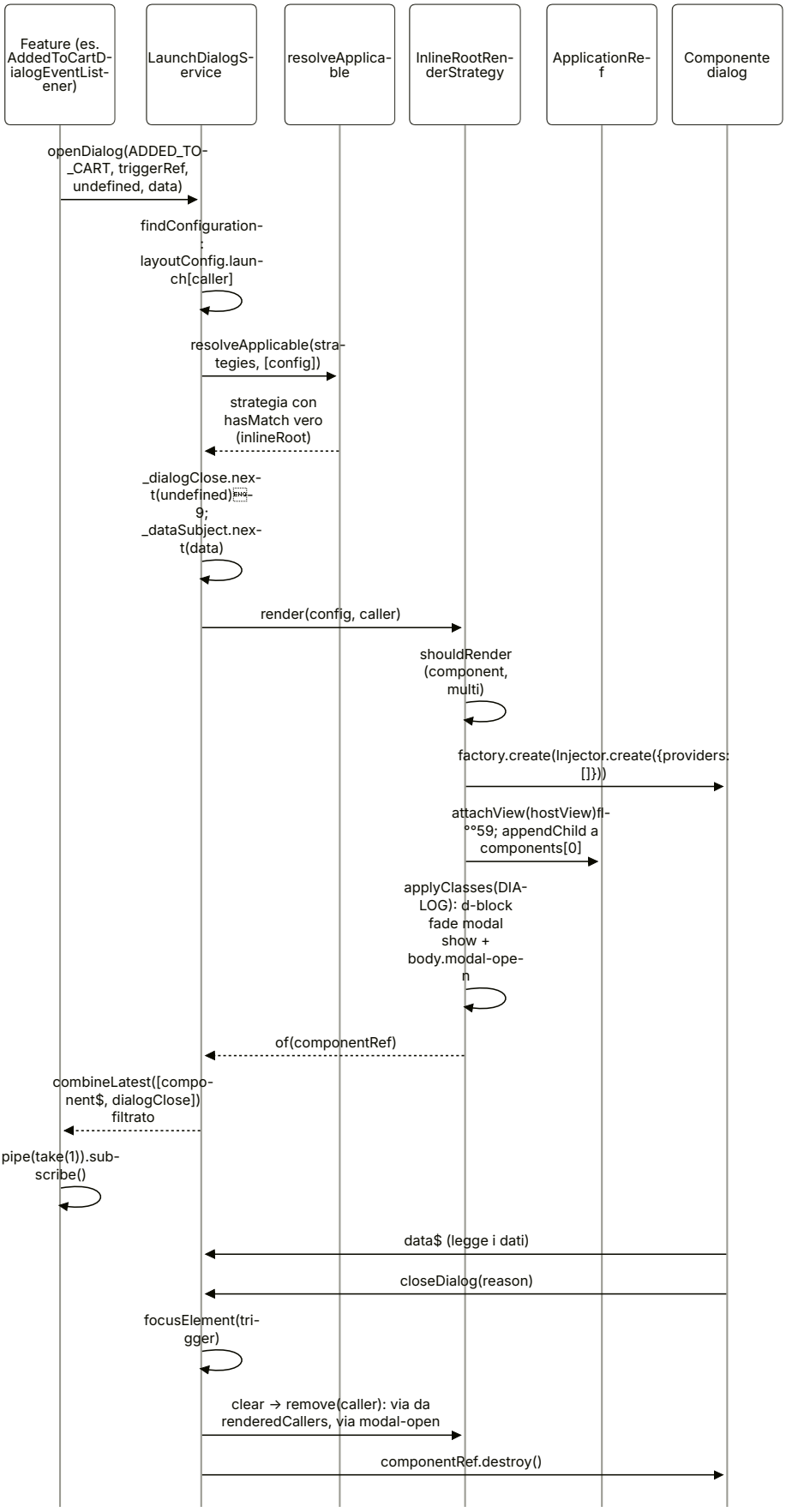
---

11. Domande di autoverifica

1. **Con la config di default, quale breakpoint corrisponde a una finestra larga 1200 px? E in SSR?** `xl`: nessun breakpoint ha `1200 < getSize(br)` e `getBreakpoint` restituisce l'ultimo; in SSR sempre `xs`, il primo della lista (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L192-L199, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L123-L131).
2. **Perché i `layoutSlots` non sono in `defaultLayoutConfig` e che cosa usare?** `defaultLayoutConfig` contiene solo i breakpoint; gli slot stanno nella recipe, oggi tramite `layoutConfigFactory()` con `provideConfigFactory`, che unifica l'header e rimuove `pageFold` per evitare CLS (core-libs/storefront/layout/config/default-layout.config.ts:L9-L19, core-libs/storefront/recipes/config/layout-config.ts:L25-L25, core-libs/storefront/recipes/config/layout-config.ts:L158-L198).
3. **Hai configurato slot solo per `lg` e per il default. Cosa vede un utente su `xl` e su `md`?** Su `xl` la config `lg` (il più vicino minore che la definisce); su `md` la config di default, perché `lg` non è minore di `md` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243).
4. **Come sceglie `LaunchDialogService` la strategia di rendering?** `resolveApplicable` sulle strategie multi-provider, filtrate per `hasMatch` sulla config (`outlet`, `inline`, `cxRoute`, `inlineRoot`); a pari priorità (tutte LOW) vince l'ultimo provider (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L181-L185, core-libs/core/src/util/applicable.ts:L47-L74, core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L18-L41).
5. **Perché un dialog aperto con `openDialog` senza subscribe non si può riaprire?** Il cleanup (`clear` → `remove`) gira solo nel `tap` dell'observable restituito; il caller resta in `renderedCallers` e `shouldRender` lo blocca senza `multi` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L61-L69, core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L92-L102).
6. **Come si aggiunge un nuovo `LAUNCH_CALLER` in una libreria?** Module augmentation dell'enum in `@spartacus/storefront` più assegnazione del valore a runtime, poi una voce in `LayoutConfig.launch` (feature-libs/cart/base/root/models/augmented.model.ts:L7-L16, feature-libs/cart/base/components/added-to-cart-dialog/default-added-to-cart-layout.config.ts:L10-L18).

7. **Cosa succede alla pressione di Escape due volte di seguito dentro** `cx-storefront`**?** Al primo Escape il focus va all'host (che ha `tabindex="-1"`); al secondo, essendo il target già l'host e `focusOnDoubleEscape` vero, va al primo focusable (`core-libs/storefront/layout/main/storefront.component.ts:L86-L106`, `core-libs/storefront/layout/a11y/keyboard-focus/escape/escape-focus.service.ts:L24-L42`).
8. **Come viene intrappolato il focus quando il menu hamburger è aperto?** `trapFocusOnMenuIfExpanded` imposta `trap: true` sul `[cxFocus]` dell'header e `isEnabled: true` su `[cxSkipFocus]`, che mette `tabindex=-1` su tutti i focusabili della sezione header tranne `button.cx-hamburger`; `TrapFocusService.moveFocus` cicla Tab e frecce (`core-libs/storefront/layout/main/storefront.component.ts:L164-L174`, `core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L38-L67`, `core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.service.ts:L34-L65`).
9. **Come arriva** `SkipLinkComponent` **nel DOM, visto che non è nel template di** `cx-storefront`**?** Un `APP_INITIALIZER` (`skipLinkFactory`) aggiunge la sua factory all'outlet `cx-storefront` in posizione `BEFORE` (`core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L37-L60`, `core-libs/storefront/layout/main/storefront.component.html:L1-L1`).
10. **Perché in SSR un componente CMS con strategia DEFER viene comunque renderizzato?** `shouldLoadInstantly` restituisce `true` su piattaforma server prima di valutare la strategia; `IntersectionService` in SSR restituirebbe solo `false` (`core-libs/storefront/layout/loading/defer-loader.service.ts:L55-L64`, `core-libs/storefront/layout/loading/intersection.service.ts:L45-L58`).

TAVOLA 18.1 **4.2 Ciclo di vita di un launch dialog (inlineRoot)**



19

State management: NgRx, loader/entity/process state

Higher-order reducer, azioni base e selector per dati in caricamento, caricati o falliti, indicizzati per id.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 07 — SSR, hydration e TransferState, 11 — Configurazione, 12 — Feature module e lazy loading. Capitoli collegati: 20 — Facade, command & query, eventi, 25 — Carrello.

1. A cosa serve e perché esiste

Spartacus usa NgRx (21.0.x) come store applicativo, ma **non** si limita a fornire reducer per le singole feature: il sottosistema `core-libs/core/src/state/` è una piccola libreria di *higher-order reducer* (HOR), azioni base e selector puri che tutte le feature riusano per gestire lo stesso problema ricorrente: "ho dei dati che arrivano dal backend OCC, possono essere in caricamento, caricati, falliti o da resettare; e spesso sono molti, indicizzati per id".

Il capitolo copre cinque pezzi distinti, tutti esportati da `@spartacus/core` (core-libs/core/public_api.ts:L33-L33, core-libs/core/src/state/index.ts:L7-L11; core-libs/core/public_api.ts:L28-L28, core-libs/core/src/process/index.ts:L7-L9):

1. `StateModule.forRoot()` e il **meta-reducer di transfer state**: copia slice selezionate dello store NgRx dal render SSR al browser tramite l'`TransferState` di Angular (core-libs/core/src/state/state.module.ts:L10-L18, core-libs/core/src/state/reducers/index.ts:L23-L43).
2. `StateConfig`: la sola chiave di configurazione del sottosistema, `state.ssrTransfer.keys` (core-libs/core/src/state/config/state-config.ts:L20-L35).
3. `StatePersistenceService`: sincronizzazione di uno stream di stato con `localStorage`/`sessionStorage`, con chiave contestuale (es. per base site) (core-libs/core/src/state/services/state-persistence.service.ts:L18-L132).
4. `StateUtils`: il namespace con loader, entity, entity-loader, processes-loader, entity-processes-loader, serializer B2B e `getStateSlice` (core-libs/core/src/state/utils/index.ts:L7-L8, core-libs/core/src/state/utils/utils-group.ts:L7-L14).
5. `ProcessModule`: una feature NgRx generica, `process`, che è un `EntityLoaderState` indicizzato per *process id* e serve a tracciare operazioni "fire and forget" (dare un consenso, cancellare un ordine) senza creare una slice dedicata per ognuna (core-libs/core/src/process/store/process-store.module.ts:L13-L17, core-libs/core/src/process/store/reducers/index.ts:L13-L26).

Completa il quadro `StateEventService`, il ponte che trasforma azioni NgRx in eventi Spartacus (core-libs/core/src/state/event/state-event.service.ts:L16-L77); il sistema eventi in sé è trattato nel capitolo 20.

Il principio che lega tutto: **i reducer generici non guardano** `action.type`**, guardano** `action.meta`. Una feature crea una classe azione che estende un'azione base (es. `EntityLoadAction`), sovrascrive `type` con il proprio nome leggibile nei DevTools, e il reducer generico la interpreta in base a `meta.entityType`, `meta.entityId`, `meta.loader` e `meta.processesCountDiff` (core-libs/core/src/state/utils/loader/loader.reducer.ts:L32-L36, core-libs/core/src/state/utils/entity/entity.reducer.ts:L28-L32, core-libs/core/src/product/store/actions/product.action.ts:L29-L35).

> Nota importante: Spartacus **non crea lo store radice**. `StoreModule.forRoot({})` ed `EffectsModule.forRoot([])` sono nell'`AppModule` dell'applicazione (projects/storefrontapp/src/app/app.module.ts:L15-L22) e vengono inseriti dallo schematic `add-spartacus` (core-libs/schematics/src/add-spartacus/store.ts:L58-L65). Spartacus contribuisce solo feature (`StoreModule.forFeature`) e meta-reducer (`META_REDUCERS`).

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/state/index.ts	Barrel pubblico: config, eventi, StatePersistenceService, StateModule, StateUtils. NON esporta ./reducers (:L7-L11).
core-libs/core/src/state/state.module.ts	StateModule vuoto; forRoot() aggiunge stateMetaReducers (:L10-L18).
core-libs/core/src/state/config/state-config.ts	StorageSyncType, StateTransferType, StateConfig, augmentation di Config (:L10-L39).
core-libs/core/src/state/reducers/index.ts	TRANSFER_STATE_META_REDUCER e stateMetaReducers (registrazione su META_REDUCERS) (:L23-L43).
core-libs/core/src/state/reducers/transfer-state.reducer.ts	CX_KEY = 'cx-state', getTransferStateReducer, varianti server e browser (:L16-L96).
core-libs/core/src/state/services/state-persistence.service.ts	StatePersistenceService: syncWithStorage, readStateFromStorage, generateKeyWithContext (:L18-L132).
core-libs/core/src/state/utils/browser-storage.ts	getStorage, persistToStorage, readFromStorage, isSsr (:L11-L75).
core-libs/core/src/state/utils/get-state-slice.ts	Estrazione di sotto-alberi per path con .: getStateSlice, filterKeysByType (:L10-L112).
core-libs/core/src/state/utils/index.ts, utils-group.ts	Namespace StateUtils (index.ts:L7-L8, utils-group.ts:L7-L14).
core-libs/core/src/state/utils/loader/*	LoaderState, azioni [LOADER] *, loaderReducer, selector puri.
core-libs/core/src/state/utils/processes-loader/*	ProcessesLoaderState (processesCount), processesLoaderReducer, isStableSelector.
core-libs/core/src/state/utils/entity/*	EntityState (mappa entities), azioni remove, entityReducer.
core-libs/core/src/state/utils/entity-loader/*	EntityLoaderState = entity + loader; azioni [ENTITY] LOAD*.
core-libs/core/src/state/utils/entity-processes-loader/*	entity + processes-loader (usato dal multi-cart).
core-libs/core/src/state/utils/scoped-loader/*	loader per entità *e* scope (usato solo dal prodotto); NON in StateUtils (utils-group.ts:L7-L14).
core-libs/core/src/state/utils/entity-list-state.ts, serializer.ts	Pattern lista+entità B2B e serializzazione delle chiavi di ricerca.
core-libs/core/src/state/event/*	ActionToEventMapping, StateEventService, helper di test.
core-libs/core/src/process/process.module.ts	ProcessModule (importa ProcessStoreModule; forRoot senza provider) (:L10-L20).
core-libs/core/src/process/store/process-store.module.ts	Registra la feature process (:L13-L17).
core-libs/core/src/process/store/process-state.ts	PROCESS_FEATURE = 'process', StateWithProcess<T> (:L9-L13).
core-libs/core/src/process/store/reducers/index.ts	getReducers = entityLoaderReducer(PROCESS_FEATURE), reducerToken (:L13-L26).

PATH	RUOLO
`core-libs/core/src/process/store/selectors/*`	`getProcessState`, `getProcess{State,Loading,Success,Error}Factory`, namespace `ProcessSelectors`.
`core-libs/core/src/process/store/selectors/process-group.selectors.ts`	Barrel non importato da nessun file (dossier T24).
`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs`	(contorno, NgRx) `INIT`, `META_REDUCERS`, composizione dei meta-reducer, `State`, `ReducerManager.addFeatures/updateReducers`.

3. Come funziona passo per passo

3.1 Come NgRx compone reducer e meta-reducer (il contesto necessario)

Per capire quando e su cosa agiscono i meta-reducer di Spartacus serve sapere come NgRx 21.0.1 costruisce il reducer radice. I sorgenti NgRx sono letti dal pacchetto pubblicato (`deps:`).

1. **INIT non viene dispatchata da nessuno.** `INIT = '@ngrx/store/init'` è il valore iniziale del `BehaviorSubject ActionsSubject` (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L196-L200`).
2. **Due sorgenti di meta-reducer radice.** Il token multi `META_REDUCERS` raccoglie i meta-reducer "definiti internamente da @ngrx/store o da library author"; quelli passati a `StoreModule.forRoot(…, { metaReducers })` finiscono in un token separato (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L244-L256`). La concatenazione è `metaReducers.concat(userProvidedMetaReducers)`: prima i `META_REDUCERS`, poi quelli dell'utente (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1069-L1071`).
3. **Il reducer factory radice.** `StoreModule.forRoot` fornisce `REDUCER_FACTORY = createReducerFactory(combineReducers, metaReducerRisolti)` (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1337-L1353`). `createReducerFactory` compone `[...metaReducers, reducerFactory]` con `compose`, che applica da destra: il **primo** meta-reducer dell'array è il **più esterno**; il reducer ottenuto sostituisce `state === undefined` con `initialState` prima di chiamare la catena (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L349-L371`). Quindi i meta-reducer radice vedono lo **stato globale intero**, prima e dopo `combineReducers`.
4. **Il runtime check di NgRx usa lo stesso canale.** `StoreModule.forRoot` registra tre provider `META_REDUCERS` (immutability, serialization, in-zone) (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1241-L1258`).
5. **INIT è ridotta alla costruzione di `State`.** Il costruttore di `State` si sottoscrive alle azioni con `observeOn(queueScheduler)`, `withLatestFrom(reducer$)` e `scan` (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L485-L498`); con delay 0 il `queueScheduler` esegue subito (`deps:rxjs-7.8.2/package/dist/esm/internal/scheduler/QueueAction.js:L8-L16`). `Store` dipende dal token NgRx StateObservable (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L627-L627`), fornito da `State` con `useExisting` (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L516-L519`), il modulo NgModule restituito da `StoreModule.forRoot` inietta `Store` e ogni modulo feature inietta il modulo root prima di registrare la propria feature (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1482-L1484`, `:L1502-L1518`, `:L1523-L1523`). Conseguenza: **INIT viene ridotta quando il reducer radice contiene solo i reducer di `forRoot`** (nell'app demo: nessuno, `StoreModule.forRoot({})`).
6. **Le feature arrivano dopo, con UPDATE.** `addFeatures` costruisce il reducer della feature (applicando i `metaReducers` *della feature*), poi `updateReducers` ricostruisce il reducer radice con lo stesso `reducerFactory` (quindi con gli stessi meta-reducer radice) e dispatcha `'@ngrx/store/update-reducers'` (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L391-L391`, `:L406-L416`, `:L438-L444`).

7. `combineReducers` **e le chiavi orfane.** La funzione `combination` passa a ogni reducer `state[key]`; se almeno una chiave cambia, restituisce un **nuovo** oggetto che contiene solo le chiavi con un reducer registrato (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L319-L343`).

I punti 5-7 spiegano il funzionamento del transfer state lato browser (3.3) e una trappola (§9).

3.2 Bootstrap: `StateModule`, `ProcessModule` e i meta-reducer

1. `BaseCoreModule` importa `StateModule.forRoot()` subito dopo `ErrorHandlingModule.forRoot()` e prima di `ConfigModule.forRoot()`; più sotto importa `ProcessModule.forRoot()` (`core-libs/core/src/base-core.module.ts:L26-L36`).
2. `StateModule.forRoot()` aggiunge `stateMetaReducers`; `StateModule` importato senza `forRoot` (come fanno `ProcessStoreModule` e gli store module delle feature) non aggiunge alcun provider (`core-libs/core/src/state/state.module.ts:L10-L18`, `core-libs/core/src/process/store/process-store.module.ts:L14-L14`).
3. `stateMetaReducers` definisce il token `TRANSFER_STATE_META_REDUCER` con factory `getTransferStateReducer` e dipendenze `PLATFORM_ID`, `TransferState`, `Config`, `AuthStatePersistenceService` (le ultime tre `Optional`), e lo inserisce in `META_REDUCERS` con `useExisting` (`core-libs/core/src/state/reducers/index.ts:L23-L43`).
4. `getTransferStateReducer` decide **una volta sola, alla creazione**:
   - se c'è un `TransferState` *e* `config.state.ssrTransfer.keys`, sul browser restituisce `getBrowserTransferStateReducer(…, Boolean(authStatePersistenceService?.isUserLoggedIn()))`, sul server `getServerTransferStateReducer(…)` (`core-libs/core/src/state/reducers/transfer-state.reducer.ts:L18-L37`);
   - altrimenti restituisce il meta-reducer identità `(reducer) => reducer` (`core-libs/core/src/state/reducers/transfer-state.reducer.ts:L39-L39`).
5. `ProcessModule` importa `ProcessStoreModule` (`core-libs/core/src/process/process.module.ts:L10-L20`), che registra `StoreModule.forFeature(PROCESS_FEATURE, reducerToken)` (`core-libs/core/src/process/store/process-store.module.ts:L13-L17`) con reducer `entityLoaderReducer(PROCESS_FEATURE)` fornito da factory (`core-libs/core/src/process/store/reducers/index.ts:L13-L26`).

3.3 Transfer state: dal render SSR al browser

Configurazione. Le chiavi da trasferire sono in `state.ssrTransfer.keys`, mappa `chiave → StateTransferType` (`core-libs/core/src/state/config/state-config.ts:L20-L35`); `StateTransferType` ha un solo valore, `TRANSFER_STATE = 'SSR'` (`core-libs/core/src/state/config/state-config.ts:L16-L18`). Il core dichiara come default le feature `cms`, `product`, `siteContext`, `siteTheme`, ognuna con una `provideDefaultConfigFactory` nel proprio store module (`core-libs/core/src/cms/store/cms-store.module.ts:L21-L31`, `core-libs/core/src/product/store/product-store.module.ts:L20-L30`, `core-libs/core/src/site-context/store/site-context-store.module.ts:L20-L30`, `core-libs/core/src/site-theme/store/site-theme-store.module.ts:L20-L29`).

Lato server, dopo **ogni** azione:

1. il meta-reducer calcola `newState = reducer(state, action)` (`core-libs/core/src/state/reducers/transfer-state.reducer.ts:L51-L53`);
2. estrae le chiavi di tipo `'SSR'` con `filterKeysByType` (`core-libs/core/src/state/reducers/transfer-state.reducer.ts:L46-L49`, `core-libs/core/src/state/utils/get-state-slice.ts:L104-L112`);
3. costruisce la slice con `getStateSlice(transferStateKeys, [], newState)` e la scrive con `transferState.set(CX_KEY, stateSlice)` (`core-libs/core/src/state/reducers/transfer-state.reducer.ts:L54-L57`). `CX_KEY` è `makeStateKey('cx-state')` (`core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16`).

`TransferState.set` assegna semplicemente la chiave nel suo `store` (angular:packages/core/src/transfer_state.ts:L89-L112) e `toJson()` serializza con `JSON.stringify`, facendo l'escape di `<` e `/` (angular:packages/core/src/transfer_state.ts:L131-L146). L'inserimento dello `<script>` nell'HTML è compito di platform-server (vedi capitolo 07; DA VERIFICARE in quel capitolo: fuori dai file di questo).

Lato browser:

1. Angular `TransferState`, se non è `ngServerMode`, popola `store` con `retrieveTransferredState(DOCUMENT, APP_ID)` (angular:packages/core/src/transfer_state.ts:L66-L79), cioè leggendo lo `<script>` con id `appId + '-state'` (angular:packages/core/src/transfer_state.ts:L150-L157).
2. Il meta-reducer browser interviene **solo sull'azione** `INIT` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L76-L76). Se lo stato è vuoto lo crea chiamando il reducer (:L77-L79); poiché INIT è ridotta quando lo store contiene solo i reducer di `forRoot` (3.1, punto 5), questo stato iniziale è essenzialmente vuoto.
3. Se l'utente **non** risulta loggato e la chiave `cx-state` esiste, rilegge il valore, lo **ri-filtra** con le chiavi configurate sul client e lo fonde con `deepMerge({}, state, transferredStateSlice)` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L81-L90). `isUserLoggedIn()` legge in modo sincrono dallo storage la presenza di `token.access_token` e il suo JSDoc dice esplicitamente che esiste per il transfer state reducer (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L115-L121).
4. Ogni altra azione passa invariata (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L93-L93).
5. Quando gli store module (`cms`, `product`, ...) registrano le loro feature, `combineReducers` passa a ciascun reducer la slice già presente in `state[key]` (3.1, punti 6-7): è così che il reducer CMS "trova" lo stato del server come proprio stato precedente.

Perché il controllo sul login? Lo stato renderizzato dal server è anonimo (il server non ha il token del browser); se l'utente è loggato, contenuti come CMS e prezzi potrebbero differire, quindi Spartacus preferisce ricaricarli. Il codice esprime questa scelta con la condizione `!isLoggedIn` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L81-L81); la motivazione non è scritta nel sorgente (deduzione dell'autore, coerente con il JSDoc citato).

3.4 `getStateSlice`: estrarre sotto-alberi per path

- `getStateSliceValue` naviga lo stato con un path separato da `'.'` (core-libs/core/src/state/utils/get-state-slice.ts:L10-L21).
- `createShellObject` ricostruisce l'oggetto annidato attorno al valore con `reduceRight`; se la chiave è vuota, il valore è falsy oppure `Object.keys(value)` è vuoto (oggetti vuoti, ma anche booleani e numeri) restituisce `{}` (core-libs/core/src/state/utils/get-state-slice.ts:L23-L36).
- `getStateSlice` fonde con `deepMerge` le shell di tutte le chiavi; con array chiavi vuoto restituisce `{}` (core-libs/core/src/state/utils/get-state-slice.ts:L38-L55).
- `handleExclusions` clona e cancella i path esclusi; `getExclusionKeys` seleziona le esclusioni con `String.includes` (core-libs/core/src/state/utils/get-state-slice.ts:L57-L102). Il transfer state passa sempre `excludeKeys = []` (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L55-L55).

Le chiavi di `ssrTransfer.keys` possono quindi essere path annidati, es. `'cms.page'` (sintassi supportata da `getStateSliceValue`; nessun uso di questo tipo trovato nei default del core, che usano solo chiavi di primo livello).

3.5 Storage sync: `StatePersistenceService`

`StatePersistenceService` è `providedIn: 'root'` e dipende solo da `WindowRef` (core-libs/core/src/state/services/state-persistence.service.ts:L18-L22).

`syncWithStorage({ key, state$, context$, storageType, onRead })`:

1. default: `context$ = of('')`, `storageType = LOCAL_STORAGE`, `onRead` no-op (core-libs/core/src/state/services/state-persistence.service.ts:L44-L58);
2. risolve lo storage con `getStorage`: `LOCAL_STORAGE` → `winRef.localStorage`, `SESSION_STORAGE` → `winRef.sessionStorage`, `NO_STORAGE` → `undefined`, qualunque altro valore → `sessionStorage` (core-libs/core/src/state/utils/browser-storage.ts:L11-L37); in SSR `WindowRef` restituisce `undefined` per entrambi (core-libs/core/src/window/window-ref.ts:L50-L60);
3. **prima** sottoscrizione (lettura): a ogni emissione di `context$` legge dallo storage (o `undefined` se non c'è storage) e chiama `onRead`; il commento impone che la lettura sia sottoscritta prima della scrittura (core-libs/core/src/state/services/state-persistence.service.ts:L63-L78);
4. **seconda** sottoscrizione (scrittura, solo se lo storage esiste): a ogni emissione di `state$` scrive con l'ultimo contesto via `withLatestFrom(context$)` (core-libs/core/src/state/services/state-persistence.service.ts:L80-L90);
5. restituisce la `Subscription` composta, che il chiamante deve chiudere (core-libs/core/src/state/services/state-persistence.service.ts:L61-L61, :L92-L92).

La chiave nello storage è `spartacus⚿<contesti uniti da ⚿>⚿<key>` (core-libs/core/src/state/services/state-persistence.service.ts:L124-L131); il JSDoc porta l'esempio `spartacus⚿electronics⚿cart` (core-libs/core/src/state/services/state-persistence.service.ts:L28-L34). `persistToStorage` scrive `JSON.stringify(value)` e scarta in silenzio `undefined` (core-libs/core/src/state/utils/browser-storage.ts:L39-L55); `readFromStorage` fa `JSON.parse` senza `try/catch` (core-libs/core/src/state/utils/browser-storage.ts:L57-L71). La variante sincrona è `readStateFromStorage` (core-libs/core/src/state/services/state-persistence.service.ts:L105-L122).

Uso reale: `MultiCartStatePersistenceService` sincronizza la chiave `'cart'` con contesto `BASE_SITE_CONTEXT_ID` su `LOCAL_STORAGE` (feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L35-L43).

3.6 Loader state: la cella di base

`LoaderState<T> = { loading?, error?, success?, value? }` (core-libs/core/src/state/utils/loader/loader-state.ts:L7-L12); lo stato iniziale ha i tre flag a `false` e `value: undefined` (core-libs/core/src/state/utils/loader/loader.reducer.ts:L11-L16).

Le azioni portano un `meta: LoaderMeta = { entityType, loader }` (core-libs/core/src/state/utils/loader/loader.action.ts:L15-L29) costruito dalle meta factory: `loadMeta` → `{load: true}`, `failMeta` → `{error: error ? error : true}`, `successMeta` → `{success: true}`, `resetMeta` → `{}` (core-libs/core/src/state/utils/loader/loader.action.ts:L31-L63). Le classi `Loader{Load,Fail,Success,Reset}Action` impostano `type` e `meta`; `LoaderFailAction` implementa anche `ErrorAction` (core-libs/core/src/state/utils/loader/loader.action.ts:L65-L101), cioè ha un campo `error`, che è ciò che `ErrorActionService.isErrorAction` controlla (`'error' in action`) (core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L41-L43).

`loaderReducer(entityType, reducer?)`:

1. reagisce solo se `action.meta.loader` esiste e `meta.entityType === entityType` (core-libs/core/src/state/utils/loader/loader.reducer.ts:L32-L37);
2. **LOAD**: `loading: true`, gli altri flag invariati (:L39-L44);

3. **FAIL**: `loading: false, error: true, success: false`, `value` azzerato se non c'è reducer interno (`:L45-L52`);
4. **SUCCESS**: `value = action.payload` (o reducer interno), `loading: false, error: false, success: true` (`:L53-L60`);
5. qualunque altro `meta.loader` (es. `{}`) è un **RESET** a `initialLoaderState` (`:L61-L69`);
6. per ogni azione che non ha un `meta.loader` truthy con lo stesso `entityType` (comprese le azioni loader di altri tipi e gli increment/decrement dei processi, che hanno `loader: undefined`: core-libs/core/src/state/utils/processes-loader/processes-loader.action.ts:L30-L34), se c'è un reducer interno lo applica a `value`, cambiando riferimento solo se il valore cambia (`:L72-L78`);
7. `getReducerIfExists` usa **sempre** il reducer interno quando presente, ignorando il fallback (`:L80-L82`).

I selector di loader sono funzioni pure, non `createSelector`, con default `false` via `??` (core-libs/core/src/state/utils/loader/loader.selectors.ts:L9-L23).

3.7 Entity state e entity-loader: molte celle per id

`EntityState<T>` è `{ entities: { [id]: T } }` (core-libs/core/src/state/utils/entity/entity-state.ts:L7-L11); `EntityMeta` aggiunge `entityId?: string | string[] | null` ed `entityRemove?` (core-libs/core/src/state/utils/entity/entity.action.ts:L12-L18).

`entityReducer(entityType, reducer)`:

1. se `entityType` coincide ed `entityId !== undefined`, gli id sono `[].concat(entityId ?? '')`; se `entityRemove` è vero rimuove le entità (core-libs/core/src/state/utils/entity/entity.reducer.ts:L28-L38);
2. altrimenti (azione di altro tipo, o senza id) gli id sono **tutte** le chiavi esistenti (`:L42-L44`);
3. se `entityId` e `payload` sono entrambi array, l'entità *i* riceve `payload[i]` (`:L40-L52`);
4. applica il reducer interno per id; ogni risultato truthy, anche se identico allo stato precedente, entra negli update e lo stato cambia riferimento se c'è almeno un update (`:L46-L66`). Con un reducer interno di tipo loader, che per le azioni non pertinenti restituisce `state` (core-libs/core/src/state/utils/loader/loader.reducer.ts:L72-L78), se esiste almeno un'entità lo stato cambia riferimento a **ogni** azione dispatchata (core-libs/core/src/state/utils/entity/entity.reducer.ts:L42-L44, `:L53-L63`);
5. `removeSelectedEntities`: con `entityId === null` torna a `{entities: {}}`, altrimenti filtra (`:L69-L92`).

`entityLoaderReducer` è la composizione `entityReducer(entityType, loaderReducer(entityType, reducer))` (core-libs/core/src/state/utils/entity-loader/entity-loader.reducer.ts:L13-L25); `EntityLoaderState<T> = EntityState<LoaderState<T>>` (core-libs/core/src/state/utils/entity-loader/entity-loader-state.ts:L10-L10). Le meta entity-loader sono lo spread di loader meta + `entityMeta` (core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L30-L69); `entityMeta(type, id?)` assegna `entityId: id` anche quando `id` è `undefined` (core-libs/core/src/state/utils/entity/entity.action.ts:L20-L25). `entityLoaderStateSelector` restituisce `initialLoaderState` per id inesistenti (core-libs/core/src/state/utils/entity-loader/entity-loader.selectors.ts:L17-L54).

3.8 Processes loader ed entity-processes-loader: contare le operazioni in volo

`ProcessesLoaderState<T>` aggiunge `processesCount?` (core-libs/core/src/state/utils/processes-loader/processes-loader-state.ts:L9-L11). Le azioni increment/decrement hanno `loader: undefined` e `processesCountDiff` +1/-1 (core-libs/core/src/state/utils/processes-loader/processes-loader.action.ts:L27-L45); il reset dei processi è anche un reset del loader, con `processesCountDiff: null` (core-libs/core/src/state/utils/processes-loader/processes-loader.action.ts:L47-L55).

`processesLoaderReducer`:

1. delega prima a `loaderReducer(entityType, reducer)` (core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L34-L34);
2. se l'`entityType` coincide: in dev mode logga un `console.error` se il contatore scenderebbe sotto zero (`:L35-L52`);
3. con diff numerico somma (se il contatore è falsy diventa direttamente il diff) (`:L53-L59`); con `null` torna a `processesCount: 0` (`:L60-L66`).

`isStableSelector = !processesCount && !loading`; `hasPendingProcessesSelector = processesCount > 0` (core-libs/core/src/state/utils/processes-loader/processes-loader.selectors.ts:L9-L17).

`entityProcessesLoaderReducer = entityReducer(entityType, processesLoaderReducer(entityType, reducer))` (core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader.reducer.ts:L13-L25). Il consumatore principale è il multi-cart: la slice `carts` è `entityProcessesLoaderReducer(MULTI_CART_DATA, createCartEntitiesReducer(enabled))` (feature-libs/cart/base/core/store/reducers/index.ts:L34-L47); le azioni che modificano il carrello estendono `EntityProcessesIncrementAction` (feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L22-L22) e la stabilità si legge con `entityIsStableSelector` (feature-libs/cart/base/core/store/selectors/multi-cart.selector.ts:L53-L61).

3.9 Scoped loader: entità × scope (solo prodotto)

`ScopedLoaderState<T>` è una mappa `scope → LoaderState<T>` (core-libs/core/src/state/utils/scoped-loader/scoped-loader.state.ts:L10-L14). `scopedLoaderReducer` applica `loaderReducer` alla sotto-chiave `meta.scope ?? ''` (core-libs/core/src/state/utils/scoped-loader/scoped-loader.reducer.ts:L27-L43); `entityScopedLoaderReducer` lo avvolge in `entityReducer` (core-libs/core/src/state/utils/scoped-loader/entity-scoped-loader.reducer.ts:L20-L31). Lo usa solo lo store prodotto, per `searchByCode`, `searchByCategory` e `details` (core-libs/core/src/product/store/reducers/index.ts:L22-L35), con azioni come `LoadProduct` che estendono `EntityScopedLoaderActions.EntityScopedLoadAction` (core-libs/core/src/product/store/actions/product.action.ts:L29-L35). Dettagli sugli scope di prodotto nel capitolo 24.

3.10 La feature `process`

1. Le azioni di processo riusano le azioni entity-loader con `entityType = PROCESS_FEATURE` ed `entityId = <process id>` (core-libs/core/src/user/store/actions/customer-coupon.action.ts:L115-L139); i process id sono costanti stringa delle feature, es. `'subscribeCustomerCoupon'` (core-libs/core/src/user/store/user-state.ts:L40-L40).
2. Il reducer `entityLoaderReducer(PROCESS_FEATURE)` non ha reducer interno: per un processo interessano solo i flag (core-libs/core/src/process/store/reducers/index.ts:L13-L15).
3. Lettura: `getProcessStateFactory(id) = createSelector(getProcessState(), s => entityLoaderStateSelector(s, id))` e le derivate loading/success/error (core-libs/core/src/process/store/selectors/process.selectors.ts:L18-L53). Nel core si importano direttamente (core-libs/core/src/user/facade/user-consent.service.ts:L201-L202), nelle feature lib tramite il namespace pubblico `ProcessSelectors` (feature-libs/order/core/facade/order-history.service.ts:L193-L202).

3.11 Azioni → eventi: `StateEventService`

`register({ action, event, factory? })` registra in `EventService` uno stream ottenuto filtrando `ActionsSubject` con `ofType` sui tipi indicati e mappando ogni azione in evento (core-libs/core/src/state/event/state-event.service.ts:L36-L55). Senza factory l'evento è `createFrom(eventType, action.payload ??`

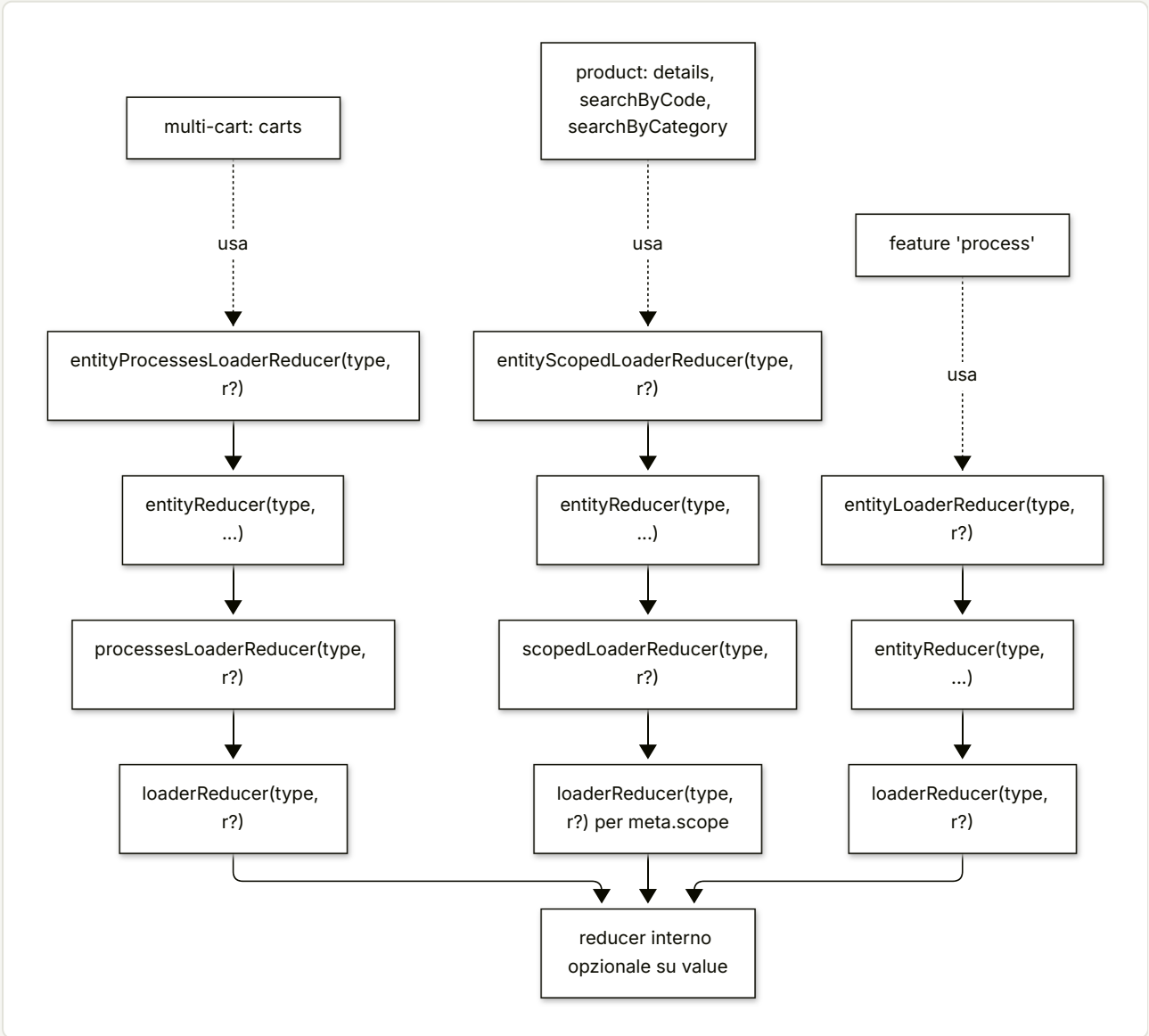
`{})` (core-libs/core/src/state/event/state-event.service.ts:L68-L76), cioè `Object.assign(new type(), data)` (core-libs/core/src/util/create-from.ts:L15-L17). Esempi: senza factory `UPDATE_USER_ADDRESS` → `UpdateUserAddressEvent` (core-libs/core/src/user/events/user-event.builder.ts:L33-L38); con factory `DeleteCartEvent` arricchito con `cartCode` (feature-libs/cart/base/core/event/cart-event.builder.ts:L137-L144).

---

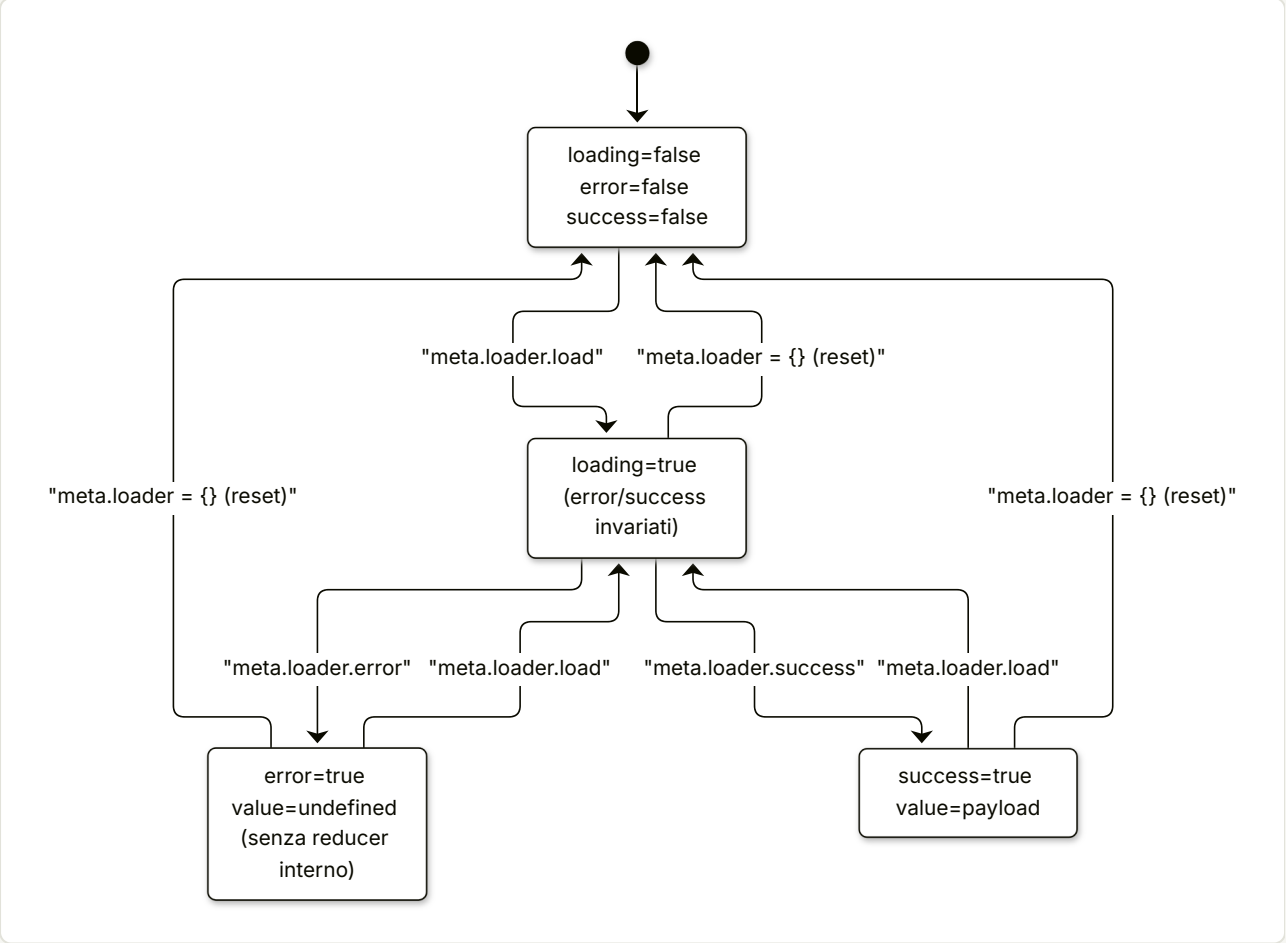
4. Diagrammi

4.1 Composizione degli higher-order reducer

Ogni freccia è "avvolge"; i riferimenti sono in 3.6-3.10.

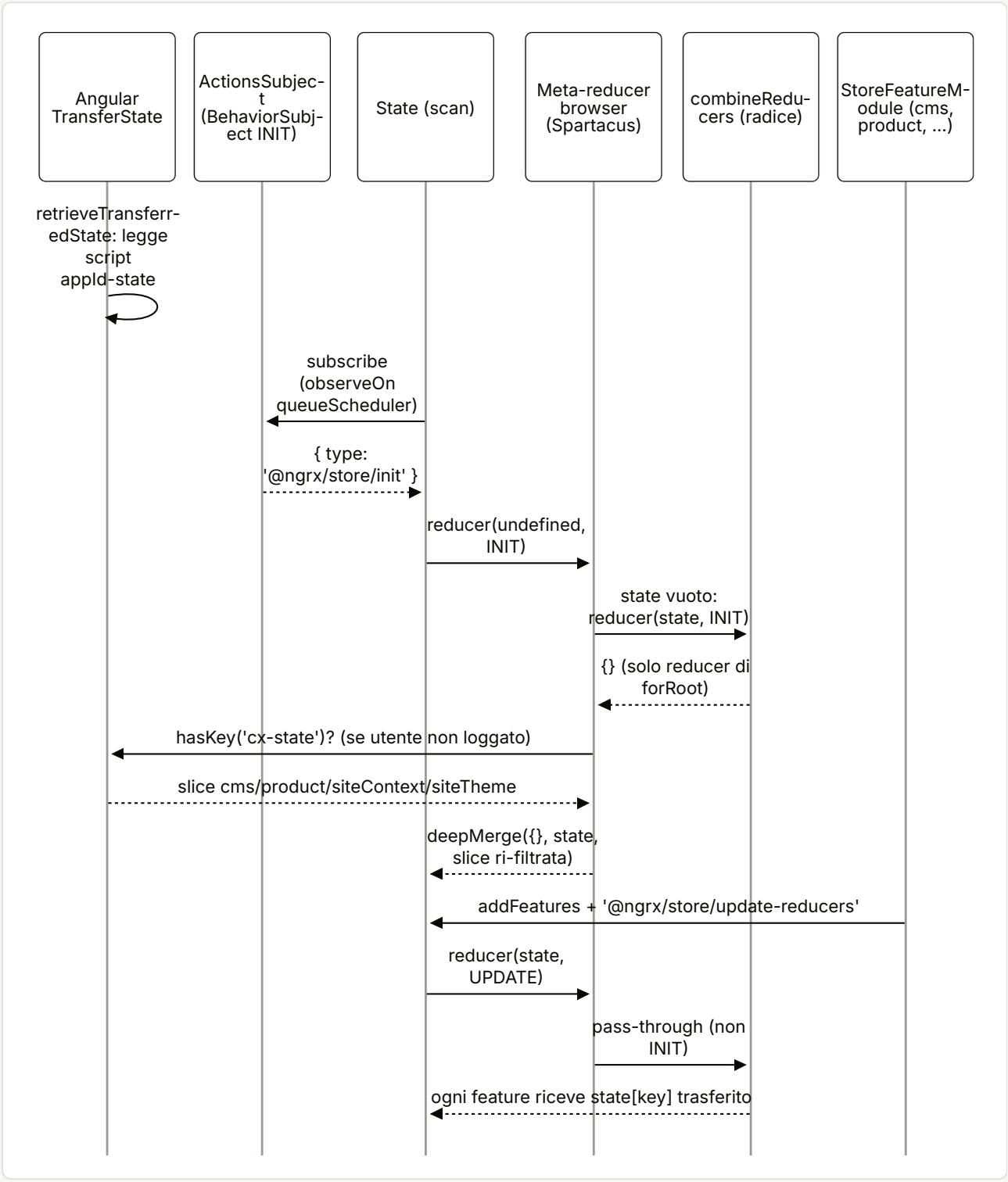


4.2 Transizioni di una cella `LoaderState`



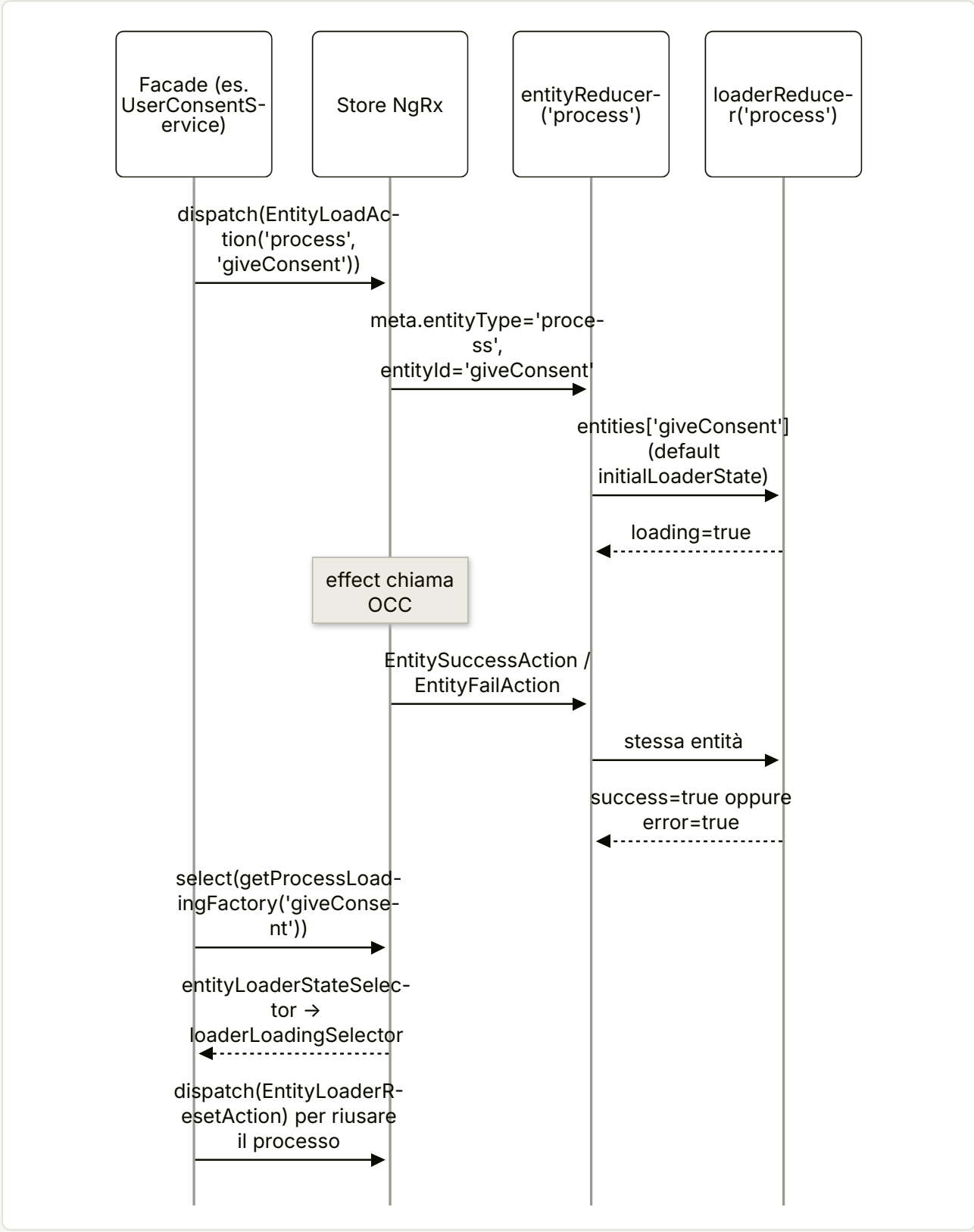
Riferimenti: `core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L69`.

4.3 Bootstrap NgRx e transfer state nel browser



Riferimenti: angular:packages/core/src/transfer_state.ts:L66-L79; deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L196-L200, :L485-L498, :L438-L444; core-libs/core/src/state/reducers/transfer-state.reducer.ts:L74-L94.

4.4 Ciclo di vita di un processo (entity-loader `process`)



Riferimenti: `core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L71-L108`, `core-libs/core/src/state/utils/entity/entity.reducer.ts:L28-L57`, `core-libs/core/src/process/store/selectors/process.selectors.ts:L18-L35`, `core-libs/core/src/user/facade/user-consent.service.ts:L201-L202`. Il process id è `GIVE_CONSENT_PROCESS_ID = 'giveConsent'` (`core-libs/core/src/user/store/user-state.ts:L24-`

L24); le azioni sono GiveUserConsent/GiveUserConsentFail/GiveUserConsentSuccess/ResetGiveUserConsentProcess (core-libs/core/src/user/store/actions/user-consents.action.ts:L71-L81, :L85-L85, :L96-L96, :L104-L104).

---

5. Estratti di codice reale

5.1 Registrazione del meta-reducer

core-libs/core/src/state/reducers/index.ts:L23-L43

```ts
export const TRANSFER_STATE_META_REDUCER = new InjectionToken(
  'TransferStateMetaReducer'
);

export const stateMetaReducers: Provider[] = [
  {
    provide: TRANSFER_STATE_META_REDUCER,
    useFactory: getTransferStateReducer,
    deps: [
      PLATFORM_ID,
      [new Optional(), TransferState],
      [new Optional(), Config],
      [new Optional(), AuthStatePersistenceService],
    ],
  },
  {
    provide: META_REDUCERS,
    useExisting: TRANSFER_STATE_META_REDUCER,
    multi: true,
  },
];
```

5.2 Meta-reducer server e browser

core-libs/core/src/state/reducers/transfer-state.reducer.ts:L51-L61

```ts
  return function (reducer: any) {
    return function (state: any, action: any) {
      const newState = reducer(state, action);
      if (newState) {
        const stateSlice = getStateSlice(transferStateKeys, [], newState);
        transferState.set(CX_KEY, stateSlice);
      }
```

```
      return newState;
    };
  };
```

core-libs/core/src/state/reducers/transfer-state.reducer.ts:L74-L95

```ts
  return function (reducer: any) {
    return function (state: any, action: any) {
      if (action.type === INIT) {
        if (!state) {
          state = reducer(state, action);
        }

        if (!isLoggedIn && transferState.hasKey(CX_KEY)) {
          const cxKey = transferState.get<Object>(CX_KEY, {});
          const transferredStateSlice = getStateSlice(
            transferStateKeys,
            [],
            cxKey
          );

          state = deepMerge({}, state, transferredStateSlice);
        }
        return state;
      }
      return reducer(state, action);
    };
  };
```

5.3 Come NgRx compone i meta-reducer

deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L359-L372

```js
function createReducerFactory(reducerFactory, metaReducers) {
    if (Array.isArray(metaReducers) && metaReducers.length > 0) {
        reducerFactory = compose.apply(null, [
            ...metaReducers,
            reducerFactory,
        ]);
    }
    return (reducers, initialState) => {
        const reducer = reducerFactory(reducers);
        return (state, action) => {
            state = state === undefined ? initialState : state;
            return reducer(state, action);
        };
    };
```

5.4 Il cuore di `loaderReducer`

core-libs/core/src/state/utils/loader/loader.reducer.ts:L32-L70

```ts
  if (
    action.meta &&
    action.meta.loader &&
    action.meta.entityType === entityType
  ) {
    const entity = action.meta.loader;

    if (entity.load) {
      return {
        ...state,
        loading: true,
        value: getReducerIfExists(state.value),
      };
    } else if (entity.error) {
      return {
        ...state,
        loading: false,
        error: true,
        success: false,
        value: getReducerIfExists(undefined),
      };
    } else if (entity.success) {
      return {
        ...state,
        value: getReducerIfExists(action.payload),
        loading: false,
        error: false,
        success: true,
      };
    } else {
      // reset state action
      return {
        ...initialLoaderState,
        value: reducer
          ? reducer(initialLoaderState.value, action)
          : initialLoaderState.value,
      };
    }
  }
```

5.5 Instradamento per id in `entityReducer`

core-libs/core/src/state/utils/entity/entity.reducer.ts:L26-L57

```ts
    let ids: string[] = [];
    let partitionPayload = false;
    if (
      action.meta &&
      action.meta.entityType === entityType &&
      action.meta.entityId !== undefined
    ) {
      ids = ([] as string[]).concat(action.meta.entityId ?? '');

      // remove selected entities
      if (action.meta.entityRemove) {
        return removeSelectedEntities(action, state, ids);
      }

      partitionPayload =
        Array.isArray(action.meta.entityId) && Array.isArray(action.payload);
    } else {
      ids = Object.keys(state.entities);
    }

    const entityUpdates: { [id: string]: T } = {};

    for (let i = 0; i < ids.length; i++) {
      const id = ids[i];
      const subAction = partitionPayload
        ? { ...action, payload: action.payload[i] }
        : action;
      const newState = reducer(state.entities[id], subAction);
      if (newState) {
        entityUpdates[id] = newState;
      }
    }
```

5.6 Azioni base entity-loader

core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L71-L100

```ts
export class EntityLoadAction implements EntityLoaderAction {
  type = ENTITY_LOAD_ACTION;
  readonly meta: EntityLoaderMeta;
  constructor(entityType: string, id: EntityId) {
    this.meta = entityLoadMeta(entityType, id);
  }
}

export class EntityFailAction implements EntityLoaderAction, ErrorAction {
  type = ENTITY_FAIL_ACTION;
  readonly meta: EntityLoaderMeta;
```

```
  public error: any;

  constructor(entityType: string, id: EntityId, error: any) {
    this.meta = entityFailMeta(entityType, id, error);
    this.error = error;
  }
}

export class EntitySuccessAction implements EntityLoaderAction {
  type = ENTITY_SUCCESS_ACTION;
  readonly meta: EntityLoaderMeta;
  constructor(
    entityType: string,
    id: EntityId,
    public payload?: any
  ) {
    this.meta = entitySuccessMeta(entityType, id);
  }
}
```

5.7 Contatore dei processi

core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L34-L69

```ts
    const loaderState = loaderReducer(entityType, reducer)(state, action);
    if (action.meta && action.meta.entityType === entityType) {
      const processesCountDiff = action.meta.processesCountDiff;
      if (
        isDevMode() &&
        state.processesCount &&
        processesCountDiff &&
        state.processesCount + processesCountDiff < 0
      ) {
        //CXSPA-3670 - extract logging to an effect
        /* eslint-disable-next-line no-console */
        console.error(
          // ...
        );
      }
      if (processesCountDiff) {
        return {
          ...loaderState,
          processesCount: state.processesCount
            ? state.processesCount + processesCountDiff
            : processesCountDiff,
        };
      } else if (processesCountDiff === null) {
        // reset action
```

```
      return {
        ...loaderState,
        ...initialProcessesState,
      };
    }
  }

  return loaderState;
```

5.8 Storage sync

core-libs/core/src/state/services/state-persistence.service.ts:L59-L92

```ts
  const storage = getStorage(storageType, this.winRef);

  const subscriptions = new Subscription();

  // Do not change order of subscription! Read should happen before write on context change.
  subscriptions.add(
    context$
      .pipe(
        map((context) => {
          return storage
            ? readFromStorage<T>(
                storage,
                this.generateKeyWithContext(context, key)
              )
            : undefined;
        }),
        tap((state) => onRead(state))
      )
      .subscribe()
  );

  if (storage) {
    subscriptions.add(
      state$.pipe(withLatestFrom(context$)).subscribe(([state, context]) => {
        persistToStorage(
          this.generateKeyWithContext(context, key),
          state,
          storage
        );
      })
    );
  }

  return subscriptions;
```

5.9 La feature `process`

core-libs/core/src/process/store/process-store.module.ts:L13-L17

```ts
@NgModule({
  imports: [StateModule, StoreModule.forFeature(PROCESS_FEATURE, reducerToken)],
  providers: [reducerProvider],
})
export class ProcessStoreModule {}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`state.ssrTransfer.keys`	`{ [key: string]: StateTransferType | undefined }`	nessun default in `state/`; il core aggiunge `cms`, `product`, `siteContext`, `siteTheme` con `provideDefaultConfigFactory`	core-libs/core/src/state/config/state-config.ts:L25-L34	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L24-L35; default in core-libs/core/src/cms/store/cms-store.module.ts:L21-L31, core-libs/core/src/product/store/product-store.module.ts:L20-L30, core-libs/core/src/site-context/store/site-context-store.module.ts:L20-L30, core-libs/core/src/site-theme/store/site-theme-store.module.ts:L20-L29
`StateTransferType.TRANSFER_STATE`	enum `'SSR'`	—	core-libs/core/src/state/config/state-config.ts:L16-L18	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L46-L49, :L69-L72
`syncWithStorage({ storageType })` (parametro, non config globale)	`StorageSyncType`	`LOCAL_STORAGE`	core-libs/core/src/state/services/state-persistence.service.ts:L48-L48	core-libs/core/src/state/utils/browser-storage.ts:L11-L37
`syncWithStorage({ context$ })`	`Observable<string | string[]>`	`of('')`	core-libs/core/src/state/services/state-persistence.service.ts:L47-L47	core-libs/core/src/state/services/state-persistence.service.ts:L65-L77, :L82-L82

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`readStateFromStorage({ context, storageType })`	`string | string[], StorageSyncType`	`'', LOCAL_STORAGE`	core-libs/core/src/state/services/state-persistence.service.ts:L107-L108	core-libs/core/src/state/services/state-persistence.service.ts:L114-L121
`CX_KEY` (chiave `TransferState`, costante non configurabile)	`StateKey<string>`	`'cx-state'`	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L56-L56, :L81-L82
Feature toggle `enableCartSlowNetworkResilience` (contesto: cambia il reducer interno di `carts`)	`boolean`	`false` (l'app demo lo imposta a `true`)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1084-L1084	feature-libs/cart/base/core/store/reducers/index.ts:L37-L37; demo projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L372-L372

Esempio di configurazione (usa solo `provideConfig`, `StateConfig`, `StateTransferType`, tutti pubblici): aggiungere una propria feature `myFeature` al transfer state.

```ts
// Esempio illustrativo, non presente nel repo
provideConfig(<StateConfig>{
  state: {
    ssrTransfer: {
      keys: { myFeature: StateTransferType.TRANSFER_STATE },
    },
  },
});
```

Per *escludere* una feature di default si può mappare la chiave a `undefined` (il tipo lo consente, core-libs/core/src/state/config/state-config.ts:L28-L31): `filterKeysByType` tiene solo le chiavi il cui valore è uguale a `'SSR'` (core-libs/core/src/state/utils/get-state-slice.ts:L104-L112). Che il merge di configurazione propaghi davvero `undefined` sopra il default dipende da `deepMerge`, che assegna i valori non-oggetto così come sono (core-libs/core/src/config/utils/deep-merge.ts:L36-L45); vedi capitolo 11.

7. Commenti e note notevoli dal sorgente

- `/** Higher order reducer that adds generic loading flag to chunk of the state … Utilizes "loader" meta field … */` — core-libs/core/src/state/utils/loader/loader.reducer.ts:L18-L23. È la dichiarazione del principio "meta, non type".
- `// processes reset action is a reset action for loader reducer, but not the other way around` — core-libs/core/src/state/utils/processes-loader/processes-loader.action.ts:L50-L50. Conseguenza

in §9 (T11).

- `//CXSPA-3670 - extract logging to an effect` e il messaggio "There should always be only one decrement action for each increment action. Make sure that you don't reset state in between those actions." — core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L43-L50. Debito tecnico dichiarato: un `console.error` dentro un reducer.
- `// Do not change order of subscription! Read should happen before write on context change.` — core-libs/core/src/state/services/state-persistence.service.ts:L63-L63.
- JSDoc di `persistToStorage`: rifiuta `undefined` perché `JSON.stringify(undefined)` non produce JSON valido — core-libs/core/src/state/utils/browser-storage.ts:L39-L46.
- JSDoc di `isUserLoggedIn`: "(required by transfer state reducer)" e invito a usare `AuthService` negli altri casi — core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L115-L118.
- `// if we want to reuse CMS_FEATURE const in config, we have to use factory instead of plain object` — core-libs/core/src/cms/store/cms-store.module.ts:L22-L22. Spiega perché i default di `ssrTransfer` sono factory.
- `/* With ProcessesLoaderResetAction we run it on every entity of this type. */` — feature-libs/cart/base/core/store/actions/cart.action.ts:L238-L241. Uso intenzionale del "broadcast" di `entityReducer` (T7).
- JSDoc di `StateEventService.createEvent`: afferma che il `payload` è usato "come argomento del costruttore dell'evento" — core-libs/core/src/state/event/state-event.service.ts:L57-L67; in realtà `createFrom` chiama `new type()` senza argomenti e poi `Object.assign` (core-libs/core/src/util/create-from.ts:L15-L17).
- NgRx: il commento su `META_REDUCERS` lo riserva ai meta-reducer "defined either internally by @ngrx/store or by library authors" — `deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L248-L251`. Spartacus lo usa esattamente in questo ruolo di library author.
- Nessun `@deprecated` nei file del capitolo (dossier, sezione N).

8. Extension point

COSA VUOI FARE	MECCANISMO	DOVE NEL CODICE	RICETTA
Trasferire da SSR una slice della propria feature	config `state.ssrTransfer.keys`	core-libs/core/src/state/config/state-config.ts:L20-L35	50 — Ricette config/DI
Persistere una slice in local/session storage	`StatePersistenceService.syncWithStorage` (pubblico)	core-libs/core/src/state/services/state-persistence.service.ts:L44-L93	53 — Ricette feature/cart
Cambiare il formato della chiave di storage	estendere `StatePersistenceService` e sovrascrivere `protected generateKeyWithContext`, fornendolo via DI	core-libs/core/src/state/services/state-persistence.service.ts:L124-L131	50
Nuova slice con stato di caricamento per id	`StateUtils.entityLoaderReducer` + classi azione `StateUtils.Entity*Action` sovrascrivendo `type`	core-libs/core/src/state/utils/entity-loader/entity-loader.reducer.ts:L13-L25, core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L71-L108	53

COSA VUOI FARE	MECCANISMO	DOVE NEL CODICE	RICETTA
Tracciare un'operazione senza slice dedicata	feature `process` con un nuovo process id + `ProcessSelectors`	core-libs/core/src/process/store/process-state.ts:L9-L13, core-libs/core/src/process/store/selectors/index.ts:L7-L8	53
Contare operazioni concorrenti su un'entità	`StateUtils.entityProcessesLoaderReducer` + `EntityProcesses{Increment,Decrement}Action`	core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader.reducer.ts:L13-L25, core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader.action.ts:L60-L88	53
Esporre un'azione NgRx come evento Spartacus	`StateEventService.register({action, event, factory?})`	core-libs/core/src/state/event/state-event.service.ts:L36-L41	50 (eventi custom)
Aggiungere un proprio meta-reducer radice	provider multi `META_REDUCERS` di `@ngrx/store` (stesso meccanismo di Spartacus)	core-libs/core/src/state/reducers/index.ts:L38-L42	50
Sostituire il meta-reducer di transfer state	**sconsigliato**: `TRANSFER_STATE_META_REDUCER` non è esportato da `@spartacus/core` (core-libs/core/src/state/index.ts:L7-L11; dossier F+15)	core-libs/core/src/state/reducers/index.ts:L23-L25	54 — API pubblica vs interna
Usare scoped loader fuori dal prodotto	**non pubblico**: `scoped-loader` non è in `StateUtils` né nei barrel (core-libs/core/src/state/utils/utils-group.ts:L7-L14)	core-libs/core/src/state/utils/scoped-loader/scoped-loader.reducer.ts:L27-L43	54

Simboli pubblici (barrel): `StateModule`, `StateConfig`, `StorageSyncType`, `StateTransferType`, `StatePersistenceService`, `StateEventService`, `ActionToEventMapping`, `StateUtils`, `ProcessModule`, `PROCESS_FEATURE`, `StateWithProcess`, `ProcessSelectors` (dossier, sezione E). Non pubblici: `TRANSFER_STATE_META_REDUCER`, `CX_KEY`, `getTransferStateReducer`, `getStorage`/`persistToStorage`/`readFromStorage`, tutto `scoped-loader`.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **I reducer generici rispondono al `meta`, non al `type`.** Qualunque azione con `meta.entityType` e `meta.loader` corrispondenti viene interpretata, anche se il `type` è "custom" (core-libs/core/src/state/utils/loader/loader.reducer.ts:L32-L36). Due feature che riusano lo stesso `entityType` si pestano i piedi.
2. **LOAD non azzera `error`/`success`.** Durante un ricaricamento, `success: true` (o `error: true`) resta visibile insieme a `loading: true` (core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L44). Un componente che mostra "errore" quando `error === true` lo mostrerà anche durante il retry.
3. **FAIL cancella** `value` se non c'è reducer interno (core-libs/core/src/state/utils/loader/loader.reducer.ts:L45-L52).
4. **Ordine dei controlli**: `load` → `error` → `success` → reset. `loader: {}` è un reset; `failMeta` sostituisce un errore falsy con `true`, quindi una FAIL non può degenerare in reset (core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L69, core-libs/core/src/state/utils/loader/loader.action.ts:L40-L47).
5. **Con un reducer interno il `payload` della SUCCESS non viene usato automaticamente**: `getReducerIfExists` restituisce il risultato del reducer interno (core-libs/core/src/state/utils/loader/

loader.reducer.ts:L56-L56, :L80-L82).

6. **Broadcast di** `entityReducer`**.** Un'azione senza `entityId` (o con `entityId: undefined`, come produce `entityResetMeta(type)` senza id) viene applicata a **tutte** le entità (core-libs/core/src/state/utils/entity/entity.reducer.ts:L42-L44, core-libs/core/src/state/utils/entity/entity.action.ts:L20-L25). È voluto per `ResetCartDetails` (feature-libs/cart/base/core/store/actions/cart.action.ts:L238-L257), ma è anche il motivo per cui ogni azione dello store passa per ogni entità (costo O(n) per azione con reducer interno). Con un reducer interno di tipo loader, inoltre, se esiste almeno un'entità la slice (es. `process`, `cart.carts`) riceve un nuovo riferimento a ogni azione, perché ogni risultato truthy entra negli update (core-libs/core/src/state/utils/entity/entity.reducer.ts:L53-L63, core-libs/core/src/state/utils/loader/loader.reducer.ts:L72-L78).
7. `entityId: null` **senza** `entityRemove` **diventa l'id** `''` per effetto di `?? ''` (core-libs/core/src/state/utils/entity/entity.reducer.ts:L33-L33).
8. **Reducer interno che restituisce falsy**: l'entità non viene aggiornata né creata (core-libs/core/src/state/utils/entity/entity.reducer.ts:L53-L56).
9. **Contatore dei processi negativo senza log.** Se `processesCount` è `0` e arriva un decrement, il controllo dev-mode non scatta (richiede `state.processesCount` truthy) e il contatore diventa `-1`; un valore negativo è truthy, quindi `isStableSelector` restituisce `false` finché un increment non riporta il contatore a `0` o arriva un reset dei processi; da quel momento il contatore resta sfasato (core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L53-L66, core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L37-L42, :L56-L58, core-libs/core/src/state/utils/processes-loader/processes-loader.selectors.ts:L9-L11).
10. **Un reset del loader non azzera** `processesCount`: serve `processesCountDiff: null`, cioè `ProcessesLoaderResetAction`/`EntityProcessesLoaderResetAction` (core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L53-L66, core-libs/core/src/state/utils/processes-loader/processes-loader.action.ts:L47-L55).
11. **Id inesistente = stabile.** `entityIsStableSelector` su un id assente usa `initialLoaderState` (senza `processesCount`) → `true` (core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader.selectors.ts:L19-L45, core-libs/core/src/state/utils/entity-loader/entity-loader.selectors.ts:L17-L22).
12. **Eventi senza factory**: l'evento è creato con `new type()` senza argomenti; costruttori con parametri obbligatori non ricevono il payload (core-libs/core/src/util/create-from.ts:L15-L17).
13. **Transfer state ignorato per l'utente loggato**, valutato **una sola volta** alla creazione del meta-reducer (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L26-L30, :L81-L81).
14. **Il server riscrive** `cx-state` **a ogni azione** (costo proporzionale al numero di azioni durante il render) (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L52-L57).
15. **Merge browser: gli array trasferiti sostituiscono quelli iniziali**, gli oggetti si fondono ricorsivamente: `deepMerge` tratta gli array come valori scalari (core-libs/core/src/config/utils/deep-merge.ts:L7-L9, :L36-L45).
16. **Valori falsy, oggetti vuoti e primitivi senza chiavi (booleani, numeri) non vengono trasferiti**: `createShellObject` restituisce `{}` quando `Object.keys(value)` è vuoto (core-libs/core/src/state/utils/get-state-slice.ts:L28-L30).
17. **Le chiavi devono esistere anche nella config del browser**: la slice ricevuta viene ri-filtrata con le chiavi del client (core-libs/core/src/state/reducers/transfer-state.reducer.ts:L83-L87).
18. **Slice trasferite e chiavi orfane in** `combineReducers`**.** Al momento di INIT esistono solo i reducer di `forRoot` (3.1); la slice trasferita resta nello stato come chiavi "senza reducer". `combineReducers`, alla prima azione che cambia qualche chiave, costruisce un nuovo oggetto con **solo** le chiavi registrate (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L319-L343). Ne segue, per deduzione, che una slice trasferita

per una feature registrata *dopo* un cambiamento di stato andrebbe persa. DA VERIFICARE a runtime: dipende dall'ordine effettivo di istanziazione degli store module (core eager vs feature lazy), che non si ricava dai soli file del capitolo.

19. **Meta-reducer registrati in moduli lazy.** Il reducer factory radice (token interno di NgRx prodotto da `createReducerFactory`) è costruito una volta nell'injector root a partire da `META_REDUCERS` (`deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1337-L1353`). `CartPersistenceModule` registra su `META_REDUCERS` un meta-reducer che reagisce a INIT (`feature-libs/cart/base/core/cart-persistence.module.ts:L39-L52`, `:L67-L71`), ma nell'app demo è raggiunto solo attraverso `CartBaseWrapperModule`, caricato con `import()` (`projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L28-L32`, `projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L22-L25`, `feature-libs/cart/base/cart-base.module.ts:L13-L13`, `feature-libs/cart/base/core/cart-base-core.module.ts:L24-L28`). DA VERIFICARE: se in quella configurazione il meta-reducer venga effettivamente applicato (il provider vive nell'injector del modulo lazy, creato dopo INIT); approfondimento nel capitolo 25 e, per gli injector lazy, nel capitolo 12. I meta-reducer *di feature* (es. `multiCartMetaReducers` passato a `forFeature`) non hanno questo problema perché viaggiano con la feature (`feature-libs/cart/base/core/store/multi-cart-store.module.ts:L25-L27`, `deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L406-L416`).
20. **Storage**: `readFromStorage` lancia su JSON corrotto (`core-libs/core/src/state/utils/browser-storage.ts:L65-L70`); in SSR `onRead` è comunque chiamato con `undefined` e la scrittura non è sottoscritta (`core-libs/core/src/state/services/state-persistence.service.ts:L67-L75`, `:L80-L90`); un tipo di storage sconosciuto ricade su `sessionStorage`, mentre il default del servizio è `localStorage` (`core-libs/core/src/state/utils/browser-storage.ts:L31-L33`); emissioni di `state$` prima della prima emissione di `context$` sono perse per via di `withLatestFrom` (`core-libs/core/src/state/services/state-persistence.service.ts:L82-L82`).
21. **I processi non si puliscono da soli**: lo stato di un process id resta finché non arriva un reset esplicito (es. `ResetSubscribeCustomerCouponProcess`) (`core-libs/core/src/user/store/actions/customer-coupon.action.ts:L136-L141`).
22. `StateModule` **senza** `forRoot` **non fa nulla**: importarlo in un modulo feature non registra il meta-reducer (`core-libs/core/src/state/state.module.ts:L10-L18`).

---

10. Come fare debug

- **Redux DevTools**: i `type` delle azioni feature sono leggibili (es. `'[Product] Load Product Data'`, costante `LOAD_PRODUCT`: `core-libs/core/src/product/store/actions/product.action.ts:L15-L15`), ma per capire *quale cella* cambia guardate `meta.entityType`, `meta.entityId`, `meta.loader`, `meta.processesCountDiff`, `meta.scope` (3.6-3.9).
- **Transfer state**: nell'HTML renderizzato dal server cercate lo `<script id="<appId>-state">` e la chiave `cx-state` (`angular:packages/core/src/transfer_state.ts:L150-L157`, `core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16`). Se lo stato non viene riusato nel browser, verificate nell'ordine: (a) `config.state.ssrTransfer.keys` contiene la chiave anche nel browser (`:L24-L24`, `:L83-L87`); (b) nello storage non c'è un `token.access_token` (`core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L119-L121`); (c) la slice non è un oggetto vuoto (`core-libs/core/src/state/utils/get-state-slice.ts:L28-L30`).
- **Ispezionare lo stato di un processo**: `store.select(ProcessSelectors.getProcessStateFactory('<id>'))` restituisce l'intera cella `LoaderState` (`core-libs/core/src/process/store/selectors/process.selectors.ts:L18-L26`).

- **Carrello che non diventa "stabile"**: controllate `processesCount` dell'entità nello stato `cart.carts.entities[<id>]`; un valore negativo indica decrement non bilanciati (§9.9); in dev mode cercate in console il messaggio di `CXSPA-3670` (core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L45-L51).
- **Storage**: le chiavi hanno il prefisso `spartacus⚿` (core-libs/core/src/state/services/state-persistence.service.ts:L128-L130); un valore corrotto fa lanciare `JSON.parse` (§9.20).
- **Meta-reducer**: per verificare quali meta-reducer radice sono attivi, mettete un breakpoint in `createReducerFactory` di NgRx e ispezionate `metaReducers` (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L359-L365).
- **Test di mapping azione→evento**: `testActionToEventMapping` (non esportato dai barrel) mostra il pattern: sottoscrivere `eventService.get(...)` con `take(1)`, fare `actions$.next(action)`, confrontare (core-libs/core/src/state/event/testing-utils/test-action-to-event-mapping.ts:L12-L31).

---

11. Domande di autoverifica

1. **Chi crea lo store radice NgRx in un'app Spartacus?** L'applicazione, con `StoreModule.forRoot({})` nell'`AppModule`, inserito dallo schematic `add-spartacus`. Spartacus aggiunge solo feature e meta-reducer. — projects/storefrontapp/src/app/app.module.ts:L15-L22, core-libs/schematics/src/add-spartacus/store.ts:L58-L65.
2. **Perché il meta-reducer browser del transfer state agisce solo su INIT, e con quale stato?** Perché INIT è il valore iniziale di `ActionsSubject` e viene ridotta alla costruzione di `State`, quando esistono solo i reducer di `forRoot`; il meta-reducer pre-popola lo stato con la slice trasferita che le feature troveranno come stato precedente. — deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L196-L200, :L485-L498; core-libs/core/src/state/reducers/transfer-state.reducer.ts:L76-L91.
3. **In che ordine NgRx applica `META_REDUCERS` e i `metaReducers` di `forRoot`?** Prima i `META_REDUCERS`, poi quelli utente (`concat`); il primo dell'array è il più esterno. — deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1069-L1071, :L349-L371.
4. **Un utente loggato riceve lo stato CMS renderizzato dal server?** No: se `isUserLoggedIn()` (presenza di `token.access_token` in storage) è vero alla creazione del meta-reducer, la slice non viene fusa. — core-libs/core/src/state/reducers/transfer-state.reducer.ts:L26-L30, :L81-L81; core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L119-L121.
5. **Cosa succede a `error` e `success` quando arriva una nuova LOAD?** Restano invariati; cambia solo `loading: true`. — core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L44.
6. **Come si resetta lo stato di tutti i carrelli in un colpo solo?** Con un'azione senza id (`ResetCartDetails extends ProcessesLoaderResetAction`): `entityReducer` la applica a tutte le chiavi esistenti. — feature-libs/cart/base/core/store/actions/cart.action.ts:L238-L257, core-libs/core/src/state/utils/entity/entity.reducer.ts:L42-L44.
7. **Perché un `LoaderResetAction` non basta a sbloccare un carrello con `processesCount` sporco?** Perché il reset del loader non porta `processesCountDiff: null`; solo il reset dei processi azzera il contatore. — core-libs/core/src/state/utils/processes-loader/processes-loader.action.ts:L47-L55, core-libs/core/src/state/utils/processes-loader/processes-loader.reducer.ts:L53-L66.
8. **Dove finisce nello storage lo stato sincronizzato con `key: 'cart'` e contesto `electronics`?** Sotto `spartacus⚿electronics⚿cart`, in `localStorage` per default. — core-libs/core/src/state/services/state-persistence.service.ts:L28-L34, :L48-L48, :L124-L131.

9. **Posso usare `entityScopedLoaderReducer` in una mia feature importandolo da `@spartacus/core`?** No: lo scoped loader non è in `StateUtils` né in altri barrel pubblici. — core-libs/core/src/state/utils/utils-group.ts:L7-L14.
10. **Come si trasforma un'azione NgRx in evento senza scrivere una factory, e che vincolo impone sulla classe evento?** `StateEventService.register({action, event})`: l'evento è `Object.assign(new Event(), payload ?? {})`, quindi la classe deve essere istanziabile senza argomenti. — core-libs/core/src/state/event/state-event.service.ts:L68-L76, core-libs/core/src/util/create-from.ts:L15-L17.

20

Facade, command & query, eventi

Facade proxy lazy, command con strategie di concorrenza, query con invalidazione a eventi, `EventService`.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 01 — Dependency injection, 12 — Feature module e lazy loading in Spartacus, 19 — State management. Collegati: 14 — Routing (route semantiche), 25 — Carrello e 26 — Checkout (i principali utilizzatori di facade, command e query).

1. A cosa serve e perché esiste

Una feature Spartacus espone ai componenti (e alle altre feature) un'API stabile e indipendente da *come* i dati vengono caricati. Questo capitolo copre i quattro mattoni con cui quell'API viene costruita e con cui le feature comunicano fra loro senza dipendenze dirette:

1. **Facade (proxy lazy).** Una facade è una classe astratta che i componenti iniettano. Con `facadeFactory` la facade viene fornita nel root injector come un *proxy* che, alla prima chiamata di metodo (o alla prima sottoscrizione di una proprietà), scarica il modulo lazy della feature e inoltra la chiamata all'implementazione reale (`core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L22-L36`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24`). È ciò che consente a un header di iniettare `ActiveCartFacade` senza trascinare nel bundle iniziale tutto il codice del carrello (`feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46`).
2. **Command.** Un'astrazione per le operazioni di scrittura (POST/PUT/DELETE verso OCC) con una *strategia di concorrenza* esplicita: in coda, in parallelo, annulla la precedente, metti in errore la precedente (`core-libs/core/src/util/command-query/command.service.ts:L27-L38`). Il comando parte anche se nessuno si sottoscrive al risultato (`core-libs/core/src/util/command-query/command.service.ts:L122-L132`).
3. **Query.** Un'astrazione per le letture con stato `{loading, error, data}`, caricamento lazy alla sottoscrizione e invalidazione dichiarativa tramite eventi (`reloadOn`, `resetOn`) (`core-libs/core/src/util/command-query/query.service.ts:L33-L60`). Insieme a Command è l'alternativa "senza NgRx" usata dalle feature più recenti (es. checkout, `feature-libs/checkout/base/core/facade/checkout-query.service.ts:L32-L54`).
4. **Eventi.** Un bus tipizzato per classi (`EventService`): chiunque può *registrare* una sorgente observable per un tipo di evento, *dispatchare* un'istanza, o *ascoltare* un tipo (e quindi anche tutti i suoi sottotipi) (`core-libs/core/src/event/event.service.ts:L47-L113`). `StateEventService` trasforma azioni NgRx in eventi (`core-libs/core/src/state/event/state-event.service.ts:L16-L76`), e `@spartacus/storefront` aggiunge una catena di eventi di pagina (`NavigationEvent` → `HomePageEvent`, `ProductDetailsPageEvent`, …) pensata per tracking e personalizzazione (`core-libs/storefront/events/index.ts:L7-L10`).

I quattro pezzi sono intrecciati: le query usano eventi come trigger (`core-libs/core/src/util/command-query/query.service.ts:L144-L155`), i command dispatchano eventi di dominio dopo il successo (`feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L34-L64`), il lazy loading che sta dietro le facade notifica la creazione dei moduli con un evento (`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L99-L106`).

Tutti i simboli principali sono API pubblica di `@spartacus/core`: eventi (`core-libs/core/src/event/index.ts:L7-L8`), command/query e `createFrom` (`core-libs/core/src/util/index.ts:L10-L12`, `core-libs/core/public_api.ts:L35-L35`), facade factory (`core-libs/core/src/lazy-loading/facade-factory/index.ts:L7-L9`, `core-libs/core/public_api.ts:L24-L24`), `StateEventService` (`core-libs/core/src/state/event/index.ts:L7-L8`, `core-libs/core/public_api.ts:L33-L33`). Gli eventi di pagina sono pubblici da `@spartacus/storefront` (`core-libs/storefront/public_api.ts:L25-L25`). `MergingSubject`, il motore interno del bus, **non** è esportato ed è marcato `// PRIVATE API` (`core-libs/core/src/event/index.ts:L7-L8`, `core-libs/core/src/event/utils/merging-subject.ts:L10-L10`).

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/event/cx-event.ts	`CxEvent`: classe base astratta di tutti gli eventi, con il solo campo statico `type = 'CxEvent'` (:L12-L17).
core-libs/core/src/event/event.service.ts	`EventService`: `register`/`get`/`dispatch`, registrazione sui tipi padre, validazioni in dev mode (:L44-L223).
core-libs/core/src/event/utils/merging-subject.ts	`MergingSubject` (PRIVATE API): fonde dinamicamente N sorgenti in un `output$` con refcount (:L10-L141).
core-libs/core/src/event/index.ts	Barrel pubblico: `CxEvent`, `EventService` (:L7-L8).
core-libs/core/src/util/create-from.ts	`createFrom(type, data)`: `Object.assign(new type(), data)` (:L15-L17).
core-libs/core/src/state/event/action-to-event-mapping.ts	`ActionToEventMapping<T>`: `action`, `event`, `factory?` (:L9-L24).
core-libs/core/src/state/event/state-event.service.ts	`StateEventService`: azioni NgRx → sorgenti di eventi (:L16-L76).
core-libs/core/src/util/command-query/command.service.ts	`Command`, `CommandStrategy`, `CommandService` (:L27-L140).
core-libs/core/src/util/command-query/query.service.ts	`QueryNotifier`, `QueryState`, `Query`, `QueryService` (:L31-L160).
core-libs/core/src/util/command-query/index.ts	Barrel command/query (:L7-L8).
core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts	`FacadeDescriptor<T>` e i tipi che vincolano metodi/proprietà (:L10-L57).
core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts	Funzione `facadeFactory(descriptor)` (:L20-L24).
core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts	`FacadeFactoryService`: costruzione del proxy, resolver, `call`/`get` (:L20-L141).
core-libs/core/src/lazy-loading/feature-modules.service.ts	Risoluzione feature → `NgModuleRef` usata dal proxy (dettagli nel capitolo 12) (:L33-L100).
core-libs/core/src/lazy-loading/events/module-initialized-event.ts	`ModuleInitializedEvent` (:L16-L32).
core-libs/storefront/events/navigation/navigation.event.ts	`NavigationEvent` (:L13-L34).
core-libs/storefront/events/navigation/navigation-event.builder.ts	`NavigationEventBuilder`: `ROUTER_NAVIGATED` → `NavigationEvent` (:L23-L65).
core-libs/storefront/events/navigation/navigation-event.module.ts	`NavigationEventModule`: istanzia il builder (:L10-L15).
core-libs/storefront/events/page/page.events.ts	`PageEvent` astratta, porta `navigation` (:L13-L18).
core-libs/storefront/events/home/home-page.events.ts	`HomePageEvent` (:L12-L15).
core-libs/storefront/events/home/home-page-event.builder.ts	`HomePageEventBuilder`: `NavigationEvent` con `semanticRoute === 'home'` (:L14-L36).
core-libs/storefront/events/home/home-page-event.module.ts	`HomePageEventModule` (:L10-L15).

PATH	RUOLO
core-libs/storefront/events/product/product-page.events.ts	ProductDetailsPageEvent, CategoryPageResultsEvent, SearchPageResultsEvent (:L13-L42).
core-libs/storefront/events/product/product-page-event.builder.ts	ProductPageEventBuilder: PDP, categoria, ricerca (:L23-L125).
core-libs/storefront/events/product/product-page-event.module.ts	ProductPageEventModule (:L10-L15).
core-libs/storefront/events/index.ts e */index.ts	Barrel degli eventi storefront (core-libs/storefront/events/index.ts:L7-L10).

File di esempio citati: feature-libs/cart/base/root/facade/active-cart.facade.ts, feature-libs/cart/base/core/facade/facade-providers.ts, feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts, feature-libs/checkout/base/core/facade/checkout-query.service.ts, feature-libs/checkout/base/root/events/checkout-query-event.listener.ts, core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts, feature-libs/cart/base/core/event/cart-event.builder.ts.

---

3. Come funziona passo per passo

3.1 Il modello degli eventi: classi, non stringhe

Un evento Spartacus è un'istanza di una classe che (per convenzione) estende CxEvent. CxEvent non ha membri d'istanza: ha solo il campo statico type = 'CxEvent' e il commento "One event to rule them all" (core-libs/core/src/event/cx-event.ts:L7-L17). La chiave con cui il bus indicizza gli eventi è **la classe stessa** (il costruttore), non una stringa: lo stato di EventService è una Map privata da classe a EventMeta (core-libs/core/src/event/event.service.ts:L49-L52). Il campo statico type serve solo alla validazione in dev mode (§3.6).

Per costruire istanze da dati plain si usa createFrom(type, data), che invoca new type() **senza argomenti** e copia le proprietà con Object.assign (core-libs/core/src/util/create-from.ts:L15-L17). È il motivo per cui gli eventi Spartacus dichiarano campi e non costruttori parametrici (es. NavigationEvent, core-libs/storefront/events/navigation/navigation.event.ts:L13-L34).

3.2 EventService.register: aggiungere una sorgente

EventService è un singleton providedIn: 'root' (core-libs/core/src/event/event.service.ts:L44-L48). Per ogni tipo tiene un EventMeta con un inputSubject$ (inizialmente null) e un MergingSubject (core-libs/core/src/event/event.service.ts:L24-L34).

register(eventType, source$) (core-libs/core/src/event/event.service.ts:L68-L84):

1. recupera (o crea) il meta del tipo con getEventMeta (core-libs/core/src/event/event.service.ts:L69-L69); la creazione, in dev mode, è preceduta dalla validazione del tipo (core-libs/core/src/event/event.service.ts:L133-L142);
2. se la stessa istanza di observable è già registrata per quel tipo non la aggiunge e, in dev mode, logga un warning (core-libs/core/src/event/event.service.ts:L70-L78); altrimenti mergingSubject.add(source$) (core-libs/core/src/event/event.service.ts:L80-L80);
3. restituisce una funzione di teardown che rimuove la sorgente (core-libs/core/src/event/event.service.ts:L83-L83). Il JSDoc impone di chiamarla (es. in ngOnDestroy) quando chi ha registrato la sor-

gente non la mantiene più, per evitare memory leak (core-libs/core/src/event/event.service.ts:L57-L59).

Registrare **non sottoscrive** la sorgente: `MergingSubject.add` si abbona subito solo se c'è già un consumer attivo, altrimenti accoda e basta (core-libs/core/src/event/utils/merging-subject.ts:L71-L80).

3.3 La gerarchia: chi ascolta il padre riceve i figli

Alla creazione del meta di un tipo, `createEventMeta` risale la prototype chain della classe e registra l'`output$` del nuovo tipo come sorgente di **ogni** classe antenata (core-libs/core/src/event/event.service.ts:L144-L159). La condizione d'uscita (`Object.getPrototypeOf(parentEvent) !== Object.getPrototypeOf({})`) ferma il ciclo quando il candidato è `Function.prototype`, il cui prototipo è `Object.prototype`: per `HomePageEvent` vengono quindi registrati `PageEvent` e `CxEvent` (dossier F+10). Il comportamento è documentato come `@since 3.1` (core-libs/core/src/event/event.service.ts:L61-L61) e coperto dallo spec "should register the parent class" con un padre astratto `CartEvent` (core-libs/core/src/event/event.service.spec.ts:L176-L189).

Il collegamento avviene alla creazione del meta del sottotipo, ma questo non crea problemi di ordine: qualsiasi emissione di un sottotipo richiede `register` o `dispatch` su quel tipo, entrambi passano per `getEventMeta` (core-libs/core/src/event/event.service.ts:L68-L69, core-libs/core/src/event/event.service.ts:L120-L126), e se il consumer del padre è già attivo `MergingSubject.add` si abbona subito (core-libs/core/src/event/utils/merging-subject.ts:L76-L78) (dossier F+14).

3.4 `EventService.get`: ascoltare

`get(eventType)` restituisce `mergingSubject.output$` del tipo (creando il meta se manca); in dev mode lo avvolge con un `tap` di validazione (core-libs/core/src/event/event.service.ts:L90-L96). `output$` è un `Observable` custom seguito da `share()` (core-libs/core/src/event/utils/merging-subject.ts:L45-L58):

- alla **prima** sottoscrizione (il solo "consumer" interno è il subscriber di `share()`, core-libs/core/src/event/utils/merging-subject.ts:L46-L47) memorizza il consumer e si abbona a tutte le sorgenti registrate (core-libs/core/src/event/utils/merging-subject.ts:L51-L52, core-libs/core/src/event/utils/merging-subject.ts:L85-L89);
- ogni sorgente inoltra **solo** le notifiche `next` (core-libs/core/src/event/utils/merging-subject.ts:L102-L105): `error` e `complete` delle sorgenti non arrivano mai a chi ascolta;
- quando l'ultimo sottoscrittore se ne va (refcount di `share()` a zero) annulla il consumer e si disabbona da tutte le sorgenti (core-libs/core/src/event/utils/merging-subject.ts:L54-L57, core-libs/core/src/event/utils/merging-subject.ts:L95-L97);
- una nuova sottoscrizione successiva ripete il ciclo (core-libs/core/src/event/utils/merging-subject.ts:L36-L44).

Le sorgenti sono quindi "attive" solo mentre qualcuno ascolta. Questo è un pilastro del design: un builder di eventi costoso (es. che legge lo store) non lavora finché nessuno è interessato.

3.5 `EventService.dispatch`: emettere un'istanza

`dispatch(event, eventType?)` (core-libs/core/src/event/event.service.ts:L105-L113):

1. senza `eventType`, il tipo è `event.constructor` (core-libs/core/src/event/event.service.ts:L106-L107);
2. con `eventType` e un oggetto che non ne è istanza, lo converte con `createFrom` (core-libs/core/src/event/event.service.ts:L108-L110);
3. prende l'input subject del tipo e fa `next` (core-libs/core/src/event/event.service.ts:L111-L112).

L'input subject è un `Subject` semplice creato al primo `dispatch` e registrato come una qualsiasi altra sorgente tramite `register` (core-libs/core/src/event/event.service.ts:L120-L128; il commento `// will be created lazily by the dispatch method` è a core-libs/core/src/event/event.service.ts:L146-L146). Essendo un `Subject` non-replay, e dato che le sorgenti sono collegate al consumer solo se ce n'è uno (§3.4), **un evento dispatchato quando nessuno ascolta quel tipo (o un suo antenato) è perso** (vedi §9).

3.6 Validazioni in dev mode

Tutte le validazioni sono sotto `isDevMode()`, che in Angular è vero se `ngDevMode` è `undefined` o truthy (angular:packages/core/src/util/is_dev_mode.ts:L20-L22):

- `validateEventType`: lancia `Error` solo se `eventType` è nullish o non ha `constructor` (in pratica `null`/`undefined`); un valore non-classe ma con `constructor` (es. una stringa o un oggetto plain) passa e produce al più un warning di `validateCxEvent` (core-libs/core/src/event/event.service.ts:L166-L174);
- `validateCxEvent`: risale la catena cercando una classe con `type === 'CxEvent'`; se non la trova emette **solo un warning** (core-libs/core/src/event/event.service.ts:L181-L197);
- `getValidatedEventStream`: `tap` che avvisa se un valore emesso non è `instanceof` del tipo chiesto (core-libs/core/src/event/event.service.ts:L204-L222).

In produzione nessuno di questi controlli viene eseguito.

3.7 `StateEventService`: azioni NgRx → eventi

`StateEventService` (root) dipende da `ActionsSubject` di NgRx e da `EventService` (core-libs/core/src/state/event/state-event.service.ts:L16-L26). Il mapping è un `ActionToEventMapping<T>`: `action` (stringa o array), `event` (classe o `null`), `factory?` (core-libs/core/src/state/event/action-to-event-mapping.ts:L9-L24).

1. `register(mapping)` delega a `eventService.register(mapping.event, getFromAction(mapping))` e restituisce il teardown (core-libs/core/src/state/event/state-event.service.ts:L36-L41).
2. `getFromAction` filtra `ActionsSubject` con `ofType(...)` sulle action normalizzate in array (core-libs/core/src/state/event/state-event.service.ts:L47-L55). `ofType` è importato da `@ngrx/effects` (core-libs/core/src/state/event/state-event.service.ts:L8-L8) ed è un semplice `filter` che confronta `action.type` con la stringa (o con il `.type` di un ActionCreator) senza trasformare l'action (deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L245-L254).
3. `createEvent` usa la `factory` se presente, altrimenti `createFrom(eventType, action.payload ?? {})` (core-libs/core/src/state/event/state-event.service.ts:L68-L76).

Esempi reali: `UserAuthEventBuilder` mappa `AuthActions.LOGIN` → `LoginEvent` senza factory, mentre deriva `LogoutEvent` da `isUserLoggedIn()` con `pairwise` registrando direttamente su `EventService` (core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L40-L63); `CartEventBuilder` usa una `factory` per aggiungere `cartCode` al payload di `DELETE_CART` (feature-libs/cart/base/core/event/cart-event.builder.ts:L137-L145).

Nota NgRx: `ActionsSubject` è un `BehaviorSubject` inizializzato con `{ type: '@ngrx/store/init' }` e `Store.dispatch` fa `actionsObserver.next(action)` (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L196-L200, deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L598-L603). Ogni nuova sottoscrizione riceve quindi subito l'**ultima** action dispatchata: se coincide con una di quelle mappate, il primo consumer di un evento derivato da azioni riceve un evento "di replay" (vedi §9).

3.8 Command

`Command<PARAMS, RESULT>` è una classe astratta con il solo `execute(parameters): Observable<RESULT>` (core-libs/core/src/util/command-query/command.service.ts:L27-L29). `CommandService` (root, `OnDestroy`) li fabbrica (core-libs/core/src/util/command-query/command.service.ts:L40-L44).

`create(commandFactory, options?)` (core-libs/core/src/util/command-query/command.service.ts:L50-L135):

1. crea due `Subject`: `commands$` per i parametri e `results$` per i `ReplaySubject` che notificheranno ciascun chiamante (core-libs/core/src/util/command-query/command.service.ts:L54-L55);
2. definisce `notify(notifier$)`, un observer parziale che inoltra solo `next`/`error`/`complete`: dopo RxJS 7.3 passare il subject direttamente a `tap` inoltrerebbe anche l'handler `unsubscribe` (commento con link alla PR RxJS #6527, core-libs/core/src/util/command-query/command.service.ts:L57-L68);
3. costruisce la pipeline `zip(commands$, results$)` + operatore di flattening scelto in base alla strategia (core-libs/core/src/util/command-query/command.service.ts:L72-L118):
   - `CancelPrevious` / `ErrorPrevious` → `switchMap` (core-libs/core/src/util/command-query/command.service.ts:L73-L93); nel `finalize`, se il notifier non è già chiuso né in errore, `ErrorPrevious` lo chiude con `Error('Canceled by next command')`, `CancelPrevious` con `complete()` (core-libs/core/src/util/command-query/command.service.ts:L80-L90);
   - `Parallel` → `mergeMap` (core-libs/core/src/util/command-query/command.service.ts:L96-L105);
   - `Queue` e **default** (nessuna strategia) → `concatMap` (core-libs/core/src/util/command-query/command.service.ts:L107-L117);
   - in ogni ramo la factory è avvolta in `defer`, il risultato inoltrato al notifier via `tap(notify(...))`, e un `catchError(() => EMPTY)` impedisce che l'errore di un comando chiuda l'intera pipeline: l'errore arriva solo al notifier di quel comando (core-libs/core/src/util/command-query/command.service.ts:L99-L102);
4. **sottoscrive subito** la pipeline e la accumula in `this.subscriptions` (core-libs/core/src/util/command-query/command.service.ts:L120-L120);
5. ritorna un'istanza di una sottoclasse anonima di `Command` il cui `execute` crea un `ReplaySubject`, lo spinge su `results$`, spinge i parametri su `commands$` e restituisce il `ReplaySubject` (core-libs/core/src/util/command-query/command.service.ts:L122-L132).

Poiché `zip` accoppia l'n-esimo parametro con l'n-esimo notifier, ogni chiamante riceve esattamente il risultato del proprio comando. Il comando parte all'interno di `execute`, indipendentemente dalla sottoscrizione del risultato: lo spec lo verifica con "should execute command without subscription" e conferma `Queue` come default (core-libs/core/src/util/command-query/command.service.spec.ts:L92-L92, core-libs/core/src/util/command-query/command.service.spec.ts:L163-L163). `ngOnDestroy` del servizio chiude tutte le pipeline (core-libs/core/src/util/command-query/command.service.ts:L137-L139).

Fuori da `CommandService`, nel codice non-spec del repo `CancelPrevious` compare 24 volte e `Queue` 7; `ErrorPrevious` e `Parallel` non sono mai usate (compaiono solo nello `switch` di core-libs/core/src/util/command-query/command.service.ts:L72-L118) (dossier F51, conteggio via `git grep`). Un uso tipico: `CheckoutDeliveryAddressService` crea `createDeliveryAddressCommand` con `CancelPrevious`, chiama il connector e, in `tap`, dispatcha `CheckoutDeliveryAddressCreatedEvent` (feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L34-L64); i metodi pubblici della facade si limitano a `command.execute(...)` (feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L162-L172).

3.9 Query

Tipi pubblici (core-libs/core/src/util/command-query/query.service.ts:L31-L42): `QueryNotifier = Observable<unknown> | Type<CxEvent>`; `QueryState<T> = {loading, error: false | Error, data: T |`

undefined}; Query<RESULT, PARAMS> con get(...params) e getState(...params). QueryService è root, OnDestroy, e inietta EventService (core-libs/core/src/util/command-query/query.service.ts:L44-L50).

create(loaderFactory, options?) (core-libs/core/src/util/command-query/query.service.ts:L52-L142):

1. Stato iniziale {data: undefined, error: false, loading: true} in un BehaviorSubject (core-libs/core/src/util/command-query/query.service.ts:L61-L67).
2. onSubscribeLoad$: iif(() => state$.value.loading, of(undefined), EMPTY) — emette solo se lo stato è ancora loading, cioè alla prima sottoscrizione o se un caricamento precedente è stato interrotto da un unsubscribe (core-libs/core/src/util/command-query/query.service.ts:L69-L75).
3. Tre stream di trigger costruiti con getTriggersStream: loadTrigger$ fonde onSubscribeLoad$ (valutato per primo, come spiega il commento) con reloadOn e resetOn; resetTrigger$ e reloadTrigger$ separati (core-libs/core/src/util/command-query/query.service.ts:L77-L84). getTriggersStream converte le classi evento in eventService.get(trigger) e fonde tutto con merge; con lista vuota ritorna EMPTY (core-libs/core/src/util/command-query/query.service.ts:L144-L155).
4. loaderFactory() è invocata **una sola volta**, qui, e il suo observable è interrotto da un reset con takeUntil(resetTrigger$) (core-libs/core/src/util/command-query/query.service.ts:L86-L86).
5. load$: a ogni trigger imposta loading: true (preservando data), fa switchMap sul loader, su dato imposta {loading: false, error: false, data}, su errore {loading: false, error, data: undefined} e ritorna la sorgente (si risottoscrive ai trigger); il tutto è condiviso con share() (core-libs/core/src/util/command-query/query.service.ts:L88-L103).
6. Se reloadOn non è vuoto, una subscription **permanente** in this.subscriptions mette loading: true a ogni trigger, anche senza sottoscrittori della query (core-libs/core/src/util/command-query/query.service.ts:L105-L114).
7. Se resetOn non è vuoto, un'altra subscription permanente riporta lo stato a initialState (se non lo è già) (core-libs/core/src/util/command-query/query.service.ts:L116-L129).
8. query$ = using(() => load$.subscribe(), () => state$): sottoscrivere lo stato attiva load$ come "risorsa" che vive quanto il sottoscrittore (core-libs/core/src/util/command-query/query.service.ts:L131-L134).
9. get() restituisce data con distinctUntilChanged, getState() lo stato completo; i params sono ignorati (core-libs/core/src/util/command-query/query.service.ts:L136-L141).

Esempio: CheckoutQueryService crea la query sui dettagli del checkout con reloadOn: [CheckoutQueryReloadEvent] e resetOn: [CheckoutQueryResetEvent], restituiti da metodi protected sovrascrivibili (feature-libs/checkout/base/core/facade/checkout-query.service.ts:L32-L54). Un listener separato, CheckoutQueryEventListener, traduce eventi di dominio negli eventi di controllo: cambio lingua/valuta → reload; login, logout, merge/save/restore carrello, ordine piazzato → reset (feature-libs/checkout/base/root/events/checkout-query-event.listener.ts:L38-L62). I due eventi di controllo sono emessi con dispatch (Subject non-replay), ma non vanno persi perché le subscription permanenti di §3.9 punti 6-7 li ascoltano sempre (core-libs/core/src/util/command-query/query.service.ts:L105-L129).

3.10 Facade proxy con facadeFactory

Il descrittore FacadeDescriptor<T> ha facade (classe astratta) e feature obbligatori, methods, properties, async opzionali (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L10-L37). I tipi ausiliari MethodKeys/PropertyKeys limitano a compile time le chiavi a metodi che ritornano void | Observable e a proprietà Observable; StrictlyAllowedFacade<T> diventa never se la facade contiene membri di altro tipo (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L39-L57). Il JSDoc lo ribadisce: i ritorni non-observable vengono ignorati, e async serve a rendere pronto lo store NgRx lazy (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L19-L36).

Flusso, con `ActiveCartFacade` come esempio:

1. La facade astratta è dichiarata `@Injectable({ providedIn: 'root', useFactory: () => facadeFactory({...}) })` con feature `CART_BASE_CORE_FEATURE`, la lista esplicita dei metodi e `async: true` (feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46).
2. `facadeFactory` chiama `inject(FacadeFactoryService).create(descriptor)` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24); `inject()` di Angular richiede un injection context attivo (senza, `injectInjectorOnly` lancia `MISSING_INJECTION_CONTEXT`: angular:packages/core/src/di/injector_compatibility.ts:L93-L103), che `useFactory` fornisce.
3. `create` costruisce il resolver con `getResolver(feature, facade, async)` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L117-L117). Se la feature non è configurata (`isConfigured` è vero solo se la config, dopo la risoluzione degli alias, ha `module`, core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35) il resolver è un `throwError` permanente; altrimenti è `resolveFeature(feature)` → `moduleRef.injector.get(facadeClass)`, con `delay(0)` se `async`, e `shareReplay()` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57). Il resolver è cold: nessun caricamento avviene qui (spec "should not trigger lazy loading", core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.spec.ts:L118-L132).
4. Istanzia una **sottoclasse anonima** della facade astratta, assegna una funzione proxy per ogni metodo elencato e un observable per ogni proprietà elencata, e marca l'istanza con `proxyFacadeInstance = true` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L119-L130, costante a core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L20-L20, verifica con `isProxyFacadeInstance` a core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L138-L140).
5. Alla chiamata di un metodo, `call()` crea un `connectable(resolver$.pipe(map(service => service[method](...args))), {connector: ReplaySubject, resetOnDisconnect: false})` e fa **subito** `connect()`: il modulo viene scaricato e il metodo reale eseguito anche se il chiamante non si sottoscrive (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L82).
6. `resolveFeature('cartBaseCore')` risolve l'alias `cartBaseCore` → `cartBase` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100, feature-libs/cart/base/root/cart-base-root.module.ts:L45-L53), il cui `module` è l'import dinamico fornito dall'app (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L26-L33); la risoluzione è cachata per feature (core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73).
7. `LazyModulesService`, dopo aver creato il modulo e i suoi initializer, dispatcha `ModuleInitializedEvent` con `feature` e `moduleRef` (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L99-L106).
8. `moduleRef.injector.get(ActiveCartFacade)` trova nel modulo lazy il provider `{provide: ActiveCartFacade, useExisting: ActiveCartService}` e restituisce l'implementazione concreta, non il proxy root (feature-libs/cart/base/core/facade/facade-providers.ts:L27-L30, feature-libs/cart/base/core/cart-base-core.module.ts:L36-L36).
9. Il risultato del metodo reale viene appiattito con `switchMap`: se è observable viene inoltrato, altrimenti si ritorna `EMPTY` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L84-L91).
10. Le **proprietà** sono invece `resolver$.pipe(switchMap(service => service[property]))`: cold, il caricamento avviene solo alla sottoscrizione (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L103-L108).

Lo spec copre tutti questi casi: chiamata di metodo che innesca il lazy loading senza subscribe, proprietà non sottoscritta che non lo innesca, `async: true` che ritarda l'inizializzazione (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.spec.ts:L156-L187 per metodi e proprietà, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.spec.ts:L126-L137 per l'opzione `async`). Nel repo 62 file non-spec usano `facadeFactory(` (dossier F98).

3.11 Eventi di pagina dello storefront

La catena parte dal router store di NgRx:

1. Spartacus configura `StoreRouterConnectingModule.forRoot({ routerState: RouterState.Minimal, stateKey: ROUTING_FEATURE })` e, in `RoutingModule.forRoot()`, fornisce `RouterStateSerializer` con `useClass: CustomSerializer` (core-libs/core/src/routing/routing.module.ts:L77-L80, core-libs/core/src/routing/routing.module.ts:L89-L92). Senza un serializer esplicito NgRx userebbe MinimalRouterStateSerializer (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L359-L366). I provider di `RoutingModule.forRoot()` (importato da `BaseStorefrontModule`, core-libs/storefront/base-storefront.module.ts:L41-L41) sono raccolti dopo quelli dei suoi import: `walkProviderTree` elabora prima gli import, poi i `ModuleWithProviders` importati come `StoreRouterConnectingModule.forRoot(...)` (angular:packages/core/src/di/provider_collection.ts:L293-L321, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L414-L420), poi i provider del modulo stesso (angular:packages/core/src/di/provider_collection.ts:L346-L354), e restituisce `true` per un `ModuleWithProviders` così che il genitore ne visiti i `providers` dopo (angular:packages/core/src/di/provider_collection.ts:L360-L362). Per i token non-multi `R3Injector` sovrascrive il record e vince l'ultimo (angular:packages/core/src/di/r3_injector.ts:L521-L521): quindi vince `CustomSerializer` (vedi capitolo 19 e capitolo 14); il funzionamento di `HomePageEventBuilder`, che legge `semanticRoute`, lo presuppone.
2. Su `NavigationEnd` (se la navigazione non è stata innescata dallo store) il router store serializza `router.routerState.snapshot` e dispatcha `{type: ROUTER_NAVIGATED, payload: {routerState, event}}` (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L245-L253, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L279-L305); `ROUTER_NAVIGATED` vale `'@ngrx/router-store/navigated'` (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L32-L33). In modalità `Minimal`, `payload.event` è ridotto a `{id, url, urlAfterRedirects}` (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L283-L305).
3. `CustomSerializer.serialize` produce un `ActivatedRouterStateSnapshot` con `url`, `queryParams`, `params`, `context`, `cmsRequired`, `semanticRoute` (core-libs/core/src/routing/store/routing-state.ts:L18-L25); `semanticRoute` viene da `data.cxRoute` di una route antenata o, in mancanza, da `lookupSemanticRoute(urlString)` (core-libs/core/src/routing/store/reducers/router.reducer.ts:L125-L155).
4. `NavigationEventBuilder` (root) registra nel costruttore una sorgente per `NavigationEvent` (core-libs/storefront/events/navigation/navigation-event.builder.ts:L23-L39): filtra `ActionsSubject` con `ofType<RouterNavigatedAction<ActivatedRouterStateSnapshot>>(ROUTER_NAVIGATED)` e legge `payload.routerState` (core-libs/storefront/events/navigation/navigation-event.builder.ts:L57-L64). Il generico di `RouterNavigatedAction` è solo una dichiarazione di tipo del payload (deps:ngrx-router-store-21.0.1/package/types/ngrx-router-store.d.ts:L129-L139): nessun controllo runtime. L'evento copia `context`, `semanticRoute`, `url`, `params` — non `queryParams` né `cmsRequired` (core-libs/storefront/events/navigation/navigation-event.builder.ts:L44-L55).
5. `HomePageEventBuilder` ascolta `NavigationEvent`, filtra `semanticRoute === 'home'` e crea `HomePageEvent` con `navigation` (core-libs/storefront/events/home/home-page-event.builder.ts:L22-L35). `HomePageEvent extends PageEvent` (core-libs/storefront/events/home/home-page.events.ts:L12-L15), e `PageEvent` è astratta e porta il `NavigationEvent` originario (core-libs/storefront/events/page/page.events.ts:L13-L18).
6. `ProductPageEventBuilder` (dipende da `EventService`, `ProductService`, `ProductSearchService`) registra tre sorgenti (core-libs/storefront/events/product/product-page-event.builder.ts:L23-L48):
   - **PDP**: `semanticRoute === 'product'` → `switchMap` su `productService.get(context.id)` → primo prodotto truthy (`take(1)`) → `ProductDetailsPageEvent` con categorie, codice, nome, prezzo (core-libs/storefront/events/product/product-page-event.builder.ts:L50-L69);

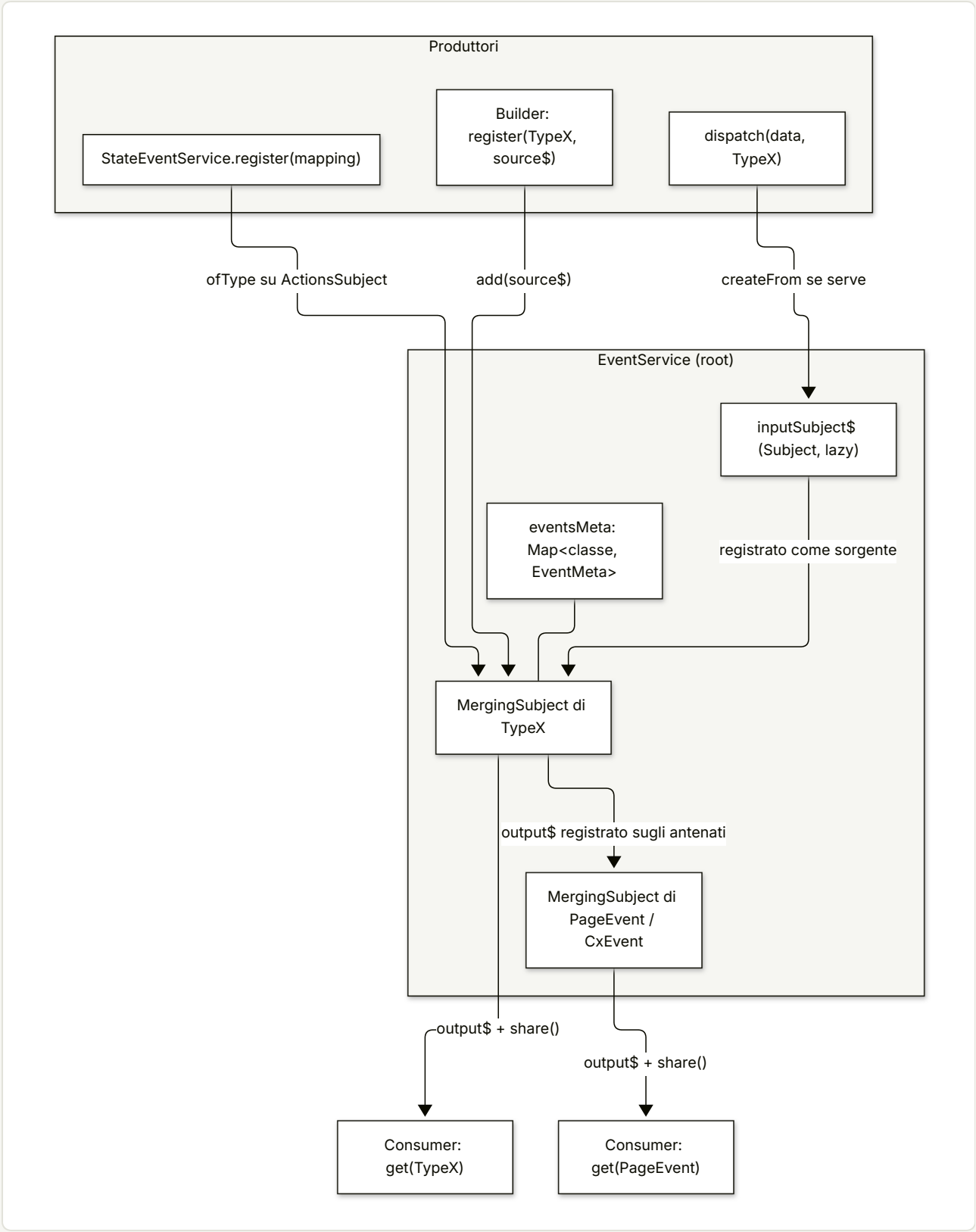
- **Categoria**: per ogni `NavigationEvent`, `switchMap` che ritorna `EMPTY` se la route non è `'category'`, altrimenti mappa `productSearchService.getResults().pipe(skip(1))` a `CategoryPageResultsEvent` (core-libs/storefront/events/product/product-page-event.builder.ts:L71-L97);
- **Ricerca**: stessa struttura con `'search'` e `searchTerm = freeTextSearch ?? ''` (core-libs/storefront/events/product/product-page-event.builder.ts:L99-L124). `getResults()` è una `select` sullo store (core-libs/core/src/product/facade/product-search.service.ts:L32-L34); lo `skip(1)` salta il valore corrente (iniziale o della ricerca precedente), come spiega il commento (core-libs/storefront/events/product/product-page-event.builder.ts:L73-L73).

7. I builder sono `providedIn: 'root'` ma nessuno li inietta da solo: vengono istanziati dai moduli vuoti `NavigationEventModule`, `HomePageEventModule`, `ProductPageEventModule`, che li ricevono nel costruttore (core-libs/storefront/events/navigation/navigation-event.module.ts:L10-L15, core-libs/storefront/events/home/home-page-event.module.ts:L10-L15, core-libs/storefront/events/product/product-page-event.module.ts:L10-L15). L'app demo li importa in `SpartacusFeaturesModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L239-L242) e lo schematic `add-spartacus` li aggiunge al modulo generato (core-libs/schematics/src/add-spartacus/spartacus-features.ts:L241-L243).

Grazie alla gerarchia di §3.3, un consumer di `PageEvent` riceve `HomePageEvent`, `ProductDetailsPageEvent`, `CategoryPageResultsEvent` e `SearchPageResultsEvent` (core-libs/core/src/event/event.service.ts:L151-L158).

4. Diagrammi

4.1 Struttura interna di `EventService`



Riferimenti: `register` core-libs/core/src/event/event.service.ts:L68-L84; input subject core-libs/core/src/event/event.service.ts:L120-L128; registrazione sugli antenati core-libs/core/src/event/event.service.ts:L151-L158; `output$` core-libs/core/src/event/utils/merging-subject.ts:L45-L58; `StateEventService` core-libs/core/src/state/event/state-event.service.ts:L36-L55.

4.2 Catena degli eventi di pagina

■ **TAVOLA 20.1** diagramma a tutta pagina, in fondo al capitolo

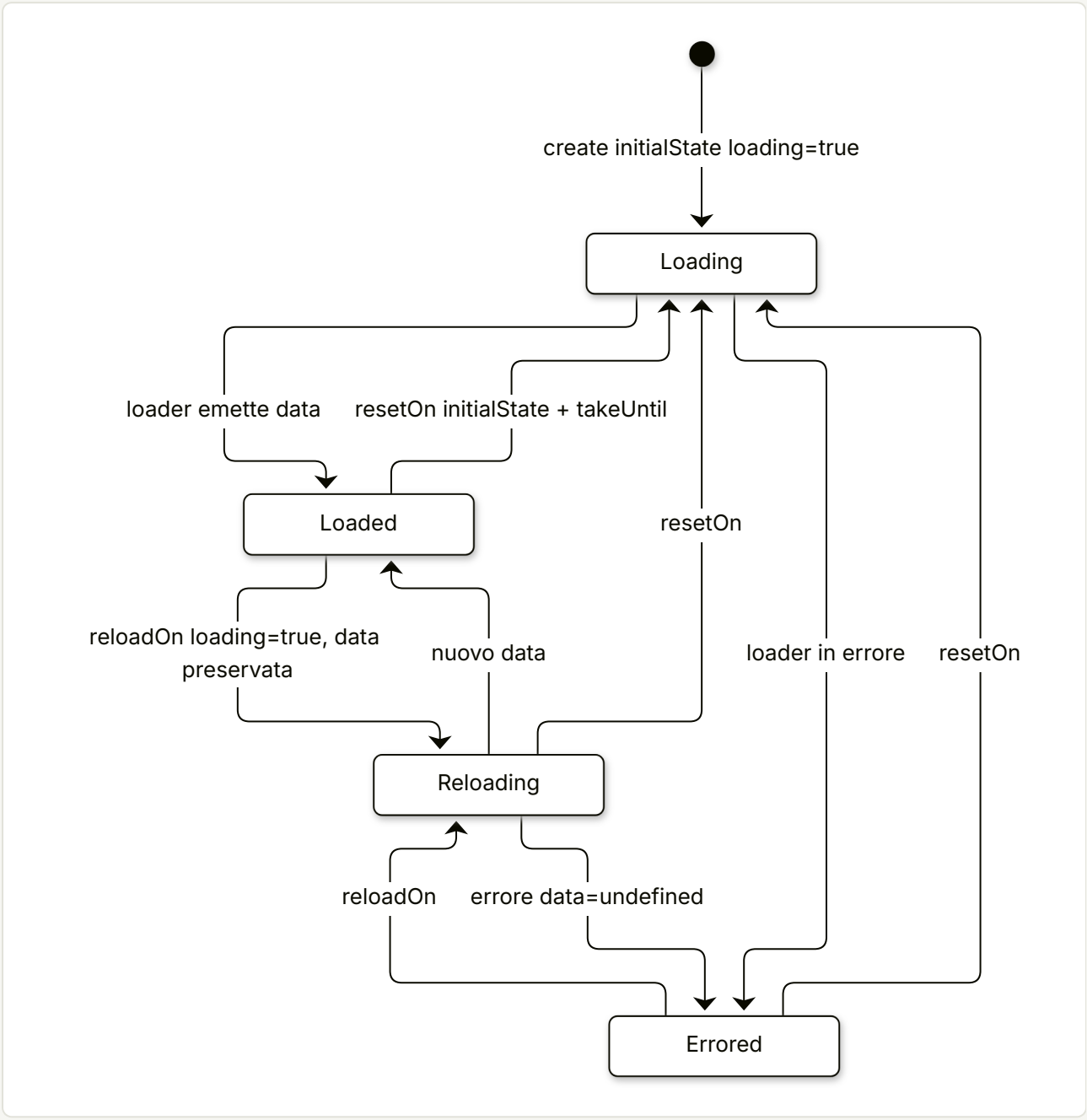
Riferimenti: deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L245-L253; core-libs/storefront/events/navigation/navigation-event.builder.ts:L44-L64; core-libs/storefront/events/home/home-page-event.builder.ts:L26-L35; core-libs/storefront/events/product/product-page-event.builder.ts:L50-L69. La sottoscrizione a cascata (consumer → sorgente → `get` del builder a monte) discende da core-libs/core/src/event/utils/merging-subject.ts:L45-L58.

4.3 Facade proxy: prima chiamata di metodo

■ **TAVOLA 20.2** diagramma a tutta pagina, in fondo al capitolo

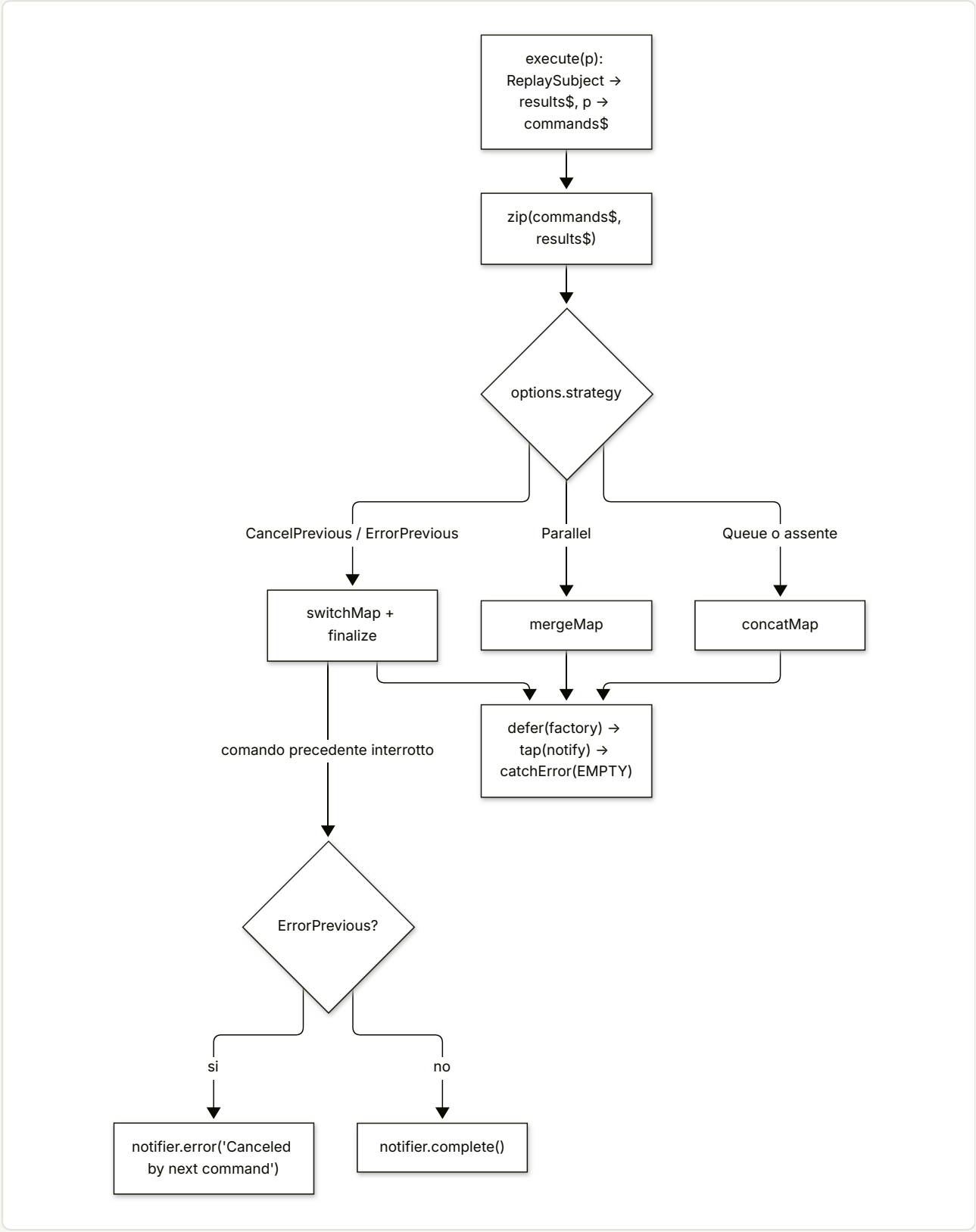
Riferimenti: feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46; core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L131; core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L100; core-libs/core/src/lazy-loading/lazy-modules.service.ts:L99-L106; feature-libs/cart/base/core/facade/facade-providers.ts:L27-L30.

4.4 Stati di una Query



Riferimenti: `core-libs/core/src/util/command-query/query.service.ts:L61-L67`, `core-libs/core/src/util/command-query/query.service.ts:L86-L129`. `Reloading` non è uno stato distinto nel codice: è `loading: true` con `data` ancora valorizzato (`core-libs/core/src/util/command-query/query.service.ts:L89-L93`).

4.5 Strategie di Command



Riferimenti: `core-libs/core/src/util/command-query/command.service.ts:L72-L132`.

5. Estratti di codice reale

Registrazione sugli antenati — core-libs/core/src/event/event.service.ts:L144-L159

```ts
private createEventMeta<T>(eventType: AbstractType<T>): void {
  const eventMeta: EventMeta<T> = {
    inputSubject$: null, // will be created lazily by the `dispatch` method
    mergingSubject: new MergingSubject<T>(),
  };
  this.eventsMeta.set(eventType, eventMeta);

  let parentEvent = Object.getPrototypeOf(eventType);
  while (
    parentEvent !== null &&
    Object.getPrototypeOf(parentEvent) !== Object.getPrototypeOf({})
  ) {
    this.register(parentEvent, eventMeta.mergingSubject.output$);
    parentEvent = Object.getPrototypeOf(parentEvent);
  }
}
```

dispatch — core-libs/core/src/event/event.service.ts:L105-L113

```ts
dispatch<T extends object>(event: T, eventType?: Type<T>): void {
  if (!eventType) {
    eventType = event.constructor as Type<T>;
  } else if (!(event instanceof eventType)) {
    event = createFrom(eventType, event);
  }
  const inputSubject$ = this.getInputSubject(eventType);
  inputSubject$.next(event);
}
```

MergingSubject.output$ — core-libs/core/src/event/utils/merging-subject.ts:L45-L58

```ts
readonly output$: Observable<T> = new Observable<T>((consumer) => {
  // There can be only 0 or 1 consumer of this observable coming from the `share()` operator
  // that is piped right after this observable.
  // `share()` not only multicasts the results but also  When all end-subscribers unsubscribe from
`share()` operator, it will unsubscribe
  // from this observable (by the nature `refCount`-nature of the `share()` operator).

  this.consumer = consumer;
  this.bindAllSourcesToConsumer(consumer);

  return () => {
    this.consumer = null;
```

```
      this.unbindAllSourcesFromConsumer();
    };
  }).pipe(share());
```

Binding: solo `next` — core-libs/core/src/event/utils/merging-subject.ts:L102-L105

```ts
  private bindSourceToConsumer(source: Observable<T>, consumer: Subscriber<T>) {
    const subscriptionToSource = source.subscribe((val) => consumer.next(val)); // passes all emissions
from source to consumer
    this.subscriptionsToSources.set(source, subscriptionToSource);
  }
```

Azioni → eventi — core-libs/core/src/state/event/state-event.service.ts:L36-L41

```ts
  register<T>(mapping: ActionToEventMapping<T>): () => void {
    return this.eventService.register(
      mapping.event as Type<T>,
      this.getFromAction(mapping)
    );
  }
```

`ofType` **di NgRx** — deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L245-L254

```js
function ofType(...allowedTypes) {
    return filter((action) => allowedTypes.some((typeOrActionCreator) => {
        if (typeof typeOrActionCreator === 'string') {
            // Comparing the string to type
            return typeOrActionCreator === action.type;
        }
        // We are filtering by ActionCreator
        return typeOrActionCreator.type === action.type;
    }));
}
```

Strategia a switch di `CommandService` — core-libs/core/src/util/command-query/command.service.ts:L72-L94

```ts
    switch (options?.strategy) {
      case CommandStrategy.CancelPrevious:
      case CommandStrategy.ErrorPrevious:
        process$ = zip(commands$, results$).pipe(
          switchMap(([cmd, notifier$]) =>
            defer(() => commandFactory(cmd)).pipe(
              tap(notify(notifier$)),
              catchError(() => EMPTY),
              finalize(() => {
                // do not overwrite existing existing ending state
```

```
            if (!notifier$.closed && !notifier$.hasError) {
              // command has not ended yet, so close notifier$ according to strategy
              if (options.strategy === CommandStrategy.ErrorPrevious) {
                notifier$.error(new Error('Canceled by next command'));
              } else {
                notifier$.complete();
              }
            }
          })
        )
      )
    );
    break;
```

execute — core-libs/core/src/util/command-query/command.service.ts:L120-L132

```ts
this.subscriptions.add(process$.subscribe());

const command: Command<PARAMS, RESULT> = new (class extends Command<
  PARAMS,
  RESULT
> {
  execute = (parameters: PARAMS) => {
    const result$ = new ReplaySubject<RESULT>();
    results$.next(result$);
    commands$.next(parameters);
    return result$;
  };
})();
```

Cuore di `QueryService.create` — core-libs/core/src/util/command-query/query.service.ts:L86-L103 e core-libs/core/src/util/command-query/query.service.ts:L131-L141

```ts
const loader$ = loaderFactory().pipe(takeUntil(resetTrigger$));

const load$ = loadTrigger$.pipe(
  tap(() => {
    if (!state$.value.loading) {
      state$.next({ ...state$.value, loading: true });
    }
  }),
  switchMap(() => loader$),
  tap((data) => {
    state$.next({ loading: false, error: false, data });
  }),
  catchError((error, sourceStream$) => {
    state$.next({ loading: false, error, data: undefined });
    return sourceStream$;
```

```
    }),
    share()
  );
  // ...
  const query$ = using(
    () => load$.subscribe(),
    () => state$
  );

  const data$ = query$.pipe(
    map((queryState) => queryState.data),
    distinctUntilChanged()
  );

  return { get: () => data$, getState: () => query$ };
```

Proxy della facade — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L92

```ts
protected call(
  resolver$: Observable<any>,
  method: string,
  args: unknown[]
): Observable<unknown> {
  const callResult$ = connectable(
    resolver$.pipe(map((service) => service[method](...args))),
    {
      connector: () => new ReplaySubject(),
      resetOnDisconnect: false,
    }
  );
  callResult$.connect();

  return callResult$.pipe(
    switchMap((result) => {
      if (isObservable(result)) {
        return result;
      }
      return EMPTY;
    })
  );
}
```

Costruzione del proxy — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L110-L131

```ts
create<T extends object>({
  facade,
  feature,
  methods,
  properties,
  async,
}: FacadeDescriptor<T>): T {
  const resolver$ = this.getResolver(feature, facade, async);

  const result: any = new (class extends (facade as any) {})();
  (methods ?? []).forEach((method) => {
    result[method] = (...args: any[]) =>
      this.call(resolver$, method as string, args);
  });
  (properties ?? []).forEach((property) => {
    result[property] = this.get(resolver$, property as string);
  });

  result[PROXY_FACADE_INSTANCE_PROP] = true;

  return result;
}
```

Sorgente di `NavigationEvent` — core-libs/storefront/events/navigation/navigation-event.builder.ts:L57-L64

```ts
private getNavigatedEvent(): Observable<ActivatedRouterStateSnapshot> {
  return this.actions.pipe(
    ofType<RouterNavigatedAction<ActivatedRouterStateSnapshot>>(
      ROUTER_NAVIGATED
    ),
    map((event) => event.payload.routerState)
  );
}
```

Risultati di categoria — core-libs/storefront/events/product/product-page-event.builder.ts:L71-L97

```ts
protected buildCategoryResultsPageEvent(): Observable<CategoryPageResultsEvent> {
  const searchResults$ = this.productSearchService.getResults().pipe(
    // skipping the initial value, and preventing emission of the previous search state
    skip(1)
  );

  return this.eventService.get(NavigationEvent).pipe(
    switchMap((navigationEvent) => {
      if (navigationEvent?.semanticRoute !== 'category') {
        return EMPTY;
      }
```

```
      return searchResults$.pipe(
        map((searchResults) =>
          createFrom(CategoryPageResultsEvent, {
            navigation: navigationEvent,
            ...{
              categoryCode: navigationEvent?.context?.id,
              numberOfResults: searchResults?.pagination?.totalResults ?? 0,
              categoryName: searchResults.breadcrumbs?.[0].facetValueName,
            },
          })
        )
      );
    })
  );
}
```

6. Configurazione

Questo sottosistema non ha chiavi in `provideConfig`: si configura con argomenti/descrittori nel codice e, per le facade, con la config `featureModules` del capitolo 12.

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`FacadeDescriptor.facade`	`AbstractType<StrictlyAllowedFacade<T>>`	obbligatoria	core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L14-L14	core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L117-L119
`FacadeDescriptor.feature`	`string`	obbligatoria	core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L18-L18	core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L49
`FacadeDescriptor.methods`	`MethodKeys<T>[]`	`[]` (via `?? []`)	core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L25-L25	core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L120-L123
`FacadeDescriptor.properties`	`PropertyKeys<T>[]`	`[]` (via `?? []`)	core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L31-L31	core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L124-L126
`FacadeDescriptor.async`	`boolean`	`false` (parametro `async = false`)	core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L36-L36	core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L41-L55

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`featureModules[name].module`	`() => Promise<any>`	—	capitolo 12	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35`
`featureModules[name]` come stringa	alias	—	es. `feature-libs/cart/base/root/cart-base-root.module.ts:L53-L53`	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100`
`options.strategy` (`CommandService.create`)	`CommandStrategy`	`Queue` (ramo `default`)	`core-libs/core/src/util/command-query/command.service.ts:L52-L52`	`core-libs/core/src/util/command-query/command.service.ts:L72-L118`
`options.reloadOn` (`QueryService.create`)	`QueryNotifier[]`	`[]`	`core-libs/core/src/util/command-query/query.service.ts:L56-L56`	`core-libs/core/src/util/command-query/query.service.ts:L79-L84`, `core-libs/core/src/util/command-query/query.service.ts:L106-L114`
`options.resetOn` (`QueryService.create`)	`QueryNotifier[]`	`[]`	`core-libs/core/src/util/command-query/query.service.ts:L58-L58`	`core-libs/core/src/util/command-query/query.service.ts:L80-L86`, `core-libs/core/src/util/command-query/query.service.ts:L117-L129`
`ActionToEventMapping.action` / `event` / `factory`	`string | string[]` / `Type<T> | null` / funzione	senza `factory` → `createFrom(event, payload ?? {})`	`core-libs/core/src/state/event/action-to-event-mapping.ts:L13-L23`	`core-libs/core/src/state/event/state-event.service.ts:L47-L76`
`isDevMode()` (Angular)	`boolean`	`true` se `ngDevMode` è `undefined`	`angular:packages/core/src/util/is_dev_mode.ts:L20-L22`	`core-libs/core/src/event/event.service.ts:L71-L71`, `core-libs/core/src/event/event.service.ts:L92-L92`, `core-libs/core/src/event/event.service.ts:L135-L135`
`data.cxRoute` (route data)	`string`	—	es. `feature-libs/cart/base/root/cart-base-root.module.ts:L69-L69`	`core-libs/core/src/routing/store/reducers/router.reducer.ts:L126-L128` → `NavigationEvent.semanticRoute`
`routerState` del router store	`RouterState`	`Minimal` in Spartacus	`core-libs/core/src/routing/routing.module.ts:L77-L80`	`deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L283-L305` (forma di `payload.event`)

7. Commenti e note notevoli dal sorgente

- `"One event to rule them all".` — la dichiarazione d'intenti di `CxEvent` (core-libs/core/src/event/cx-event.ts:L10-L10).
- `CAUTION: To avoid memory leaks, the returned teardown function should be called ...` — il contratto di `register` (core-libs/core/src/event/event.service.ts:L57-L59).
- `@since 3.1 - registers the given source$ for the parent classes ...` — la gerarchia è stata aggiunta dopo; prima di 3.1 l'ascolto di un padre non riceveva i figli (core-libs/core/src/event/event.service.ts:L61-L61).
- `// PRIVATE API` su `MergingSubject` (core-libs/core/src/event/utils/merging-subject.ts:L10-L10) e l'avvertenza `In particular closed event sources won't be automatically removed.` (core-libs/core/src/event/utils/merging-subject.ts:L20-L22).
- Commento troncato: `` `share()` not only multicasts the results but also When all end-subscribers unsubscribe ... `` — la frase è grammaticalmente incompleta nel sorgente (core-libs/core/src/event/utils/merging-subject.ts:L48-L49).
- `// SkipIfOngoing`, / `// ErrorIfOngoing` — strategie di comando previste ma mai implementate (core-libs/core/src/util/command-query/command.service.ts:L36-L37).
- Il commento su RxJS 7.3.0 e la PR #6527: perché `notify` elenca gli handler uno per uno (core-libs/core/src/util/command-query/command.service.ts:L57-L61).
- `// do not overwrite existing existing ending state` (con la ripetizione) (core-libs/core/src/util/command-query/command.service.ts:L81-L81).
- `// Intentional empty constructor` in `CommandService` (core-libs/core/src/util/command-query/command.service.ts:L46-L48) e nei moduli eventi (core-libs/storefront/events/navigation/navigation-event.module.ts:L10-L15).
- `/** Reloads the query, while preserving the data until the new data is loaded */` e `/** Resets the query to the initial state */` — la semantica di `reloadOn`/`resetOn` (core-libs/core/src/util/command-query/query.service.ts:L55-L58).
- `// we need to evaluate onSubscribeLoad$ before other triggers in order to avoid other triggers changing state$ value` (core-libs/core/src/util/command-query/query.service.ts:L78-L78).
- `Required to make lazy NgRx store feature ready.` — la ragione d'essere di `async` (core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L33-L35), e il `// TODO:#deprecation-checkout - remove once we remove ngrx` sopra `async: true` nelle facade di checkout (feature-libs/checkout/base/root/facade/checkout-delivery-address.facade.ts:L24-L25).
- Refuso `Property has to be an aobservable` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L97-L97).
- `// skipping the initial value, and preventing emission of the previous search state` (core-libs/storefront/events/product/product-page-event.builder.ts:L73-L73, ripetuto a core-libs/storefront/events/product/product-page-event.builder.ts:L101-L101).

8. Extension point

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
Evento custom	Classe che estende `CxEvent` (campi, niente costruttore parametrico), `static readonly type`	core-libs/core/src/event/cx-event.ts:L12-L17, core-libs/core/src/util/create-from.ts:L15-L17	Parte III, cap. 50
Sorgente di eventi custom	`EventService.register(Type, source$)` conservando il teardown	core-libs/core/src/event/event.service.ts:L68-L84	Parte III, cap. 50
Evento da azione NgRx	`StateEventService.register({action, event, factory?})`	core-libs/core/src/state/event/state-event.service.ts:L36-L41	Parte III, cap. 50
Ascolto di una famiglia di eventi	`eventService.get(PageEvent)` (o un altro padre astratto)	core-libs/core/src/event/event.service.ts:L151-L158	Parte III, cap. 50
Arricchire/sostituire gli eventi di pagina	Sottoclasse di `NavigationEventBuilder` / `HomePageEventBuilder` / `ProductPageEventBuilder` (metodi `register()`/`build*()` protected) fornita al posto dell'originale	core-libs/storefront/events/navigation/navigation-event.builder.ts:L37-L55, core-libs/storefront/events/product/product-page-event.builder.ts:L35-L124	Parte III, cap. 50
Facade per una feature lib custom	Classe astratta `@Injectable({providedIn: 'root', useFactory: () => facadeFactory({...})})` + nel modulo lazy `{provide: Facade, useExisting: Impl}`	feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46, feature-libs/cart/base/core/facade/facade-providers.ts:L27-L30	Parte III, cap. 53
Sostituire l'implementazione di una facade lazy	Fornire un altro provider per la classe facade nel modulo lazy (o sottoclassare il servizio concreto)	feature-libs/cart/base/core/facade/facade-providers.ts:L27-L30	Parte III, cap. 50
Command/Query in un servizio custom	`CommandService.create(factory, {strategy})`, `QueryService.create(loader, {reloadOn, resetOn})`	core-libs/core/src/util/command-query/command.service.ts:L50-L53, core-libs/core/src/util/command-query/query.service.ts:L52-L60	Parte III, cap. 53
Cambiare gli eventi che invalidano la query di checkout	Sottoclasse di `CheckoutQueryService` che sovrascrive `getCheckoutQueryReloadEvents()` / `getCheckoutQueryResetEvents()`	feature-libs/checkout/base/core/facade/checkout-query.service.ts:L32-L40	Parte III, cap. 53
Reagire al caricamento di un modulo lazy	`eventService.get(ModuleInitializedEvent)`	core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L32	Parte III, cap. 50
Da evitare	Importare `MergingSubject` (non esportato, PRIVATE API) o affidarsi a membri `private` di `EventService`	core-libs/core/src/event/index.ts:L7-L8, core-libs/core/src/event/event.service.ts:L52-L52	Parte III, cap. 54

Note sui punti di estensione:

- `StateEventService` è sottoclassabile: `getFromAction` e `createEvent` sono `protected` (core-libs/core/src/state/event/state-event.service.ts:L47-L76).
- `FacadeFactoryService` ha `getResolver`, `call`, `get` `protected` ed è `providedIn: 'root'`, quindi tecnicamente sostituibile via DI (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L29-L108); farlo cambia il comportamento di tutte le 62 facade del repo (dossier F98).
- `QueryService.getTriggersStream` è `protected` (core-libs/core/src/util/command-query/query.service.ts:L144-L155).

- Gli eventi di pagina sono opt-in: per averli bisogna importare i tre `*EventModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L239-L242).

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Eventi dispatchati senza ascoltatori sono persi.** `dispatch` usa un `Subject` non-replay e le sorgenti sono collegate solo se esiste un consumer (core-libs/core/src/event/event.service.ts:L123-L127, core-libs/core/src/event/utils/merging-subject.ts:L76-L78). Se un listener si sottoscrive "tardi" (es. dopo che un componente lazy è stato caricato) non vede gli eventi già emessi. `QueryService` evita il problema mantenendo subscription permanenti sui trigger (core-libs/core/src/util/command-query/query.service.ts:L105-L129).
2. **Le sorgenti lavorano solo mentre qualcuno ascolta.** Un builder che dipende dallo stato corrente vede solo ciò che accade dopo la sottoscrizione; con `skip(1)` su `getResults()` il primo valore (lo stato attuale dello store) viene scartato (core-libs/core/src/event/utils/merging-subject.ts:L45-L58, core-libs/storefront/events/product/product-page-event.builder.ts:L72-L75).
3. **Errori e complete delle sorgenti sono silenziosi.** `MergingSubject` inoltra solo `next`: una sorgente che va in errore smette di emettere senza che nessuno lo sappia e resta registrata (core-libs/core/src/event/utils/merging-subject.ts:L102-L105, core-libs/core/src/event/utils/merging-subject.ts:L20-L22).
4. **Deduplicazione per identità.** `register` ignora solo la *stessa istanza* di observable (core-libs/core/src/event/event.service.ts:L70-L81, core-libs/core/src/event/utils/merging-subject.ts:L138-L140): chiamare due volte un metodo `build*()` crea due observable diversi e quindi eventi doppi.
5. **Possibile doppia ricezione ascoltando un "nonno".** Il ciclo di `createEventMeta` registra l'`output$` di `HomePageEvent` sia su `PageEvent` sia su `CxEvent`, e anche `PageEvent` registra il proprio `output$` su `CxEvent` (core-libs/core/src/event/event.service.ts:L151-L158): per lettura del codice, `get(CxEvent)` dovrebbe ricevere due volte ogni `HomePageEvent`, mentre `get(PageEvent)` lo riceve una volta. `DA VERIFICARE` con un test runtime: lo spec copre solo gerarchie a due livelli (core-libs/core/src/event/event.service.spec.ts:L176-L189) e `node_modules` non è installato.
6. **Validazioni solo in dev mode e quasi solo warning.** Un evento che non estende `CxEvent` produce un warning in dev e nulla in produzione; lancia solo un `eventType` nullish o senza `constructor`, in pratica `null`/`undefined` (core-libs/core/src/event/event.service.ts:L135-L137, core-libs/core/src/event/event.service.ts:L166-L222).
7. `validateCxEvent` **guarda il campo** `type`**, non l'ereditarietà.** Una classe non derivata da `CxEvent` con `static type = 'CxEvent'` passerebbe la validazione (core-libs/core/src/event/event.service.ts:L187-L189). Correlato: `PageEvent` non ridefinisce `type` e quindi eredita `'CxEvent'` (core-libs/storefront/events/page/page.events.ts:L13-L18, core-libs/core/src/event/cx-event.ts:L16-L16).
8. `createFrom` **chiama** `new Type()` **senza argomenti.** Eventi con costruttore parametrico ricevono `undefined` quando creati da `dispatch(plainObject, Type)` o da `StateEventService` senza factory (core-libs/core/src/util/create-from.ts:L15-L17, core-libs/core/src/event/event.service.ts:L108-L110).
9. **Action senza payload → eventi vuoti.** `createFrom(eventType, action.payload ?? {})` (core-libs/core/src/state/event/state-event.service.ts:L73-L75). Il tipo di `ActionToEventMapping.event` ammette `null`, ma viene castato e passato a `EventService.register`; nel codice di produzione non esistono mapping con `event: null` (dossier F+11).
10. **Replay dell'ultima action.** `ActionsSubject` è un `BehaviorSubject` (deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L196-L200): quando il primo consumer attiva una sorgente basata su `ofType`, questa riceve subito l'ultima action dispatchata e, se corrisponde, emette un evento relativo a un'azione

avvenuta *prima* della sottoscrizione. Per `NavigationEvent` succede solo se l'ultima action in assoluto è proprio `ROUTER_NAVIGATED`.

11. `execute` **è hot.** Il comando parte dentro `execute` anche se nessuno sottoscrive il risultato; il `ReplaySubject` conserva tutte le emissioni (core-libs/core/src/util/command-query/command.service.ts:L126-L131). Sottoscrivere due volte il risultato **non** riesegue il comando.
12. `CancelPrevious` **≠ errore.** Il chiamante annullato riceve un `complete` senza valori; solo con `ErrorPrevious` riceve `Error('Canceled by next command')` (core-libs/core/src/util/command-query/command.service.ts:L80-L90). Codice che usa `firstValueFrom` sul risultato di un comando `CancelPrevious` annullato riceve un rifiuto `EmptyError` di RxJS, a meno di passare `firstValueFrom(result$, { defaultValue })` (deps:rxjs-7.8.2/package/dist/esm/internal/firstValueFrom.js:L12-L18).
13. **Pipeline dei comandi legate al root.** Vivono quanto `CommandService` (root): un comando creato in un servizio a vita breve non viene rilasciato alla sua distruzione (core-libs/core/src/util/command-query/command.service.ts:L120-L120, core-libs/core/src/util/command-query/command.service.ts:L137-L139).
14. `loaderFactory()` **è chiamata una sola volta.** Il codice imperativo nel corpo della factory (fuori dall'observable restituito) non viene rieseguito ai reload: va messo dentro l'observable (es. `defer`/`switchMap`) (core-libs/core/src/util/command-query/query.service.ts:L86-L86).
15. **Le query cambiano stato anche quando nessuno ascolta.** Le subscription di reload/reset vivono nel root; alla sottoscrizione successiva `onSubscribeLoad$` riparte se `loading` è `true` (core-libs/core/src/util/command-query/query.service.ts:L105-L129, core-libs/core/src/util/command-query/query.service.ts:L69-L75).
16. **Errore →** `data` **perso.** Lo stato di errore azzera `data` (core-libs/core/src/util/command-query/query.service.ts:L98-L101); la query resta comunque reattiva ai trigger successivi perché `catchError` ritorna la sorgente.
17. **Parametri di** `get`**/**`getState` **ignorati.** L'interfaccia `Query` li dichiara, l'implementazione li ignora (core-libs/core/src/util/command-query/query.service.ts:L39-L42, core-libs/core/src/util/command-query/query.service.ts:L141-L141).
18. `distinctUntilChanged` **per riferimento.** Un reload che restituisce un oggetto nuovo ma uguale emette comunque su `get()` (core-libs/core/src/util/command-query/query.service.ts:L136-L139).
19. **Metodo dimenticato nella lista =** `undefined`**.** Sul proxy esistono solo i membri elencati in `methods`/`properties`; la facade è astratta, quindi chiamare un metodo non elencato produce un `TypeError` a runtime (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L119-L126).
20. **Ritorni non-observable persi.** Un metodo proxato che ritorna, per esempio, un `boolean` restituisce `EMPTY`; anche i metodi `void` restituiscono un observable che non emette (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L84-L91, core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L21-L24).
21. **"Feature X is not configured properly" è definitivo.** `isConfigured` è valutato una sola volta, quando il proxy viene creato alla prima iniezione: se in quel momento la feature non ha `module`, il resolver è un `throwError` per tutta la vita dell'app (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L117-L117).
22. **Facade non ri-fornita nel modulo lazy.** Se il modulo lazy non fornisce la classe facade, `moduleRef.injector.get(facadeClass)` risale al root injector e ottiene il proxy stesso, che richiamerebbe sé stesso (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52). DA VERIFICARE: deduzione dalla gerarchia degli injector, nessun test nel repo.
23. **I metodi scaricano subito il chunk, le proprietà no.** `call()` fa `connect()` immediato, `get()` resta cold (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L75-L82, core-libs/

core/src/lazy-loading/facade-factory/facade-factory.service.ts:L103-L108): chiamare un metodo in un costruttore anticipa il download del modulo.

24. `categoryName` **con breadcrumbs vuoti.** `searchResults.breadcrumbs?.[0].facetValueName` lancia se `breadcrumbs` è un array vuoto, e `searchResults` è usato senza `?.` (core-libs/storefront/events/product/product-page-event.builder.ts:L90-L90). DA VERIFICARE se OCC restituisca mai `breadcrumbs: []` su una pagina categoria.
25. `numberOfResults: Number`**.** Tipizzato con il wrapper `Number` invece di `number` (core-libs/storefront/events/product/product-page.events.ts:L30-L30, core-libs/storefront/events/product/product-page.events.ts:L41-L41).
26. **Una sola** `ProductDetailsPageEvent` **per navigazione.** `take(1)` dopo il primo prodotto truthy; una nuova navigazione annulla l'attesa (`switchMap`) (core-libs/storefront/events/product/product-page-event.builder.ts:L53-L56, core-libs/core/src/product/facade/product.service.ts:L39-L46).
27. **Eventi di categoria/ricerca ripetuti.** Dopo il `switchMap`, ogni nuova emissione di `getResults()` sulla stessa pagina (paginazione, facet) genera un nuovo evento finché non arriva un'altra navigazione (core-libs/storefront/events/product/product-page-event.builder.ts:L77-L96). DA VERIFICARE se paginazione/facet producano una nuova `ROUTER_NAVIGATED` (e quindi un nuovo `switchMap`) o solo una nuova emissione dello store.
28. **Nessun modulo, nessun evento di pagina.** Senza `NavigationEventModule` & co. i builder non vengono istanziati e nessuna sorgente è registrata (core-libs/storefront/events/navigation/navigation-event.module.ts:L10-L15).
29. `HomePageEvent` **richiede la route semantica** `home`**.** Serve `data.cxRoute: 'home'` o un riconoscimento via `lookupSemanticRoute` (core-libs/storefront/events/home/home-page-event.builder.ts:L28-L28, core-libs/core/src/routing/store/reducers/router.reducer.ts:L126-L155).
30. `NavigationEvent` **non porta i query param.** Copia solo `context`, `semanticRoute`, `url`, `params` (core-libs/storefront/events/navigation/navigation-event.builder.ts:L44-L55); i `queryParams` sono nello snapshot ma vanno letti altrove.

---

10. Come fare debug

- **Vedere i warning del bus.** Le validazioni di `EventService` sono attive solo in dev mode (`isDevMode()`, angular:packages/core/src/util/is_dev_mode.ts:L20-L22); in sviluppo cercare in console i messaggi "has been already registered for the type" (registrazione doppia, core-libs/core/src/event/event.service.ts:L72-L77), "does not inherit from the CxEvent" (core-libs/core/src/event/event.service.ts:L194-L196) e "that is not an instance of the declared type" (core-libs/core/src/event/event.service.ts:L211-L218). Passano per `LoggerService` (core-libs/core/src/event/event.service.ts:L48-L48).
- **Tracciare tutto il traffico.** Sottoscrivere temporaneamente `eventService.get(CxEvent)` mostra ogni evento il cui tipo estende `CxEvent` (§3.3), con la possibile duplicazione descritta in §9 punto 5. Per gli eventi di pagina basta `get(PageEvent)`.
- **Evento atteso che non arriva.** Verificare nell'ordine: (1) il builder è istanziato? (moduli `*EventModule` importati, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L239-L242); (2) c'era un consumer quando l'evento è stato emesso? (§9 punti 1-2); (3) la sorgente è andata in errore? (`MergingSubject` lo nasconde, §9 punto 3); (4) per gli eventi di pagina, il `semanticRoute` nello stato router è quello atteso (core-libs/core/src/routing/store/reducers/router.reducer.ts:L125-L155): con Redux DevTools si vede la action `@ngrx/router-store/navigated` e il suo `payload.routerState` (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L32-L33).

- **Eventi da azioni NgRx.** Confrontare in DevTools il `type` esatto dell'action con la stringa del mapping: `ofType` fa un confronto `===` (`deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L245-L254`).
- **Facade.** Per capire se si sta parlando con il proxy o con l'implementazione: `inject(FacadeFactoryService).isProxyFacadeInstance(facade)` (`core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L138-L140`) o, in console, la proprietà `proxyFacadeInstance` sull'istanza. Il caricamento del modulo è osservabile con `eventService.get(ModuleInitializedEvent)` (`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L99-L106`) o nel tab Network (chunk scaricato alla prima chiamata di metodo). L'errore "Feature X is not configured properly" indica un `featureModules[X].module` mancante (`core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47`).
- **Command.** Ricordare che il risultato è un `ReplaySubject`: sottoscriverlo in ritardo non perde il valore (`core-libs/core/src/util/command-query/command.service.ts:L127-L130`). Un risultato che completa senza valori con `CancelPrevious` indica un comando annullato dal successivo (§9 punto 12).
- **Query.** Usare `getState()` invece di `get()` per vedere `loading` ed `error` (`core-libs/core/src/util/command-query/query.service.ts:L141-L141`); un `loading: true` persistente indica un loader che non emette né va in errore (per esempio una dipendenza che non emette mai). Le precondizioni non soddisfatte di `checkoutPreconditions` producono invece uno stato di errore: il metodo lancia `Error('Checkout conditions not met')` (`feature-libs/checkout/base/core/facade/checkout-query.service.ts:L66-L84`) e la query passa a `{loading: false, error, data: undefined}` (`core-libs/core/src/util/command-query/query.service.ts:L98-L101`).
- **Nei test.** Gli spec del repo sono una buona guida: `core-libs/core/src/event/event.service.spec.ts`, `core-libs/core/src/util/command-query/command.service.spec.ts`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.spec.ts`.

---

11. Domande di autoverifica

1. **Con quale chiave** `EventService` **indicizza i tipi di evento, e a cosa serve il campo statico** `type`**?** Con la classe stessa (una `Map` da costruttore a meta, `core-libs/core/src/event/event.service.ts:L49-L52`). `type` è usato solo dalla validazione in dev mode `validateCxEvent` (`core-libs/core/src/event/event.service.ts:L181-L197`).
2. **Registro una sorgente per** `AddToCartSuccessEvent` **e mi sottoscrivo a un padre astratto** `CartEvent`**. Ricevo gli eventi? Perché?** Sì: alla creazione del meta del sottotipo il suo `output$` viene registrato come sorgente di ogni antenato (`core-libs/core/src/event/event.service.ts:L151-L158`), comportamento verificato dallo spec "should register the parent class" (`core-libs/core/src/event/event.service.spec.ts:L176-L189`).
3. **Perché un evento passato a** `dispatch` **può andare perso?** Perché l'input subject è un `Subject` semplice (`core-libs/core/src/event/event.service.ts:L123-L127`) e `MergingSubject` si abbona alle sorgenti solo se ha un consumer (`core-libs/core/src/event/utils/merging-subject.ts:L71-L80`): senza sottoscrittori l'emissione non arriva a nessuno.
4. **Che evento produce** `StateEventService` **per un'action** `{type: 'X', payload: {a: 1}}` **mappata senza factory a** `MyEvent`**?** `createFrom(MyEvent, {a: 1})`, cioè `Object.assign(new MyEvent(), {a: 1})` (`core-libs/core/src/state/event/state-event.service.ts:L68-L76`, `core-libs/core/src/util/create-from.ts:L15-L17`). Il filtro avviene con `ofType`, un `filter` per uguaglianza di `type` (`deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L245-L254`).
5. **Qual è la strategia di default di** `CommandService.create`**, e cosa riceve il chiamante annullato con** `CancelPrevious`**?** `Queue` (`concatMap`, ramo `default`, `core-libs/core/src/util/command-query/command.`

service.ts:L107-L117). Con `CancelPrevious` il notifier annullato viene completato senza valori; con `ErrorPrevious` riceve `Error('Canceled by next command')` (core-libs/core/src/util/command-query/command.service.ts:L80-L90).

6. **Se chiamo `command.execute(p)` senza sottoscrivere il risultato, la chiamata HTTP parte?** Sì: la pipeline è sottoscritta in `create` (core-libs/core/src/util/command-query/command.service.ts:L120-L120) e `execute` spinge subito parametri e notifier (core-libs/core/src/util/command-query/command.service.ts:L126-L131); lo spec "should execute command without subscription" lo conferma (core-libs/core/src/util/command-query/command.service.spec.ts:L92-L92).
7. **Che differenza c'è fra un evento in `reloadOn` e uno in `resetOn` per una query?** `reloadOn` mette `loading: true` preservando `data` e ricarica; `resetOn` interrompe il loader (`takeUntil`), riporta lo stato a `initialState` (`data: undefined`) e ricarica (core-libs/core/src/util/command-query/query.service.ts:L55-L58, core-libs/core/src/util/command-query/query.service.ts:L86-L129).
8. **Quando viene scaricato il modulo lazy dietro una facade creata con `facadeFactory`?** Non all'iniezione (resolver cold, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57), ma alla prima chiamata di un metodo — `call()` fa `connect()` immediato (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L75-L82) — o alla prima sottoscrizione di una proprietà (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L103-L108).
9. **Perché `moduleRef.injector.get(ActiveCartFacade)` restituisce l'implementazione e non il proxy?** Perché il modulo core lazy fornisce `{provide: ActiveCartFacade, useExisting: ActiveCartService}` (feature-libs/cart/base/core/facade/facade-providers.ts:L27-L30, incluso in feature-libs/cart/base/core/cart-base-core.module.ts:L36-L36), che l'injector del modulo trova prima del provider root.
10. **Quale action NgRx alimenta `NavigationEvent`, quando viene emessa e quali campi dello snapshot vengono copiati?** `ROUTER_NAVIGATED` (`'@ngrx/router-store/navigated'`), dispatchata dal router store su `NavigationEnd` (deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L32-L33, deps:ngrx-router-store-21.0.1/package/fesm2022/ngrx-router-store.mjs:L245-L253). Il builder copia `context`, `semanticRoute`, `url`, `params` da `payload.routerState` (core-libs/storefront/events/navigation/navigation-event.builder.ts:L44-L64).

TAVOLA 20.1 **4.2 Catena degli eventi di pagina**

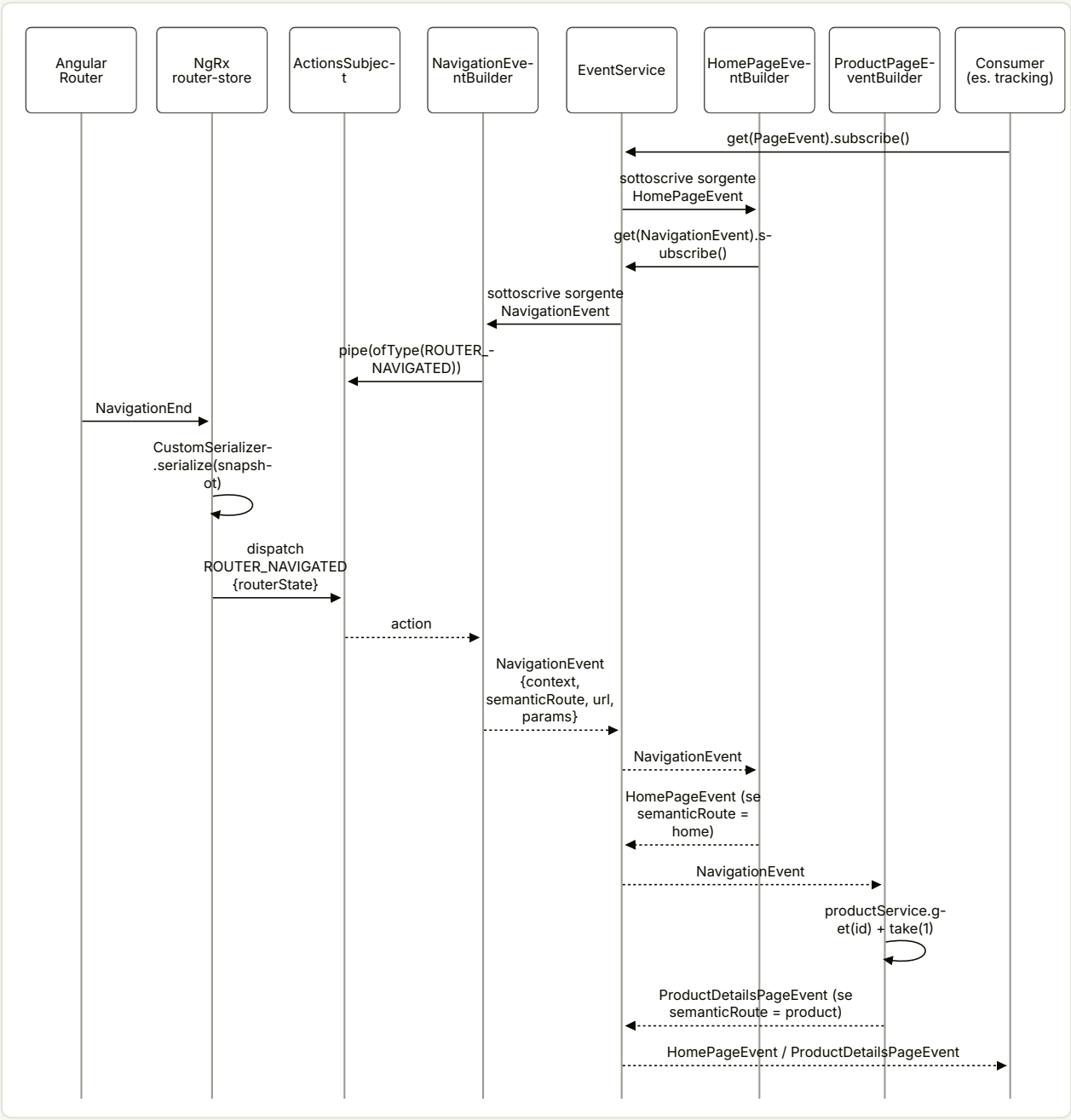
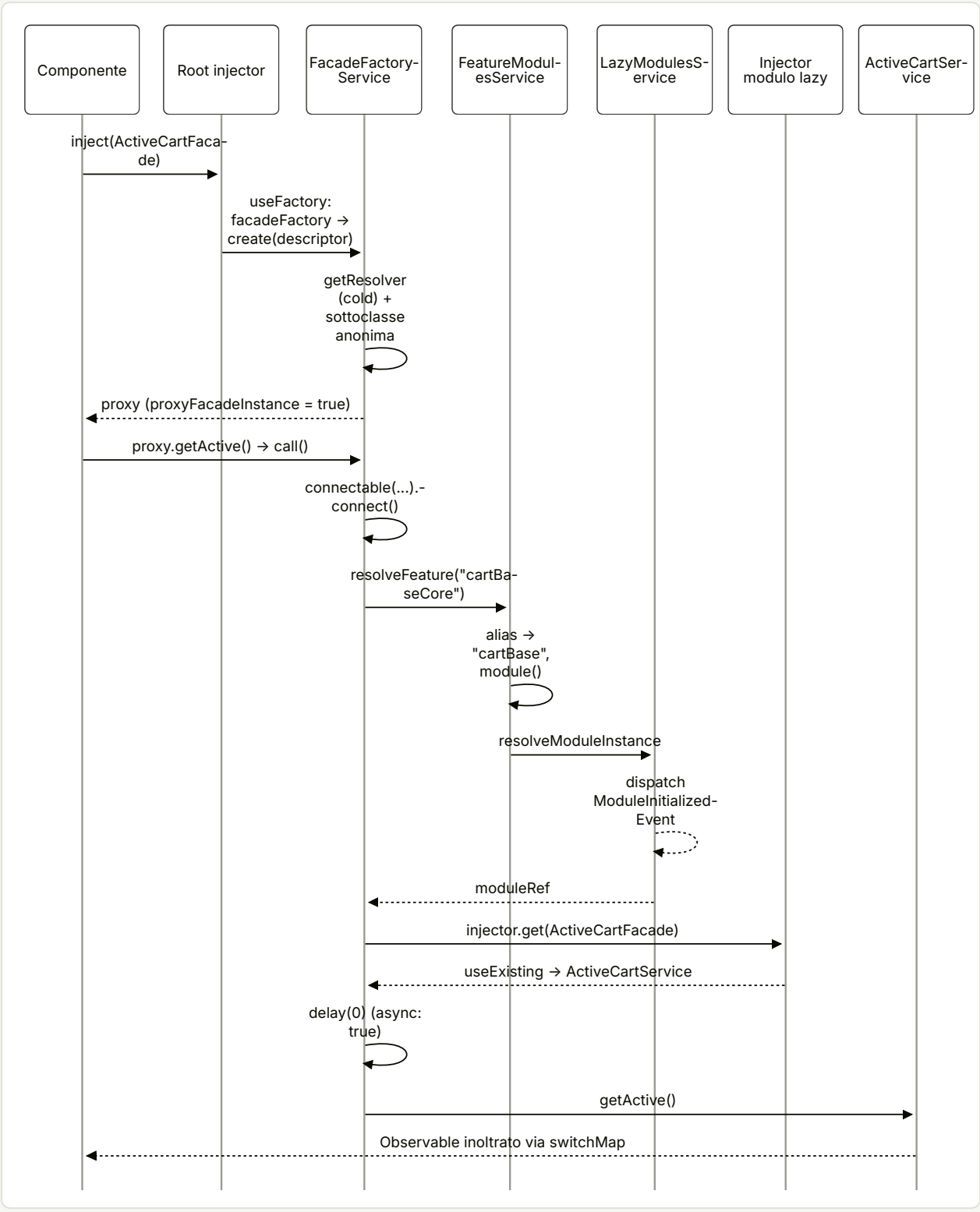


TAVOLA 20.2 **4.3 Facade proxy: prima chiamata di metodo**



21

Livello OCC: adapter, connector, converter, endpoint

Adapter astratti e connector, endpoint configurabili, converter impilabili, scope di caricamento.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

OCC (Omni Commerce Connect) è l'API REST di SAP Commerce Cloud. In Spartacus il "livello OCC" è lo strato che separa la logica di dominio (facade, store NgRx, servizi di caricamento) dal formato concreto delle chiamate HTTP verso il backend. È costruito su quattro idee che si ripetono in tutto il codice, dal core alle feature lib:

1. **Adapter astratto + connector.** Il dominio dipende da una classe astratta (`ProductAdapter`, `CmsPageAdapter`, `SiteAdapter`, …) e da un *connector* `providedIn: 'root'` che la delega; il modulo OCC lega la classe astratta all'implementazione concreta (`OccProductAdapter`, …) via DI. Cambiare backend significa fornire un altro adapter, non toccare facade e store (`core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30`, `core-libs/core/src/product/connectors/product/product.connector.ts:L13-L33`, `core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L76`).
2. **Endpoint configurabili.** Gli URL non sono scritti negli adapter: l'adapter chiede a `OccEndpointsService.buildUrl('product', { urlParams, scope })` e il servizio compone `baseUrl + prefix + baseSite + template` leggendo `backend.occ.endpoints` dalla configurazione globale (`core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150`, `core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L54-L59`). Questo permette di cambiare il parametro `fields` o l'intero path senza estendere classi.
3. **Converter impilabili.** I dati raw (`Occ.*`) passano per `ConverterService`, che applica in catena tutti i converter registrati come multi-provider su un `InjectionToken` (*normalizer* backend→UI, *serializer* UI→backend) (`core-libs/core/src/util/converter.service.ts:L13-L30`, `core-libs/core/src/util/converter.service.ts:L130-L137`).
4. **Scope di caricamento.** Un modello può essere caricato a "pezzi" (scope `list`, `details`, `attributes`, …), ciascuno con un proprio endpoint/`fields`; `LoadingScopesService` espande gli scope inclusi e `OccRequestsOptimizerService` fonde in un'unica richiesta gli scope che puntano allo stesso URL (`core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53`, `core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L34-L80`).

Attorno a questi pilastri ci sono gli interceptor HTTP OCC (credenziali, lingua/valuta, timeout), gli header/context token di "segnalazione" per altri interceptor, la normalizzazione degli errori e il namespace `Occ` dei modelli raw.

Questo capitolo segue il filo di un caso concreto — **il caricamento di un prodotto** — dal facade fino all'URL OCC e ritorno. Il dettaglio di store e facade prodotto è nel capitolo 24 — Prodotto e ricerca; lo state management generico è nel 19; la pipeline di configurazione (`provideDefaultConfig`, validator, factory) nel 11.

2. Mappa dei file

PATH	RUOLO
`core-libs/core/src/occ/config/occ-config.ts`	`BackendConfig`, `OccConfig` (`useExisting: Config`), augmentation di `Config`
`core-libs/core/src/occ/config/default-occ-config.ts`	`defaultOccConfig` (`prefix: '/occ/v2/'`)

PATH	RUOLO
core-libs/core/src/occ/config/occ-config-validator.ts	occConfigValidator (richiede backend.occ.baseUrl)
core-libs/core/src/occ/config/loading-scopes-config.ts	LoadingScopeConfig, LoadingScopesConfig, LoadingScopes
core-libs/core/src/occ/config/config-from-meta-tag-factory.ts	baseUrl OCC/media da <meta>; provideConfigFromMetaTags
core-libs/core/src/occ/config/meta-tag-config.module.ts	MetaTagConfigModule.forRoot()
core-libs/core/src/occ/base-occ.module.ts	BaseOccModule: CMS/site-context OCC, WithCredentialsInterceptor, default + validator
core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts	WithCredentialsInterceptor
core-libs/core/src/occ/occ-models/occ-endpoints.model.ts	DEFAULT_SCOPE, OccEndpoint, ProductOccEndpoint, OccEndpoints (augmentabile)
core-libs/core/src/occ/occ-models/occ.models.ts	namespace Occ con i modelli raw del backend
core-libs/core/src/occ/services/occ-endpoints.service.ts	OccEndpointsService: buildUrl, getBaseUrl, isConfigured, getRawEndpointValue
core-libs/core/src/occ/services/loading-scopes.service.ts	LoadingScopesService: expand, getMaxAge, getReloadTriggers
core-libs/core/src/occ/services/occ-fields.service.ts	OccFieldsService.getOptimalUrlGroups (raggruppa URL che differiscono solo per fields)
core-libs/core/src/occ/services/occ-requests-optimizer.service.ts	OccRequestsOptimizerService.scopedDataLoad
core-libs/core/src/occ/utils/occ-url-util.ts	urlPathJoin
core-libs/core/src/occ/utils/occ-fields.ts	mergeFields, optimizeFields, parseFields, stringifyFields, extractFields
core-libs/core/src/occ/utils/interceptor-util.ts	header USE_CLIENT_TOKEN, USE_CUSTOMER_SUPPORT_AGENT_TOKEN, USE_CAPTCHA_TOKEN; InterceptorUtil
core-libs/core/src/occ/utils/occ-http-token.ts	OccHttpContext, OCC_HTTP_TOKEN
core-libs/core/src/occ/utils/occ-user-ids.ts, occ-constants.ts, index.ts	costanti user id/cart id, OCC_USER_ID_CONSTANTS
core-libs/core/src/occ/adapters/product/product-occ.module.ts	ProductOccModule: binding adapter + normalizer prodotto
core-libs/core/src/occ/adapters/product/default-occ-product-config.ts	endpoint prodotto e loadingScopes.product.details
core-libs/core/src/occ/adapters/product/product-occ-config.ts	ProductScopesConfig + augmentation LoadingScopes.product
core-libs/core/src/occ/adapters/product/occ-product.adapter.ts	OccProductAdapter (load, loadMany)
core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts	ricerca, by codes, suggerimenti, keyword redirect

PATH	RUOLO
core-libs/core/src/occ/adapters/product/converters/*	ProductImageNormalizer, ProductNameNormalizer, normalizer search/references
core-libs/core/src/occ/adapters/cms/*	CmsOccModule, OccCmsPageAdapter, OccCmsComponentAdapter, LegacyOccCmsComponentAdapter, OccCmsPageNormalizer
core-libs/core/src/occ/adapters/site-context/*	SiteContextOccModule, OccSiteAdapter, SiteContextInterceptor, BaseSiteNormalizer
core-libs/core/src/occ/adapters/user/*	UserOccModule e adapter utente (address, consent, payment, ...)
core-libs/core/src/occ/adapters/cost-center/*	CostCenterOccModule, normalizer/serializer cost center
core-libs/core/src/http/http-timeout/*	HttpTimeoutInterceptor, HTTP_TIMEOUT_CONFIG, default timeout
core-libs/core/src/util/converter.service.ts	Converter, ConverterService
core-libs/core/src/util/try-normalize-http-error.ts	tryNormalizeHttpError
core-libs/core/src/util/occ-http-error-constants.ts	OccHttpErrorType, OccHttpErrorReason
core-libs/core/src/product/connectors/product/*	ProductAdapter, ProductConnector, PRODUCT_NORMALIZER, ScopedProductData

3. Come funziona passo per passo

3.1 Configurazione: `OccConfig` e `BackendConfig`

`BackendConfig` raggruppa tutto ciò che riguarda il backend: `occ` (`baseUrl`, `prefix`, `useWithCredentials`, `endpoints`), `media` (`baseUrl`, `prefix`) e `loadingScopes` (core-libs/core/src/occ/config/occ-config.ts:L13-L40). `OccConfig` è una classe astratta `providedIn: 'root'` con `useExisting: Config` che estende `SiteContextConfig` e aggiunge `backend?: BackendConfig`: iniettare `OccConfig` restituisce quindi l'oggetto di configurazione globale già fuso (core-libs/core/src/occ/config/occ-config.ts:L42-L48). Lo stesso file aumenta `Config` (module augmentation) così che `Config` estenda `OccConfig` (core-libs/core/src/occ/config/occ-config.ts:L50-L52); `Config` in sé è una classe astratta vuota (core-libs/core/src/config/config-tokens.ts:L21-L21).

Il default OCC imposta solo `prefix: '/occ/v2/'`, nessun `baseUrl` (core-libs/core/src/occ/config/default-occ-config.ts:L9-L15); per questo `occConfigValidator` segnala un errore se `backend.occ.baseUrl` è `undefined` (core-libs/core/src/occ/config/occ-config-validator.ts:L9-L17).

Altri moduli aumentano `BackendConfig`: per esempio il timeout HTTP aggiunge `timeout?: HttpTimeoutConfig` tramite `BackendHttpTimeoutConfig` (core-libs/core/src/http/http-timeout/http-timeout.config.ts:L35-L41). `BackendConfig` e `OccEndpoints` sono esportati esplicitamente nella sezione *AUGMENTABLE_TYPES* del public API di core (core-libs/core/public_api.ts:L50-L51).

3.2 Bootstrap del livello OCC

`BaseCoreModule` importa, in quest'ordine, `SiteContextModule.forRoot()`, `MetaTagConfigModule.forRoot()`, `BaseOccModule.forRoot()`, `LazyLoadingModule.forRoot()`, `HttpModule.forRoot()` (core-libs/core/src/base-

core.module.ts:L38-L42).

- BaseOccModule importa CmsOccModule e SiteContextOccModule; forRoot() registra WithCredentialsInterceptor in HTTP_INTERCEPTORS (useExisting, multi), defaultOccConfig e il validator (core-libs/core/src/occ/base-occ.module.ts:L17-L35). HTTP_INTERCEPTORS è il multi-token Angular InjectionToken<readonly HttpInterceptor[]> (angular:packages/common/http/src/interceptor.ts:L195-L197).
- SiteContextOccModule registra SiteContextInterceptor, OccSiteAdapter per SiteAdapter, BaseSiteNormalizer su BASE_SITE_NORMALIZER e gli endpoint di default site-context (core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts:L18-L38).
- CmsOccModule lega CmsPageAdapter→OccCmsPageAdapter e CmsComponentAdapter→OccCmsComponentAdapter con useExisting e registra OccCmsPageNormalizer su CMS_PAGE_NORMALIZER (core-libs/core/src/occ/adapters/cms/cms-occ.module.ts:L16-L34).
- HttpModule importa HttpTimeoutModule, che registra HttpTimeoutInterceptor e il default di timeout; HttpModule.forRoot() non aggiunge provider (core-libs/core/src/http/http-timeout/http-timeout.module.ts:L13-L23, core-libs/core/src/http/http.module.ts:L10-L19).

Attenzione: ProductOccModule *non* è importato da BaseCoreModule. L'app demo lo importa esplicitamente e lo schematic add-spartacus lo aggiunge al modulo delle feature (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L15-L15, core-libs/schematics/src/add-spartacus/spartacus-features.ts:L187-L187).

3.3 baseUrl dai meta tag

MetaTagConfigModule.forRoot() espone provideConfigFromMetaTags(), che registra due config factory dipendenti da Meta (core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L37-L42, core-libs/core/src/occ/config/meta-tag-config.module.ts:L10-L18). La lettura usa meta.getTag('name="..."') (core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L32-L35). I meta tag sono occ-backend-base-url e media-backend-base-url, con placeholder OCC_BACKEND_BASE_URL_VALUE / MEDIA_BACKEND_BASE_URL_VALUE (core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L12-L16). Se il tag esiste e non contiene il placeholder, la factory contribuisce { backend: { occ: { baseUrl } } } (o backend.media.baseUrl), altrimenti {} (core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L18-L30). In pratica il baseUrl può essere iniettato nell'index.html al deploy, senza ricompilare.

3.4 Il modello degli endpoint e gli scope

DEFAULT_SCOPE vale 'default' (core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L7-L7). Un OccEndpoint è una mappa *scope → template* con chiave opzionale default; ProductOccEndpoint aggiunge list, details, attributes, variants (core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L9-L19). In OccEndpoints ogni voce è string | OccEndpoint (es. product?: string | ProductOccEndpoint), salvo eccezioni come gli endpoint del configuratore, tipizzati solo string (core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L33, core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L162-L167).

Le feature lib estendono OccEndpoints via module augmentation di @spartacus/core, per esempio il carrello (feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L97-L98).

I template di default per il prodotto stanno in defaultOccProductConfig: l'endpoint product ha scope default, list, details, promotions, attributes, price, stock, unit, list_item, ciascuno con il proprio fields (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28). Lo stesso file definisce productAvailabilities, productReviews, productReferences, productSearch (scope default, carousel, carouselMinimal), productSearchByCategory (default, code), productSuggestions (core-libs/core/src/occ/

adapters/product/default-occ-product-config.ts:L29-L49). Lo scope `variants` dell'endpoint `product` viene invece dalla feature lib `product/variants` (feature-libs/product/variants/occ/config/default-occ-product-variants-config.ts:L9-L16). Gli endpoint CMS (`component`, `components`, `pages`, `page`) sono definiti in `defaultCmsModuleConfig`, non nel modulo OCC (core-libs/core/src/cms/config/default-cms-config.ts:L9-L17).

3.5 `OccEndpointsService`: da chiave a URL

`OccEndpointsService` è `providedIn: 'root'`, inietta `OccConfig` e, opzionalmente, `BaseSiteService`; se presente, si sottoscrive a `getActive()` e memorizza `_activeBaseSite` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L31-L55). Finché il valore non arriva, il getter `activeBaseSite` ricade sul primo valore configurato del contesto `baseSite` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L37-L42, core-libs/core/src/site-context/config/context-config-utils.ts:L28-L34).

`buildUrl(endpoint, attributes?, propertiesToOmit?)` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150) procede così:

1. **Risoluzione del template per scope** con `getEndpointForScope` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L196-L225):
   - se `backend.occ.endpoints` manca → stringa vuota (core-libs/core/src/occ/services/occ-endpoints.service.ts:L197-L201);
   - se lo scope richiesto è configurato → quel template;
   - se lo scope è `default` e l'endpoint è una stringa → la stringa;
   - altrimenti, in dev mode, `logger.warn("<endpoint> endpoint configuration missing for scope ...")`, e poi fallback: la stringa, oppure lo scope `default`, oppure **il nome stesso dell'endpoint** (core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L224).
2. **Path params**: se ci sono `urlParams`, `StringTemplate.resolve(url, urlParams, true)` sostituisce ogni `${label}` con il valore codificato da `encodeURIComponent` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L121-L123, core-libs/core/src/config/utils/string-template.ts:L15-L31).
3. **Query params**: se ci sono `queryParams`, la query già presente nel template viene separata (`split('?')`) e passata come `fromString` a un `HttpParams` con encoder `HttpParamsURIEncoder`; poi `getHttpParamsFromQueryParams` applica ogni chiave: `undefined` è ignorato, `null` **cancella** il parametro (anche se proveniva dal template), gli altri valori fanno `set` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L146, core-libs/core/src/occ/services/occ-endpoints.service.ts:L152-L168). `HttpParams` con `fromString` analizza la stringa con il codec fornito (angular:packages/common/http/src/params.ts:L165-L180); `HttpParamsURIEncoder` usa `encodeURIComponent` puro per chiavi e valori, al contrario del codec di default di Angular che de-codifica alcuni caratteri (core-libs/core/src/util/http-params-uri.encoder.ts:L7-L23).
4. **Prefisso base**: `buildUrlFromEndpointString` fa `urlPathJoin(this.getBaseUrl(propertiesToOmit), endpointString)` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L233-L238).

`getBaseUrl()` unisce con `urlPathJoin` `backend.occ.baseUrl` (default `''`), il prefix e il baseSite, omettendo ogni parte il cui flag in `BaseOccUrlProperties` vale esplicitamente `false` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102, core-libs/core/src/occ/services/occ-endpoints.service.ts:L19-L29). `getPrefix()` antepone `/` se manca (core-libs/core/src/occ/services/occ-endpoints.service.ts:L240-L248).

`urlPathJoin` scarta le parti falsy, toglie gli slash iniziali/finali da ogni parte, unisce con `/` e preserva lo slash iniziale della prima parte e quello finale dell'ultima (core-libs/core/src/occ/utils/occ-url-util.ts:L12-L26). Il test documenta che `urlPathJoin('https://test123', '/occ/v2/', 'test3')` dà `'https://test123/occ/v2/test3'` (core-libs/core/src/occ/utils/occ-url-util.spec.ts:L20-L24).

Due metodi ausiliari:

- `isConfigured(endpoint, scope)` usa `getEndpointFromConfig`, che **non** fa fallback: uno scope non configurato dà `undefined` → `false` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L75-L79, core-libs/core/src/occ/services/occ-endpoints.service.ts:L170-L193).
- `getRawEndpointValue(endpoint, scope)` restituisce il template non risolto, con lo stesso fallback di `getEndpointForScope` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L63-L67).

Esempio (calcolo a mano). Con `baseUrl = 'https://api.example.com'`, prefix di default `'/occ/v2/'`, baseSite attivo `electronics-spa`, `buildUrl('product', { urlParams: { productCode: '358639' }, scope: 'list' })` prende il template `list` (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L17-L17), sostituisce `${productCode}` e restituisce `https://api.example.com/occ/v2/electronics-spa/products/358639?fields=code,purchasable,name,summary,price(formattedValue),images(DEFAULT,galleryIndex),baseProduct`. La query del template resta letterale perché non ci sono `queryParams` (vedi trappola 9.8). I valori `api.example.com`, `electronics-spa`, `358639` sono esemplificativi.

3.6 Adapter e connector

`ProductAdapter` è una classe astratta con `load(productCode, scope?)` obbligatorio e `loadMany?(products)` opzionale (core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30). `ProductConnector` (`providedIn: 'root'`) inietta l'adapter: `get` ha scope di default `''`; `getMany` usa `loadMany` se l'adapter lo implementa, altrimenti chiama `load` per ogni elemento e mette l'observable in `data$` (core-libs/core/src/product/connectors/product/product.connector.ts:L13-L33). `ScopedData<T>` ha `scope?` e `data$?: Observable<T>`; `ScopedProductData` aggiunge `code` (core-libs/core/src/model/scoped-data.ts:L12-L15, core-libs/core/src/product/connectors/product/scoped-product-data.ts:L13-L15).

`ProductOccModule` collega le classi astratte alle implementazioni OCC (`useClass`): `ProductAdapter`→`OccProductAdapter`, `ProductAvailabilityAdapter`→`OccProductAvailabilityAdapter`, e così references, search e reviews; registra `ProductImageNormalizer` e poi `ProductNameNormalizer` su `PRODUCT_NORMALIZER` (`multi`, in quest'ordine) (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L76). Il modulo importa `./product-occ-config` per side effect, il che attiva l'augmentation `LoadingScopes.product` (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L26-L26, core-libs/core/src/occ/adapters/product/product-occ-config.ts:L13-L30).

`OccProductAdapter.load` fa `http.get(getEndpoint(code, scope))` e applica `converter.pipeable(PRODUCT_NORMALIZER)` (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L29-L33). `loadMany` costruisce gli URL per ciascun `(code, scope)`, li passa a `OccRequestsOptimizerService.scopedDataLoad` e aggiunge il normalizer a ogni `data$` (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L35-L52). `getEndpoint` è `buildUrl('product', { urlParams: { productCode }, scope })` (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L54-L59).

Lo stesso schema vale per gli altri domini: CMS (core-libs/core/src/occ/adapters/cms/cms-occ.module.ts:L16-L34), site context (core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts:L18-L38), utente (`UserOccModule` registra 8 adapter e 3 normalizer, core-libs/core/src/occ/adapters/user/user-occ.module.ts:L34-L72), cost center (core-libs/core/src/occ/adapters/cost-center/cost-center-occ.module.ts:L20-L40).

3.7 `ConverterService`: normalizer e serializer

L'interfaccia `Converter<SOURCE, TARGET>` ha un solo metodo `convert(source, target?)`; il commento stabilisce la convenzione *normalize* = backend→UI, *serialize* = UI→backend, e che i converter si possono impilare

(core-libs/core/src/util/converter.service.ts:L13-L30).

`ConverterService`:

- mantiene una cache `Map<InjectionToken, Converter[]>` che viene svuotata ogni volta che `UnifiedInjector.injectors$` emette, cioè quando appare un nuovo injector (tipicamente un modulo lazy) (core-libs/core/src/util/converter.service.ts:L35-L50);
- legge i converter di un token in modo sincrono con `getLastValueSync(this.unifiedInjector.getMulti(token))` (core-libs/core/src/util/converter.service.ts:L52-L65). `UnifiedInjector.getMulti` accumula con `scan` le istanze multi-provider dell'injector root e dei moduli lazy, e lancia un errore se trova un provider non-multi (core-libs/core/src/lazy-loading/unified-injector.ts:L75-L93);
- `pipeable(token)` restituisce `map(convertSource)` se ci sono converter, altrimenti un operatore identità; `pipeableMany` fa lo stesso sugli array (core-libs/core/src/util/converter.service.ts:L80-L101);
- `convert`/`convertMany` restituiscono la sorgente (cast) se il token non ha converter (core-libs/core/src/util/converter.service.ts:L106-L128);
- `convertSource` applica i converter in catena con `reduce` partendo da `target = undefined`: il primo converter crea il target, i successivi lo arricchiscono (core-libs/core/src/util/converter.service.ts:L130-L137).

`PRODUCT_NORMALIZER` è un `InjectionToken<Converter<any, Product>>` (core-libs/core/src/product/connectors/product/converters.ts:L11-L13).

I due normalizer di prodotto mostrano bene il contratto:

- `ProductImageNormalizer` crea il target come copia della sorgente se assente e sostituisce `images` con la struttura normalizzata (core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L18-L26). Le immagini sono raggruppate per `imageType` e `format`; quelle con `galleryIndex` finiscono in array (`images.GALLERY[i]`), le altre in oggetto (`images.PRIMARY`) (core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L36-L61). Gli URL relativi sono prefissati con `backend.media.baseUrl`, altrimenti `backend.occ.baseUrl`; quelli che iniziano con `http`, `data:image` o `//` restano invariati (core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L91-L100).
- `ProductNameNormalizer` rimuove i tag HTML da `name`, genera `slug` e conserva l'originale in `nameHtml` (core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts:L17-L33); lo slug sostituisce i caratteri riservati `!*'();:@&=+$,/?%#[]` con `-` e comprime i trattini multipli (core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts:L42-L64).

I normalizer possono annidarsi: `OccProductSearchPageNormalizer` converte ogni prodotto della pagina con `converterService.convert(product, PRODUCT_NORMALIZER)` (core-libs/core/src/occ/adapters/product/converters/occ-product-search-page-normalizer.ts:L31-L47), e `OccProductReferencesListNormalizer` fa lo stesso su `references[].target` (core-libs/core/src/occ/adapters/product/converters/occ-product-references-list-normalizer.ts:L22-L39). Il serializer è il percorso inverso: `OccProductReviewsAdapter.post` passa la review per `PRODUCT_REVIEW_SERIALIZER` e la invia come `application/x-www-form-urlencoded` (core-libs/core/src/occ/adapters/product/occ-product-reviews.adapter.ts:L39-L55); `OccCostCenterSerializer` rinomina `active` in `activeFlag` (core-libs/core/src/occ/adapters/cost-center/converters/occ-cost-center-serializer.ts:L18-L25).

3.8 Loading scopes

`LoadingScopeConfig` ha `include?: string[]`, `maxAge?: number` (secondi) e `reloadOn?: Type<CxEvent>[]`; `LoadingScopes` è una mappa *modello → scope → config* (core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L31). Il default dichiara `loadingScopes.product.details.include = [LIST, VARIANTS]` (core-

libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58). L'enum `ProductScope` elenca `list`, `details`, `attributes`, `variants`, `code`, `price`, `stock`, `unit`, `promotions`, `list_item`, `multi_dimensional`, `multi_dimensional_availability` (core-libs/core/src/product/model/product-scope.ts:L7-L21).

`LoadingScopesService.expand(model, scopes)` scorre gli scope da destra e inserisce gli scope inclusi **prima** dello scope che li include, senza duplicati, ricorsivamente sugli scope appena inseriti (core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53). Lo spec lo conferma: con `list.include = ['base']`, `expand('product', ['list'])` dà `['base', 'list']`, e con `detail.include = ['list']` si ottiene `['base', 'list', 'detail']` (core-libs/core/src/occ/services/loading-scopes.service.spec.ts:L52-L60). Con la configurazione di default, quindi, `['details']` diventa `['list', 'variants', 'details']`. Il JSDoc stabilisce che i dati degli scope si fondono nell'ordine dato, con l'ultimo che sovrascrive (core-libs/core/src/occ/services/loading-scopes.service.ts:L17-L28).

`getMaxAge` converte `maxAge` da secondi a millisecondi (default 0) e `getReloadTriggers` restituisce `reloadOn` o `[]` (core-libs/core/src/occ/services/loading-scopes.service.ts:L61-L77).

3.9 Ottimizzazione delle richieste: fusione dei `fields`

`OccRequestsOptimizerService.scopedDataLoad(scopedDataWithUrls, dataFactory?)` usa di default `http.get(url)` come data factory e raggruppa i modelli con `OccFieldsService.getOptimalUrlGroups` (core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L34-L44).

`getOptimalUrlGroups` (core-libs/core/src/occ/services/occ-fields.service.ts:L59-L82):

1. per ogni modello separa l'URL "senza `fields`" (con gli altri parametri ordinati alfabeticamente) dal valore di `fields`, tramite `splitFields` (core-libs/core/src/occ/services/occ-fields.service.ts:L89-L116; il nome del parametro è `FIELDS_PARAM = 'fields'`, core-libs/core/src/occ/services/occ-fields.service.ts:L48-L48);
2. analizza `fields` in un AST con `parseFields` (`a,b(c,d)` → `{a:{}, b:{c:{}, d:{}}}`) (core-libs/core/src/occ/utils/occ-fields.ts:L49-L84, core-libs/core/src/occ/utils/occ-fields.spec.ts:L76-L85);
3. indicizza ciascun gruppo per scope e ricostruisce l'URL con i fields fusi.

`mergeFields` fonde gli AST con `deepMerge`, li ottimizza e li riconverte in stringa (core-libs/core/src/occ/utils/occ-fields.ts:L14-L20); `optimizeFields` rimuove `DEFAULT` e `BASIC` se c'è `FULL`, e `BASIC` se c'è `DEFAULT`, ricorsivamente (core-libs/core/src/occ/utils/occ-fields.ts:L27-L41).

Infine, se un URL ottimale copre un solo scope il suo `data$` viene passato così com'è; se ne copre più di uno viene fatta un'unica richiesta condivisa con `shareReplay(1)` e ogni scope ne estrae la propria parte con `extractFields` (core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L55-L74). `extractFields` restituisce l'oggetto intero se i fields sono vuoti o contengono `BASIC`/`DEFAULT`/`FULL` (definizione ambigua), altrimenti copia solo le chiavi richieste (core-libs/core/src/occ/utils/occ-fields.ts:L106-L139). `OccFieldsService` è usato anche fuori dal core, per esempio negli adapter order-history e quote (feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L48-L48, feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L48-L48).

3.10 Interceptor OCC

- `WithCredentialsInterceptor` clona la request con `withCredentials: true` quando `requiresWithCredentials` è vero (core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L28-L38). La condizione è `useWithCredentials !== undefined` *e* l'URL contiene il `prefix` configurato, con match per sottostringa (core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L43-L52). Il test conferma che anche un URL di terze parti che contiene il prefix riceve

withCredentials (core-libs/core/src/occ/interceptors/with-credentials.interceptor.spec.ts:L151-L160).

- SiteContextInterceptor (providedIn: 'root') inizializza lingua e valuta dai default della configurazione e poi si sottoscrive a LanguageService.getActive() e CurrencyService.getActive() (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L25-L52). In intercept aggiunge i query param lang e curr solo se l'URL contiene occEndpoints.getBaseUrl() (baseUrl + prefix + baseSite) (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68).
- HttpTimeoutInterceptor sceglie il timeout dando priorità al context token HTTP_TIMEOUT_CONFIG rispetto a config.backend.timeout, e usa browser o server in base a windowRef.isBrowser() (core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L77-L84). Se il valore è undefined la richiesta passa senza timeout (core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L44-L47). Il timer parte solo all'evento HttpEventType.Sent: switchMap su NEVER.pipe(startWith(event), timeout(timeoutValue)), abbandonato all'evento successivo (core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L57). Un TimeoutError RxJS diventa un HttpErrorResponse il cui error è un Error con il messaggio, loggato anche con logger.warn (core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L92-L124). Default: 20 000 ms lato server, nessun timeout in browser (core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15).

L'ordine effettivo della catena HTTP_INTERCEPTORS (questi tre più quelli di auth, ASM, error handling) dipende dall'ordine di registrazione dei provider nei moduli importati. DA VERIFICARE: non ricostruito end-to-end in questo capitolo; vedi 06 — HttpClient e interceptor.

3.11 Header di segnalazione e HttpContext

Gli adapter a volte devono chiedere un comportamento a un interceptor che sta "più avanti". Spartacus usa due meccanismi.

Header di segnalazione. Le costanti sono cx-use-client-token, cx-use-csagent-token e sap-commerce-cloud-captcha-token (core-libs/core/src/occ/utils/interceptor-util.ts:L9-L11). InterceptorUtil.createHeader serializza il parametro con JSON.stringify (in append se gli header esistono), getInterceptorParam lo rilegge con JSON.parse e removeHeader clona la request senza l'header (core-libs/core/src/occ/utils/interceptor-util.ts:L13-L47). Esempio lato produttore: OccUserAddressAdapter.verify aggiunge cx-use-client-token quando l'utente è anonymous (core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L96-L106). Esempio lato consumatore: ClientTokenInterceptor riconosce la richiesta con getInterceptorParam e rimuove l'header prima di inoltrarla (core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L42-L49, core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L99-L105) — dettagli nel capitolo 22 — Autenticazione. USE_CAPTCHA_TOKEN è invece usato come header reale, con il token del provider captcha come valore (feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L199-L199).

HttpContext. OCC_HTTP_TOKEN è un HttpContextToken<OccHttpContext> con default {}; il campo sendUserIdAsHeader (string | boolean) chiede di inviare l'id del cliente emulato (ASM) (core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20). In Angular HttpContextToken contiene solo una factory defaultValue (angular:packages/common/http/src/context.ts:L16-L18). OccProductSearchAdapter.search e OccUserCostCenterAdapter.loadActiveList lo impostano a true (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L46-L71, core-libs/core/src/occ/adapters/user/occ-user-cost-centers.adapter.ts:L31-L55). Il consumatore è UserIdHttpHeaderInterceptor di ASM: se il valore è una stringa la usa come user id, se è true prende lo user id da UserIdService, il tutto solo con asm.userIdHttpHeader.enable (feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L49-L69); vedi 29 — ASM.

Costanti correlate: OCC_USER_ID_CURRENT = 'current', OCC_USER_ID_ANONYMOUS = 'anonymous', OCC_USER_ID_GUEST = 'guest', OCC_CART_ID_CURRENT = 'current' (core-libs/core/src/occ/utils/occ-user-

ids.ts:L7-L9, core-libs/core/src/occ/utils/occ-constants.ts:L7-L7); OCC_USER_ID_CONSTANTS è un InjectionToken root la cui factory restituisce il namespace OCC_USER_IDS (core-libs/core/src/occ/utils/index.ts:L9-L24).

3.12 Errori

tryNormalizeHttpError converte un HttpErrorResponse in HttpErrorModel (message, status, statusText, url), copiando error.error.errors in details o, per gli errori OAuth, error/error_description (core-libs/core/src/util/try-normalize-http-error.ts:L21-L57). Se l'errore non è né HttpErrorResponse né HttpErrorModel lo restituisce invariato e in dev mode logga un errore (core-libs/core/src/util/try-normalize-http-error.ts:L59-L65). Gli adapter lo usano nel proprio catchError (es. core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L55-L58), oppure lo applica l'effetto NgRx (prodotto, §3.13). Il formato raw è Occ.ErrorModel (message, reason, subject, subjectType, type) dentro Occ.ErrorList (core-libs/core/src/occ/occ-models/occ.models.ts:L104-L137); OccHttpErrorType (NotFoundError, ClassMismatchError) e OccHttpErrorReason (notFound) sono costanti pubbliche per riconoscerli (core-libs/core/src/util/occ-http-error-constants.ts:L9-L16, core-libs/core/public_api.ts:L54-L57). La gestione globale degli errori è nel capitolo 41.

3.13 Esempio end-to-end: caricamento di un prodotto

1. Un componente chiama ProductService.get(code, scopes); lo scope di default è DEFAULT_SCOPE e il facade delega a ProductLoadingService.get passando un array (core-libs/core/src/product/facade/product.service.ts:L39-L46).
2. ProductLoadingService.get espande gli scope con loadingScopes.expand('product', scopes) e crea o riusa un observable per scope (core-libs/core/src/product/services/product-loading.service.ts:L49-L56). Con ['details'] si ottiene ['list', 'variants', 'details'] (§3.8).
3. Per ciascuno scope, se lo stato non è loading/success/error (o scatta un reload trigger) e non è già in caricamento, viene dispatchata LoadProduct(productCode, scope) (core-libs/core/src/product/services/product-loading.service.ts:L102-L133).
4. ProductEffects.loadProduct$ raggruppa le LOAD_PRODUCT contemporanee con bufferDebounceTime e le passa tutte a productConnector.getMany; le richieste sono annullate (withdrawOn) al cambio di lingua/valuta (core-libs/core/src/product/store/effects/product.effect.ts:L34-L57).
5. ProductConnector.getMany chiama OccProductAdapter.loadMany (core-libs/core/src/product/connectors/product/product.connector.ts:L23-L32).
6. loadMany costruisce un URL per scope via buildUrl('product', …) (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L35-L59); buildUrl risolve template, parametri e base (core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150).
7. scopedDataLoad raggruppa gli URL che differiscono solo per fields: i tre scope dell'esempio hanno tutti products/${productCode}?fields=…, quindi finiscono nello stesso gruppo e producono **una sola** http.get con fields fusi, condivisa con shareReplay(1) (core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L55-L74, core-libs/core/src/occ/services/occ-fields.service.ts:L59-L82). Se però lo scope variants non è configurato (feature product/variants assente), il suo URL ricade sul template default (§3.5) e il risultato cambia: vedi trappola 9.5.
8. La richiesta attraversa gli interceptor: SiteContextInterceptor aggiunge lang/curr, WithCredentialsInterceptor eventualmente withCredentials, HttpTimeoutInterceptor il timeout (§3.10).
9. Ogni scope estrae la propria parte con extractFields (core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L64-L72) e passa per converter.pipeable(PRODUCT_NORMALIZER): prima ProductImageNormalizer, poi ProductNameNormalizer (core-libs/core/src/occ/adapters/product/occ-

product.adapter.ts:L43-L51, core-libs/core/src/util/converter.service.ts:L130-L137, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51).

10. L'effetto emette `LoadProductSuccess({ code, ...data }, scope)` oppure, in errore, `LoadProductFail(code, tryNormalizeHttpError(error, logger), scope)` (core-libs/core/src/product/store/effects/product.effect.ts:L59-L91); se `data$` manca emette `LoadProductFail` con `Error('Scoped product data does not exist')` (core-libs/core/src/product/store/effects/product.effect.ts:L82-L89).
11. `ProductLoadingService` combina gli scope con `uniteLatest` e `deepMerge({}, ...productParts)`: finché uno degli scope non ha dati emette `undefined`, poi il prodotto fuso (con `distinctUntilChanged`) (core-libs/core/src/product/services/product-loading.service.ts:L74-L85).

---

4. Diagrammi

4.1 Product load: dal facade alla risposta normalizzata

■ **TAVOLA 21.1** diagramma a tutta pagina, in fondo al capitolo

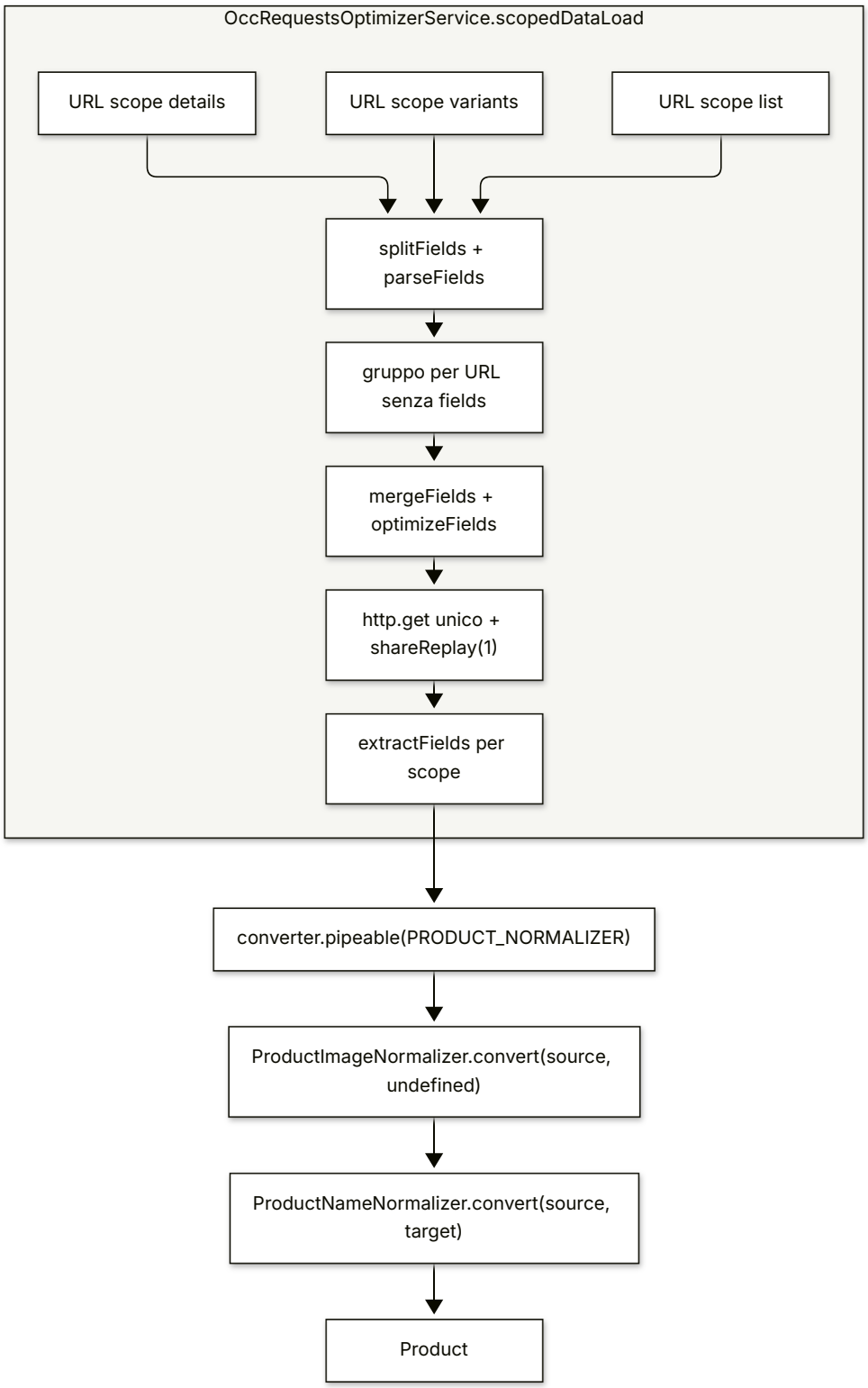
Riferimenti: passi 1-11 di §3.13.

4.2 Costruzione dell'URL in `buildUrl`

■ **TAVOLA 21.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L168, core-libs/core/src/occ/services/occ-endpoints.service.ts:L196-L238.

4.3 Catena dei converter e fusione dei fields



Riferimenti: core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L34-L80, core-libs/core/src/occ/services/occ-fields.service.ts:L59-L116, core-libs/core/src/occ/utils/occ-fields.ts:L14-L41, core-libs/core/src/util/converter.service.ts:L130-L137, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51.

5. Estratti di codice reale

5.1 `OccConfig` e `BackendConfig`

core-libs/core/src/occ/config/occ-config.ts:L13-L52

```ts
export interface BackendConfig {
  occ?: {
    baseUrl?: string;
    prefix?: string;
    /**
     * Indicates whether or not cross-site Access-Control requests should be made
     * using credentials such as cookies, authorization headers or TLS client certificates
     */
    useWithCredentials?: boolean;

    endpoints?: OccEndpoints;
  };
  media?: {
    // ...
    baseUrl?: string;
    // ...
    prefix?: string;
  };
  loadingScopes?: LoadingScopes;
}

@Injectable({
  providedIn: 'root',
  useExisting: Config,
})
export abstract class OccConfig extends SiteContextConfig {
  backend?: BackendConfig;
}

declare module '../../config/config-tokens' {
  interface Config extends OccConfig {}
}
```

5.2 `buildUrl` e la scelta del template

core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150

```ts
  buildUrl(
    endpoint: string,
    attributes?: DynamicAttributes,
    propertiesToOmit?: BaseOccUrlProperties
```

```
  ): string {
    let url = this.getEndpointForScope(endpoint, attributes?.scope);

    if (attributes) {
      const { urlParams, queryParams } = attributes;

      if (urlParams) {
        url = StringTemplate.resolve(url, urlParams, true);
      }

      if (queryParams) {
        let httpParamsOptions = { encoder: new HttpParamsURIEncoder() };

        if (url.includes('?')) {
          let queryParamsFromEndpoint: string;
          [url, queryParamsFromEndpoint] = url.split('?');
          httpParamsOptions = {
            ...httpParamsOptions,
            ...{ fromString: queryParamsFromEndpoint },
          };
        }
        // ...
      }
    }

    return this.buildUrlFromEndpointString(url, propertiesToOmit);
  }
```

core-libs/core/src/occ/services/occ-endpoints.service.ts:L195-L225

```ts
  // TODO: Can we reuse getEndpointFromConfig in this method? Should we change behavior of this
function?
  private getEndpointForScope(endpoint: string, scope?: string): string {
    const endpointsConfig = this.config.backend?.occ?.endpoints;

    if (!endpointsConfig) {
      return '';
    }

    const endpointConfig: any =
      endpointsConfig[endpoint as keyof typeof endpointsConfig];

    if (scope) {
      if (endpointConfig?.[scope]) {
        return endpointConfig?.[scope];
      }
      if (scope === DEFAULT_SCOPE && typeof endpointConfig === 'string') {
        return endpointConfig;
```

```
      }
      if (isDevMode()) {
        this.logger.warn(
          `${endpoint} endpoint configuration missing for scope "${scope}"`
        );
      }
    }

    return (
      (typeof endpointConfig === 'string'
        ? endpointConfig
        : endpointConfig?.[DEFAULT_SCOPE]) || endpoint
    );
  }
```

5.3 `urlPathJoin`

core-libs/core/src/occ/utils/occ-url-util.ts:L12-L33

```ts
export function urlPathJoin(...parts: string[]): string {
  const paths: string[] = [];
  parts = parts.filter((part) => Boolean(part));
  for (const part of parts) {
    paths.push(cleanSlashes(part));
  }

  if (parts[0]?.startsWith('/')) {
    paths[0] = '/' + paths[0];
  }
  if (parts[parts.length - 1]?.endsWith('/')) {
    paths[paths.length - 1] = paths[paths.length - 1] + '/';
  }
  return paths.join('/');
}

function cleanSlashes(path: string): string {
  path = path.startsWith('/') ? path.slice(1) : path;
  path = path.endsWith('/') ? path.slice(0, -1) : path;

  return path;
}
```

5.4 Adapter OCC del prodotto e connector

core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L29-L59

```ts
  load(productCode: string, scope?: string): Observable<Product> {
    return this.http
      .get(this.getEndpoint(productCode, scope))
      .pipe(this.converter.pipeable(PRODUCT_NORMALIZER));
  }

  loadMany(products: ScopedProductData[]): ScopedProductData[] {
    const scopedDataWithUrls: ScopedDataWithUrl[] = products.map((model) => ({
      scopedData: model,
      url: this.getEndpoint(model.code, model.scope),
    }));

    return this.requestsOptimizer
      .scopedDataLoad<Occ.Product>(scopedDataWithUrls)
      .map(
        (scopedProduct) =>
          ({
            ...scopedProduct,
            data$: scopedProduct.data$?.pipe(
              this.converter.pipeable(PRODUCT_NORMALIZER)
            ),
          }) as ScopedProductData
      );
  }

  protected getEndpoint(code: string, scope?: string): string {
    return this.occEndpoints.buildUrl('product', {
      urlParams: { productCode: code },
      scope,
    });
  }
```

core-libs/core/src/product/connectors/product/product.connector.ts:L13-L33

```ts
@Injectable({
  providedIn: 'root',
})
export class ProductConnector {
  constructor(protected adapter: ProductAdapter) {}

  get(productCode: string, scope = ''): Observable<Product> {
    return this.adapter.load(productCode, scope);
  }

  getMany(products: ScopedProductData[]): ScopedProductData[] {
    if (!this.adapter.loadMany) {
      return products.map((product) => ({
        ...product,
```

```
        data$: this.adapter.load(product.code, product.scope),
      }));
    }

    return this.adapter.loadMany(products);
  }
}
```

5.5 Binding DI in `ProductOccModule`

core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L51

```ts
@NgModule({
  imports: [CommonModule],
  providers: [
    provideDefaultConfig(defaultOccProductConfig),
    {
      provide: ProductAdapter,
      useClass: OccProductAdapter,
    },
    {
      provide: ProductAvailabilityAdapter,
      useClass: OccProductAvailabilityAdapter,
    },
    {
      provide: PRODUCT_NORMALIZER,
      useExisting: ProductImageNormalizer,
      multi: true,
    },
    {
      provide: PRODUCT_NORMALIZER,
      useExisting: ProductNameNormalizer,
      multi: true,
    },
    // ...
```

5.6 Il cuore di `ConverterService`

core-libs/core/src/util/converter.service.ts:L80-L88

```ts
  pipeable<S, T>(
    injectionToken: InjectionToken<Converter<S, T>>
  ): OperatorFunction<S, T> {
    if (this.hasConverters(injectionToken)) {
      return map((model: S) => this.convertSource(model, injectionToken));
    } else {
```

```
      return (observable: Observable<any>) => observable as Observable<T>;
    }
  }
```

core-libs/core/src/util/converter.service.ts:L130-L137

```ts
  private convertSource<S, T>(
    source: S,
    injectionToken: InjectionToken<Converter<S, T>>
  ): T {
    return this.getConverters(injectionToken)?.reduce((target, converter) => {
      return converter.convert(source, target);
    }, undefined as any);
  }
```

5.7 Primo normalizer della catena

core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L18-L26

```ts
  convert(source: Occ.Product, target?: Product): Product {
    if (target === undefined) {
      target = { ...(source as any) } as Product;
    }
    if (source.images) {
      target.images = this.normalize(source.images);
    }
    return target;
  }
```

5.8 Espansione degli scope

core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53

```ts
  expand(model: string, scopes: string[]): string[] {
    const scopesConfig = this.config?.backend?.loadingScopes?.[model];

    if (scopesConfig) {
      const expandedScopes = [...scopes];
      let i = expandedScopes.length;

      while (i > 0) {
        i--;
        const includedScopes = scopesConfig[expandedScopes[i]]?.include;
        if (includedScopes) {
          for (const includedScope of includedScopes) {
            if (!expandedScopes.includes(includedScope)) {
              expandedScopes.splice(i, 0, includedScope);
```

```
            i++;
          }
        }
      }
    }

    return expandedScopes;
  }

  return scopes;
}
```

5.9 Fusione delle richieste

core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L53-L74

```ts
      const groupedModels = Object.values(groupedModelsSet);

      if (groupedModels.length === 1) {
        // only one scope for url, we can pass the data straightaway
        result.push({
          ...groupedModels[0].scopedData,
          data$: dataFactory?.(url),
        });
      } else {
        // multiple scopes per url
        // we have to split the model per each scope
        const data$ = dataFactory?.(url).pipe(shareReplay(1));

        groupedModels.forEach((modelData) => {
          result.push({
            ...modelData.scopedData,
            data$: data$?.pipe(
              map((data) => extractFields<T>(data, modelData.fields))
            ),
          });
        });
      }
```

5.10 Interceptor OCC

core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L43-L48

```ts
  protected requiresWithCredentials(request: HttpRequest<any>): boolean {
    return (
      this.occConfig?.useWithCredentials !== undefined &&
```

```
      request.url.indexOf(this.occConfig?.prefix ?? '') > -1
    );
  }
```

core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68

```ts
  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    if (request.url.includes(this.occEndpoints.getBaseUrl())) {
      request = request.clone({
        setParams: {
          lang: this.activeLang ?? '',
          curr: this.activeCurr ?? '',
        },
      });
    }

    return next.handle(request);
  }
```

core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L57

```ts
    return next.handle(request).pipe(
      switchMap((event) => {
        // When event `HttpEventType.Sent` happens, let's start counting time for timeout.
        // But when event `HttpEventType.Response` is received, `switchMap` will unsubscribe from the
following timeout observable.
        if (event.type === HttpEventType.Sent) {
          return NEVER.pipe(startWith(event), timeout(timeoutValue));
        }
        return of(event);
      }),
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.baseUrl	string	nessuno (validator obbligatorio)	core-libs/core/src/occ/config/occ-config.ts:L15-L15	core-libs/core/src/occ/services/occ-endpoints.service.ts:L93-L96; core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L95-L99; core-libs/core/src/occ/config/occ-config-validator.ts:L9-L17

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.prefix	string	'/occ/v2/' (core-libs/core/src/occ/config/default-occ-config.ts:L9-L15)	core-libs/core/src/occ/config/occ-config.ts:L16-L16	core-libs/core/src/occ/services/occ-endpoints.service.ts:L240-L248; core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L46-L46
backend.occ.useWithCredentials	boolean	undefined	core-libs/core/src/occ/config/occ-config.ts:L17-L21	core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L45-L45
backend.occ.endpoints	OccEndpoints	per modulo (prodotto, site context, utente, cost center, CMS)	core-libs/core/src/occ/config/occ-config.ts:L23-L23	core-libs/core/src/occ/services/occ-endpoints.service.ts:L174-L174, core-libs/core/src/occ/services/occ-endpoints.service.ts:L197-L197
backend.occ.endpoints.product.<scope>	string	default, list, details, … (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28); variants dalla feature lib	core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L14-L27	core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L54-L59
backend.media.baseUrl	string	undefined	core-libs/core/src/occ/config/occ-config.ts:L25-L31	core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L96-L96; core-libs/storefront/shared/components/media/media.service.ts:L399-L402
backend.media.prefix	string	undefined	core-libs/core/src/occ/config/occ-config.ts:L32-L37	core-libs/storefront/shared/components/media/media.service.ts:L405-L412 (solo con toggle enableMediaPrefix)
backend.loadingScopes.<model>.<scope>.include	string[]	product.details.include = ['list','variants'] (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58)	core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L31	core-libs/core/src/occ/services/loading-scopes.service.ts:L38-L38
backend.loadingScopes.<model>.<scope>.maxAge	number (s)	0	core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L31	core-libs/core/src/occ/services/loading-scopes.service.ts:L61-L65
backend.loadingScopes.<model>.<scope>.reloadOn	Type<CxEvent>[]	[]	core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L31	core-libs/core/src/occ/services/loading-scopes.service.ts:L75-L77
backend.timeout.server	number (ms)	20000 (core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15)	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L33	core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L77-L84

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`backend.timeout.browser`	`number` (ms)	`undefined` (nessun timeout)	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L33	core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L77-L84
`context.baseSite` (fallback del baseSite)	`string[]`	—	site context	core-libs/core/src/occ/services/occ-endpoints.service.ts:L37-L42
`HttpContext HTTP_TIMEOUT_CONFIG`	`HttpTimeoutConfig`	`undefined`	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L18	core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L78-L78
`HttpContext OCC_HTTP_TOKEN. sendUserIdAsHeader`	`string | boolean`	`{}`	core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20	feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L53-L69
Meta tag `occ-backend-base-url` / `media-backend-base-url`	`string`	placeholder ignorato	core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L12-L16	core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L18-L30
Feature toggle `enableHierarchicalAddressFormat`	`boolean`	—	FeatureToggles	core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L51-L53
Feature toggle `b2bCheckoutShippingAddressFilter`	`boolean`	—	FeatureToggles	core-libs/core/src/occ/adapters/user/occ-user-cost-centers.adapter.ts:L31-L55

Esempio di configurazione applicativa (illustrativo, usa solo chiavi della tabella):

```ts
// Esempio illustrativo, non presente nel repo
provideConfig(<OccConfig>{
  backend: {
    occ: {
      baseUrl: 'https://api.example.com',
      endpoints: {
        product: {
          list: 'products/${productCode}?
fields=code,name,price(formattedValue),images(DEFAULT,galleryIndex),baseProduct',
        },
      },
    },
    loadingScopes: {
      product: { details: { include: ['list', 'variants'], maxAge: 60 } },
    },
    timeout: { browser: 30_000 },
  },
});
```

Il merge profondo della configurazione (descritto nel capitolo 11) fa sì che il solo scope `list` venga sovrascritto lasciando invariati gli altri scope di `product`: `configFactory` esegue `deepMerge({}, inject(DefaultConfig), inject(RootConfig))` e `deepMerge` ricorre sugli oggetti annidati (core-libs/core/src/config/config-tokens.ts:L10-L11, core-libs/core/src/config/utils/deep-merge.ts:L38-L45). Gli array (es. `include`) invece vengono sostituiti, non concatenati.

---

7. Commenti e note notevoli dal sorgente

- `// TODO: Can we reuse getEndpointFromConfig in this method? Should we change behavior of this function?` — riconosce la doppia semantica tra `getEndpointFromConfig` (senza fallback) e `getEndpointForScope` (con fallback) (core-libs/core/src/occ/services/occ-endpoints.service.ts:L195-L195).
- `// TODO 4.0: default to encodeVariable = true` in `StringTemplate.resolve`: il default è ancora `false`, ma `buildUrl` passa esplicitamente `true` (core-libs/core/src/config/utils/string-template.ts:L24-L24, core-libs/core/src/occ/services/occ-endpoints.service.ts:L121-L123).
- `// CHECK SONAR` (core-libs/core/src/occ/utils/occ-fields.ts:L141-L141).
- `// TODO(#8878): Which models we can remove from here?` nel namespace `Occ` (core-libs/core/src/occ/occ-models/occ.models.ts:L3066-L3066).
- `// Uncomment this when occ gets configured` sopra la variante di `productReferences` con `referenceType` nel template (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L32-L34).
- `// Max limit of ProductSearch OCC` (`CHUNK_SIZE = 100` in `searchByCodes`) (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L73-L107).
- JSDoc di `LegacyOccCmsComponentAdapter`: "Before 1905, the OCC CMS component API required was using POST method ..." (core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts:L16-L25).
- JSDoc di `normalizeImageUrl`: "For media, we use the `backend.media.baseUrl` by default, but fallback to `backend.occ.baseUrl` if none provided." (core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L84-L90).
- JSDoc di `expand`: "I.e. if 'details' scope includes 'list' scope by configuration, it'll return ['details', 'list']" — l'ordine dell'esempio è opposto a quello reale e a quello dello spec (`['base', 'list']`) (core-libs/core/src/occ/services/loading-scopes.service.ts:L20-L20, core-libs/core/src/occ/services/loading-scopes.service.spec.ts:L52-L55).
- `// CXSPA-11104: Revise type to avoid redundant types - typescript:S6571` (core-libs/core/src/util/try-normalize-http-error.ts:L28-L28).
- Nessun `@deprecated` nei file non-spec di `core-libs/core/src/occ/` e `core-libs/core/src/http/` (verifica del dossier con `grep -rn "@deprecated"`).
- Il nome file `prduct-availability.adapter.ts` contiene un refuso, importato così da `ProductOccModule` (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L11-L11).

8. Extension point

EXTENSION POINT	MECCANISMO	DOVE	RICETTA
`backend.occ.endpoints.<key>[.<scope>]`	config: cambiare path o `fields` di uno scope senza codice	`core-libs/core/src/occ/services/occ-endpoints.service.ts:L196-L225`	52 — Ricette OCC (campi OCC aggiuntivi)
`OccEndpoints`	module augmentation `declare module '@spartacus/core'` per tipizzare endpoint nuovi	`core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L33`, esempio `feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L97-L98`	52 (endpoint custom)
`BackendConfig`	module augmentation (come fa `BackendHttpTimeoutConfig`)	`core-libs/core/src/http/http-timeout/http-timeout.config.ts:L35-L41`	50 — Ricette config/DI
`ProductAdapter`, `ProductSearchAdapter`, `ProductReferencesAdapter`, `ProductReviewsAdapter`, `ProductAvailabilityAdapter`	`{ provide: ProductAdapter, useClass: MyAdapter }`	`core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30`, `core-libs/core/src/occ/adapters/product/product-occ.module.ts:L34-L73`	52 (adapter/connector custom)
`CmsPageAdapter`, `CmsComponentAdapter` (alternativa `LegacyOccCmsComponentAdapter`)	provider della classe astratta	`core-libs/core/src/occ/adapters/cms/cms-occ.module.ts:L16-L34`, `core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts:L16-L57`	52
`*_NORMALIZER` / `*_SERIALIZER` (es. `PRODUCT_NORMALIZER`)	multi-provider aggiuntivo: il nuovo converter riceve il `target` già prodotto dai precedenti	`core-libs/core/src/util/converter.service.ts:L130-L137`, `core-libs/core/src/product/connectors/product/converters.ts:L11-L13`	52 (converter custom)
Normalizer esistenti con metodi `protected` (es. `ProductNameNormalizer.normalizeSlug/normalize`)	ereditarietà + `useExisting` della sottoclasse. Nota: ridefinire solo il campo `reservedSlugCharacters` non ha effetto, perché `slugRegex` (privato) è già calcolato dalla classe base durante l'inizializzazione dei suoi campi	`core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts:L42-L64`	52
`OccEndpointsService` (`getHttpParamsFromQueryParams` è `protected`)	sostituzione del servizio root	`core-libs/core/src/occ/services/occ-endpoints.service.ts:L152-L168`	50

EXTENSION POINT	MECCANISMO	DOVE	RICETTA
`OccRequestsOptimizerService.scopedDataLoad(…, dataFactory)`	data factory custom per altri modelli a scope	core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L34-L44	52
`backend.loadingScopes` / `LoadingScopes`	config di `include`, `maxAge`, `reloadOn`	core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L31	52
`HTTP_INTERCEPTORS`	interceptor custom; quelli OCC sono `useExisting` della classe, quindi sostituibili fornendo la classe	core-libs/core/src/occ/base-occ.module.ts:L17-L35, core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts:L18-L38	50 (interceptor custom)
`HTTP_TIMEOUT_CONFIG`, `OCC_HTTP_TOKEN`	`HttpContext` per singola richiesta	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L18, core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20	52
`InterceptorUtil` + `USE_CLIENT_TOKEN` & co.	header di segnalazione verso gli interceptor auth/ASM	core-libs/core/src/occ/utils/interceptor-util.ts:L9-L47	22 — Autenticazione
`MetaTagConfigModule` / `provideConfigFromMetaTags`	`baseUrl` iniettato a runtime via `<meta>`	core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L37-L42	50

Nota sull'API pubblica: `OccConfig`, `OccEndpointsService`, gli interceptor, gli adapter e i normalizer OCC sono raggiungibili tramite `export * from './src/occ/index'` (core-libs/core/public_api.ts:L27-L27), `ConverterService` tramite `./src/util/index` (core-libs/core/public_api.ts:L35-L35), adapter/token prodotto tramite `./src/product/index` (core-libs/core/public_api.ts:L29-L29). Alcuni file non sono riesportati dai barrel OCC (per esempio i default config degli endpoint e `loading-scopes-config`): vedi trappola 9.17 e il capitolo 54.

9. Trappole, casi limite, comportamenti controintuitivi

1. `useWithCredentials: false` **attiva comunque le credenziali.** La condizione è `!== undefined`, non `=== true` (core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L45-L45). Il test intitolato a `useWithCredentials: false` in realtà omette la chiave (core-libs/core/src/occ/interceptors/with-credentials.interceptor.spec.ts:L73-L81). Per disattivarle bisogna non impostare la chiave.
2. **Match per sottostringa sul prefix.** Anche host terzi con lo stesso path ricevono `withCredentials`; con `prefix` vuoto `indexOf('')` vale 0 e tutte le richieste ne sono coinvolte (core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L46-L46, core-libs/core/src/occ/interceptors/with-credentials.interceptor.spec.ts:L151-L160).

3. `lang`/`curr` **solo sotto il baseSite.** `SiteContextInterceptor` confronta l'URL con `getBaseUrl()`, che include il baseSite: le chiamate costruite con `{ baseSite: false }` (come `baseSites` in `OccSiteAdapter`) non ricevono `lang`/`curr` **quando il baseSite è noto** (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L58-L58, core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L85-L107). Se il baseSite è risolto dal backend (`SiteContextConfigInitializer`, senza `context.baseSite` statico: core-libs/core/src/site-context/site-context.module.ts:L34-L39, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L47-L47), `activeBaseSite` è ancora `undefined` e `urlPathJoin` scarta la parte vuota conservando lo slash finale del prefix (core-libs/core/src/occ/services/occ-endpoints.service.ts:L98-L101, core-libs/core/src/occ/utils/occ-url-util.ts:L14-L24): `getBaseUrl()` termina con il prefix, la chiamata `basesites` lo contiene e riceve `lang=`/`curr=` vuoti (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L58-L63).
4. **Uno scope mancante non fallisce.** `getEndpointForScope` logga un warning solo in dev mode, poi usa il `default`; se manca anche quello usa **il nome dell'endpoint** come path (core-libs/core/src/occ/services/occ-endpoints.service.ts:L213-L224). In produzione non c'è alcun warning.
5. `variants` **senza la feature** `product/variants`**.** Il default core fa includere `variants` a `details` (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58), ma il template `product.variants` arriva solo dalla feature lib (feature-libs/product/variants/occ/config/default-occ-product-variants-config.ts:L9-L16); senza di essa lo scope `variants` ricade sul template `default`, più pesante, con warning (core-libs/core/src/occ/services/occ-endpoints.service.ts:L213-L224). Poiché tutti gli scope `product` condividono path e differiscono solo per `fields`, il template `default` finisce comunque nello stesso gruppo di fusione (§3.9), allargando il `fields` della richiesta unica.
6. `isConfigured` **≠** `buildUrl`**.** `isConfigured('x', 'scope')` può essere `false` mentre `buildUrl` restituisce comunque un URL grazie al fallback (core-libs/core/src/occ/services/occ-endpoints.service.ts:L75-L79, core-libs/core/src/occ/services/occ-endpoints.service.ts:L170-L225).
7. `null` **cancella,** `undefined` **ignora.** Un `queryParams` con valore `null` rimuove anche un parametro già presente nel template (es. `fields`); `undefined` lo lascia (core-libs/core/src/occ/services/occ-endpoints.service.ts:L159-L164). `OccProductReferencesAdapter` sfrutta il caso `undefined` per `referenceType`/`pageSize` facoltativi (core-libs/core/src/occ/adapters/product/occ-product-references.adapter.ts:L34-L43).
8. **Ri-codifica della query del template.** Quando ci sono `queryParams`, la query del template viene ri-analizzata e ri-codificata con `encodeURIComponent` (per esempio `,` diventa `%2C`; le parentesi `()` non vengono codificate da `encodeURIComponent` e restano letterali); senza `queryParams` resta letterale (core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L146, core-libs/core/src/util/http-params-uri.encoder.ts:L21-L23). Le due forme sono equivalenti per il server, ma possono confondere confronti di URL nei test o nella cache.
9. **Placeholder non risolti.** `buildUrl` senza `urlParams` lascia i `${...}` letterali. `LegacyOccCmsComponentAdapter` costruisce l'endpoint `components` senza `userId`, quindi con il template di default `users/${userId}/cms/components` il placeholder resterebbe nell'URL (core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts:L16-L57, core-libs/core/src/cms/config/default-cms-config.ts:L9-L17). DA VERIFICARE a runtime: dipende dal template che l'applicazione configura per un backend pre-1905.
10. **Modelli senza scope scartati.** `getOptimalUrlGroups` non inserisce in nessun gruppo i modelli con `scope === undefined` (core-libs/core/src/occ/services/occ-fields.service.ts:L67-L69), quindi questi **mancano** dall'array restituito da `scopedDataLoad`, che itera solo sui modelli raggruppati (core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L53-L73). L'effetto prodotto mappa solo quell'array (core-libs/core/src/product/store/effects/product.effect.ts:L48-L53) e per loro non emette alcuna azione: niente `LoadProductSuccess` e niente `LoadProductFail`. Il messaggio `Scoped product data does not exist` riguarda solo un elemento restituito con `data$` indefinito (core-libs/core/src/product/store/effects/product.effect.ts:L82-L89).

11. `extractFields` **con** `DEFAULT`**.** Se i fields contengono `BASIC`/`DEFAULT`/`FULL` non viene estratto nulla: lo scope `default` di prodotto (che inizia con `DEFAULT,…`) riceve l'intera risposta fusa (core-libs/core/src/occ/utils/occ-fields.ts:L106-L139).
12. `splitFields` **e il carattere** `=`**.** I parametri sono divisi con `split('=')` prendendo solo l'indice `1`: un valore che contiene `=` viene troncato (core-libs/core/src/occ/services/occ-fields.service.ts:L95-L98).
13. `pipeable` **decide all'istante.** L'operatore restituito da `pipeable` è scelto alla sua creazione: se in quel momento il token non ha converter si ottiene l'identità, e un converter fornito più tardi da un modulo lazy non verrà applicato a quell'operatore, anche se la cache viene svuotata (core-libs/core/src/util/converter.service.ts:L80-L88, core-libs/core/src/util/converter.service.ts:L38-L45). Negli adapter OCC `pipeable` è invocato a ogni chiamata (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L29-L33), quindi il problema riguarda operatori memorizzati e riusati.
14. **Converter senza** `multi: true`**.** `UnifiedInjector.getMulti` emette l'errore `Multi-providers mixed with single providers for ...` dentro il `filter` dell'Observable (core-libs/core/src/lazy-loading/unified-injector.ts:L83-L88). `ConverterService` lo consuma con `getLastValueSync`, che si sottoscrive con la sola callback `next` (core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L22): RxJS rilancia l'errore in modo asincrono come unhandled error (deps:rxjs-7.8.2/package/dist/cjs/internal/util/reportUnhandledError.js:L6-L15). Intanto `ConverterService` non trova converter e restituisce la sorgente non convertita, cioè l'operatore identità (core-libs/core/src/util/converter.service.ts:L83-L87): la conversione viene saltata in silenzio.
15. **Doppio slash nelle immagini.** `normalizeImageUrl` concatena baseUrl e url senza `urlPathJoin`: un baseUrl con slash finale e un url con slash iniziale producono `//` (core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L91-L100). Inoltre il normalizer usa `||` mentre `MediaService.getBaseUrl()` usa `??`: con `backend.media.baseUrl = ''` il primo ricade su `occ.baseUrl`, il secondo no (core-libs/storefront/shared/components/media/media.service.ts:L399-L402).
16. `ProductReferenceNormalizer` **non registrato.** È esportato ma non è legato a `PRODUCT_NORMALIZER` in `ProductOccModule` (core-libs/core/src/occ/adapters/product/converters/index.ts:L8-L8, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51).
17. **File non esportati dai barrel.** `OccUserCostCenterAdapter`, `OccAddressListNormalizer`, i default config degli endpoint (`default-occ-product-config`, `default-occ-user-config`, …), `product-occ-config` e `loading-scopes-config` non sono riesportati dai barrel OCC (core-libs/core/src/occ/adapters/user/index.ts:L7-L15, core-libs/core/src/occ/adapters/user/converters/index.ts:L7-L8, core-libs/core/src/occ/adapters/product/index.ts:L7-L13, core-libs/core/src/occ/index.ts:L7-L17).
18. `loadBaseSite` **non normalizza.** A differenza di `loadBaseSites`, `OccSiteAdapter.loadBaseSite` non applica `BASE_SITE_NORMALIZER`, quindi `stores`→`baseStore` non avviene (core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L85-L120, core-libs/core/src/occ/adapters/site-context/converters/base-site-normalizer.ts:L18-L28).
19. `tryNormalizeHttpError` **e** `error: null`**.** Il codice accede a `error.error.errors` senza optional chaining: un `HttpErrorResponse` con `error: null` genera un `TypeError` (core-libs/core/src/util/try-normalize-http-error.ts:L21-L57). Angular produce davvero `error: null`: il backend XHR imposta `body = null` per una risposta di errore JSON con corpo vuoto e lo passa come `error` all'`HttpErrorResponse` (angular:packages/common/http/src/xhr.ts:L265-L265, angular:packages/common/http/src/xhr.ts:L298-L307).
20. **Timeout solo in SSR.** In browser il timeout HTTP è disabilitato di default; in SSR vale 20 s e il timer parte solo dopo `Sent` (core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15, core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L57).
21. `maxAge` **ignorato lato server.** Il trigger di ricarica per età è attivo solo con `isPlatformBrowser` (core-libs/core/src/product/services/product-loading.service.ts:L157-L191).

22. **baseSite "provvisorio".** `OccEndpointsService` si sottoscrive a `BaseSiteService.getActive()` nel costruttore senza unsubscribe; prima della prima emissione usa il baseSite di default della configurazione (core-libs/core/src/occ/services/occ-endpoints.service.ts:L37-L55).
23. `ProductOccModule` **va importato.** Non fa parte di `BaseCoreModule`: senza di esso non c'è binding per `ProductAdapter` (core-libs/core/src/base-core.module.ts:L38-L42, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L15-L15).
24. **Cost center: slash e** `fields` **sovrascritto.** Il template `getActiveCostCenters` inizia con `/` (innocuo grazie a `cleanSlashes`); con il toggle `b2bCheckoutShippingAddressFilter` il `fields` del template viene sovrascritto dal query param (core-libs/core/src/occ/adapters/cost-center/default-occ-cost-centers-config.ts:L13-L14, core-libs/core/src/occ/adapters/user/occ-user-cost-centers.adapter.ts:L31-L55).

---

10. Come fare debug

- **Quale URL viene chiamato?** In un test o da console, inietta `OccEndpointsService` e confronta `getRawEndpointValue('product', 'details')` (template, con fallback) con `isConfigured('product', 'details')` (senza fallback): se il primo restituisce il template `default` e il secondo `false`, lo scope non è configurato (core-libs/core/src/occ/services/occ-endpoints.service.ts:L63-L79). In dev mode cerca in console il warning `endpoint configuration missing for scope` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L213-L217).
- **URL senza host o senza baseSite.** `getBaseUrl()` usa `''` se `baseUrl` manca e il baseSite di default finché `BaseSiteService` non emette (core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102, core-libs/core/src/occ/services/occ-endpoints.service.ts:L37-L42). Controlla anche l'errore del validator (core-libs/core/src/occ/config/occ-config-validator.ts:L9-L17) e il meta tag `occ-backend-base-url` nell'HTML servito (core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L18-L23).
- **Una sola richiesta per più scope.** Nel tab Network, per un PDP compare una `GET products/<code>?fields=…` con i fields fusi: è `OccRequestsOptimizerService` che ha raggruppato gli scope (core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L55-L74). Se invece vedi più richieste allo stesso prodotto, gli URL differiscono per qualcosa oltre a `fields` (core-libs/core/src/occ/services/occ-fields.service.ts:L89-L116).
- **Converter che non scatta.** Verifica che il provider abbia `multi: true` (altrimenti `UnifiedInjector` produce un errore non gestito asincrono e la conversione viene saltata in silenzio, vedi trappola 14) e che il modulo che lo fornisce sia caricato prima della creazione dell'operatore `pipeable` (core-libs/core/src/lazy-loading/unified-injector.ts:L75-L93, core-libs/core/src/util/converter.service.ts:L80-L88). Un breakpoint in `convertSource` mostra l'ordine dei converter applicati (core-libs/core/src/util/converter.service.ts:L130-L137).
- `lang`**/**`curr` **mancanti.** L'URL deve contenere esattamente `getBaseUrl()` (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68).
- **Cookie inattesi.** Controlla se `useWithCredentials` è definito (anche `false`) e se il prefix compare nell'URL (core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L43-L48).
- **Richieste abortite in SSR.** Il messaggio `Request to URL '...' exceeded expected time of ...ms and was aborted.` arriva da `HttpTimeoutInterceptor` (core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L92-L124); si alza il limite globale con `backend.timeout.server` o per singola richiesta con `HTTP_TIMEOUT_CONFIG` (core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L77-L84).
- **Errori dello store prodotto.** Le azioni `LoadProductFail` contengono un `HttpErrorModel` con `details = Occ.ErrorModel[]` (core-libs/core/src/product/store/effects/product.effect.ts:L59-L91, core-libs/core/src/util/try-normalize-http-error.ts:L21-L57); con Redux DevTools si vede lo scope fallito in `meta`.

- **Nei test unitari** gli adapter user usano `MockOccEndpointsService`, che restituisce la chiave dell'endpoint prefissata da `/` e baseUrl vuoto: gli URL attesi nei test sono quindi del tipo `/addresses` e non quelli reali (core-libs/core/src/occ/adapters/user/unit-test.helper.ts:L10-L36).

---

11. Domande di autoverifica

1. **Perché** `inject(OccConfig)` **restituisce la configurazione globale e non un oggetto separato?** Perché `OccConfig` è dichiarata `providedIn: 'root'` con `useExisting: Config` (core-libs/core/src/occ/config/occ-config.ts:L42-L48).
2. **Che URL produce** `buildUrl('product', { urlParams: { productCode: 'X' }, scope: 'foo' })` **se lo scope** `foo` **non esiste?** Il template dello scope `default` di `product` (con warning solo in dev mode); se mancasse anche quello, il path sarebbe `product` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L224).
3. **Come si elimina il parametro** `fields` **del template per una singola chiamata?** Passando `queryParams: { fields: null }`: `null` provoca `httpParams.delete(key)` sui parametri analizzati dal template (core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L168).
4. **In che ordine vengono applicati** `ProductImageNormalizer` **e** `ProductNameNormalizer`**, e chi crea il** `target`**?** Nell'ordine di registrazione in `ProductOccModule` (prima Image, poi Name); `convertSource` parte da `target = undefined` e il primo converter crea la copia della sorgente (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51, core-libs/core/src/util/converter.service.ts:L130-L137, core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L18-L26).
5. **Con la configurazione di default, quali scope vengono caricati per** `ProductService.get(code, 'details')`**?** `['list', 'variants', 'details']`: `expand` inserisce gli scope inclusi prima di quello che li include (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58, core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53, core-libs/core/src/occ/services/loading-scopes.service.spec.ts:L52-L60).
6. **Perché tre scope di prodotto possono produrre una sola richiesta HTTP?** Perché `getOptimalUrlGroups` raggruppa gli URL uguali a meno di `fields`, fonde i fields e `scopedDataLoad` condivide la richiesta con `shareReplay(1)`, estraendo poi le parti con `extractFields` (core-libs/core/src/occ/services/occ-fields.service.ts:L59-L82, core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L55-L74).
7. **Impostare** `useWithCredentials: false` **disattiva** `withCredentials`**?** No: la condizione è `!== undefined`, quindi anche `false` lo attiva per gli URL che contengono il prefix (core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L43-L48).
8. **Perché la chiamata a** `basesites` **non porta** `lang` **e** `curr`**?** Perché è costruita con `{ baseSite: false }` e `SiteContextInterceptor` aggiunge i parametri solo se l'URL contiene `getBaseUrl()`, che include il baseSite (core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L85-L107, core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68). Vale però solo quando il baseSite è noto: se è risolto dal backend tramite `SiteContextConfigInitializer`, `getBaseUrl()` termina con il prefix e la chiamata `basesites` riceve `lang=/curr=` vuoti (vedi trappola 3 in §9).
9. **Come può un adapter chiedere all'interceptor ASM di inviare l'id del cliente emulato?** Impostando `new HttpContext().set(OCC_HTTP_TOKEN, { sendUserIdAsHeader: true })`, come fa `OccProductSearchAdapter.search`; `UserIdHttpHeaderInterceptor` lo legge se `asm.userIdHttpHeader.enable` è attivo (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L46-L71, feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L49-L69).

10. **Qual è il timeout HTTP di default in browser e in SSR, e da quando si conta?** Nessuno in browser, 20 000 ms in SSR; il conteggio parte all'evento `HttpEventType.Sent` (core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15, core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L49-L57).

TAVOLA 21.1 **4.1 Product load: dal facade alla risposta normalizzata**

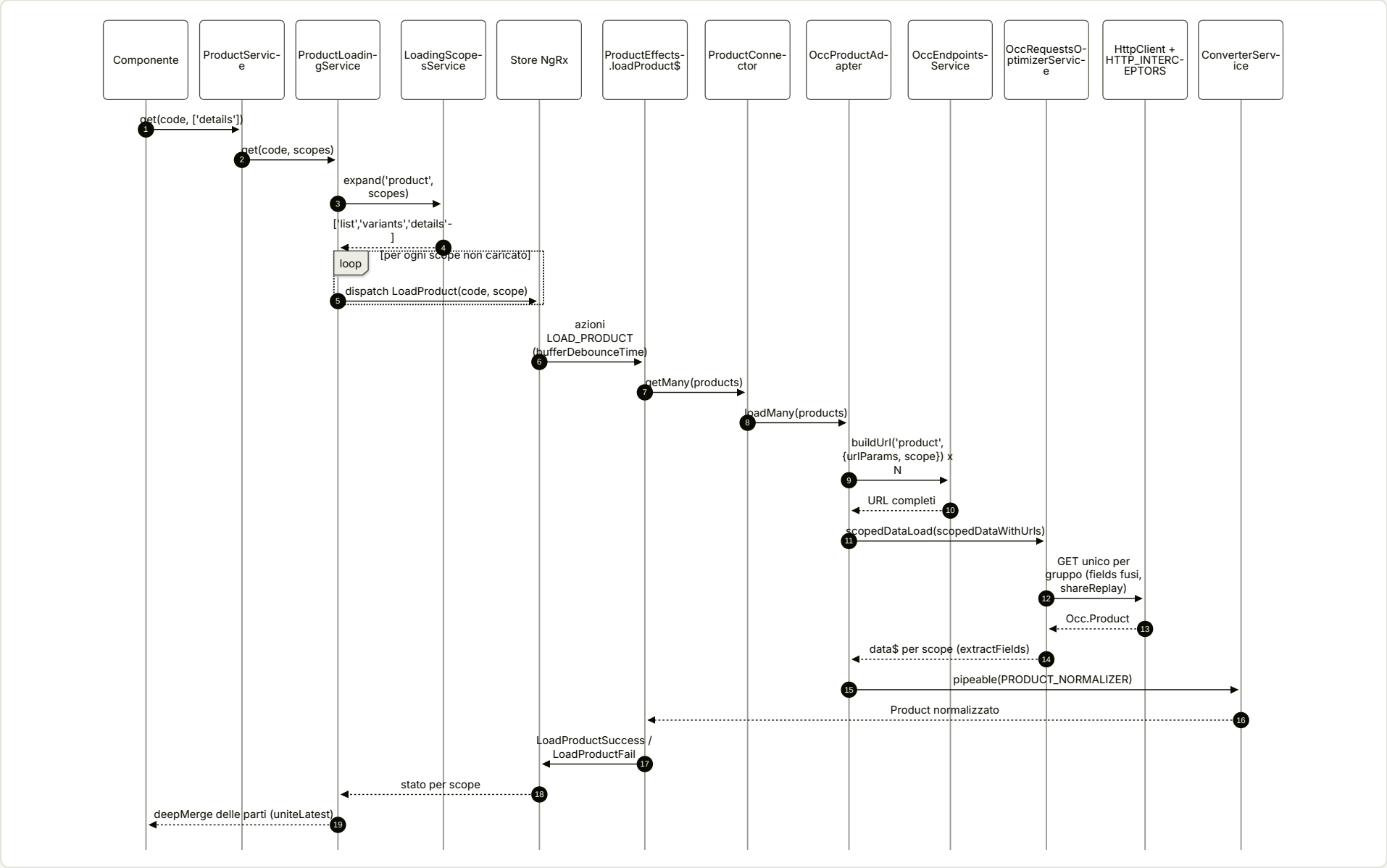
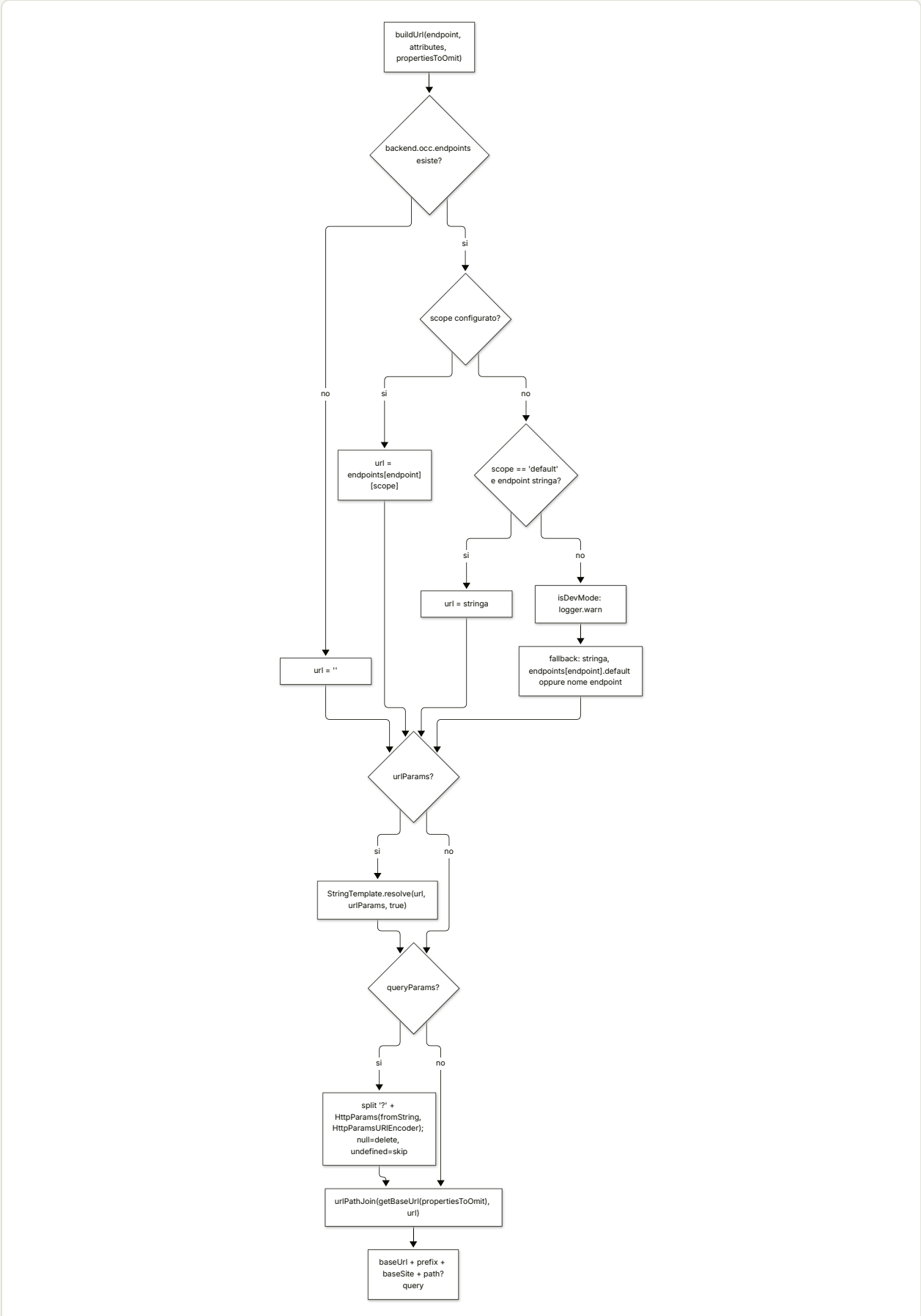


TAVOLA 21.2 **4.2 Costruzione dell'URL in buildUrl**



22

Autenticazione: OAuth, token, interceptor, refresh

Flow OAuth, conservazione del token, 401 → refresh → retry, token client, guard di auth.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Il sottosistema di autenticazione di `@spartacus/core` risolve quattro problemi distinti, che conviene tenere separati mentalmente:

1. **Ottenere un token utente** dall'Authorization Server di SAP Commerce, con uno dei tre flow OAuth supportati (Authorization Code con PKCE, Implicit, Resource Owner Password). Il flow non è configurato esplicitamente: è *dedotto* da `responseType` — core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L151-L164, enum `OAuthFlow` in core-libs/core/src/auth/user-auth/models/oauth-flow.ts:L10-L23.
2. **Conservare il token** e lo `userId` OCC (`current` / `anonymous` / customer emulato) in memoria e, in parte, nel `localStorage` — core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L18-L37, core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L19-L57.
3. **Allegare il token alle chiamate OCC** e gestire in modo trasparente la scadenza: 401 → refresh → retry, oppure logout forzato quando anche il refresh token è inutilizzabile — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L31-L88, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L247-L310.
4. **Token "client"** (grant `client_credentials`) per gli endpoint OCC che lo richiedono con l'Authorization Server legacy (es. registrazione utente). È un modulo separato, dichiarato esplicitamente un workaround rimovibile — core-libs/core/src/auth/client-auth/client-auth.module.ts:L12-L29.

Attorno a questo nucleo ruotano: il redirect alla pagina "di prima" dopo il login (`AuthRedirectService`), i guard (`AuthGuard`, `NotAuthGuard`, `CustomLoginGuard`, `OAuthCallbackGuard`, `FederatedLoginGuard`), il *federated login* (login su un dominio separato), le notifiche di logout tra tab e gli eventi `LoginEvent`/`LogoutEvent`.

Spartacus non implementa il protocollo OAuth da sé: lo delega alla libreria **angular-oauth2-oidc** (`OAuthModule.forRoot()` importato in core-libs/core/src/auth/user-auth/user-auth.module.ts:L98-L100), incapsulata da `OAuthLibWrapperService` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L32-L53). Spartacus però si riserva il controllo degli endpoint: il tipo `AuthLibConfig` *omette* dal `AuthConfig` della libreria `clientId`, `loginUrl`, `tokenEndpoint`, `revocationEndpoint` ecc., che vengono invece calcolati da `AuthConfigService` — core-libs/core/src/auth/user-auth/config/auth-config.ts:L20-L30.

> Comportamento interno di angular-oauth2-oidc 20.0.2 (verificato sul pacchetto pubblicato): la lib scrive il token campo per campo tramite `OAuthStorage.setItem` (quindi in `AuthStorageService`), sia dopo lo scambio del code sia dopo il refresh — `deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L2094-L2111`; `refreshToken()` passa da `storeAccessTokenResponse` e poi emette `token_received` — `deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L1605-L1609`; nel code flow `tryLogin()` risolve sempre `true` — `deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L2122-L2125`.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/auth/auth.module.ts	`AuthModule`: importa `UserAuthModule.forRoot()` **prima** di `ClientAuthModule.forRoot()` (ordine interceptor)
core-libs/core/src/auth/user-auth/user-auth.module.ts	`UserAuthModule`: default config, validator, initializer, interceptor, `OAuthStorage → AuthStorageService`, location initializer
core-libs/core/src/auth/user-auth/config/auth-config.ts	`AuthConfig` (abstract, `useExisting: Config`), `AuthLibConfig`
core-libs/core/src/auth/user-auth/config/default-auth-config.ts	`defaultAuthConfig`, `defaultAuthConfigFactory` (dipende dai feature toggle), token deprecato `USE_AUTHORIZATION_CODE_FLOW_BY_DEFAULT`
core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts	`AuthConfigInitializer`: `client_id` e `redirectUri` dinamici per base site
core-libs/core/src/auth/user-auth/config/base-url-config-validator.ts	warning se manca un base URL
core-libs/core/src/auth/user-auth/services/auth-config.service.ts	`AuthConfigService`: endpoint prefissati, flow dedotto
core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts	`OAuthLibWrapperService`: unico punto di contatto con `OAuthService` della lib
core-libs/core/src/auth/user-auth/services/auth-storage.service.ts	`AuthStorageService`: implementazione in memoria di `OAuthStorage`
core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts	sincronizzazione `{ userId, token, redirectUrl }` con `localStorage` chiave `auth`
core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts	`AuthHttpHeaderService`: header, refresh, retry, logout su refresh scaduto
core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts	`ExpiredRefreshTokenHandler`, `EXPIRED_REFRESH_TOKEN_HANDLERS`
core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts	`AuthRedirectService`: memorizza e ripristina l'URL pre-login
core-libs/core/src/auth/user-auth/services/auth-redirect-storage.service.ts	contenitore della redirect URL
core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts	riconosce le route con `authFlow: true`
core-libs/core/src/auth/user-auth/services/auth-multisite-isolation.service.ts	suffisso `|baseSite` allo uid per siti isolati
core-libs/core/src/auth/user-auth/facade/auth.service.ts	`AuthService`: facade di login/logout, stato refresh/logout, client ASM
core-libs/core/src/auth/user-auth/facade/user-id.service.ts	`UserIdService`: `current` / `anonymous` / emulato
core-libs/core/src/auth/user-auth/facade/auth-notification.service.ts	`AuthNotificationService`: logout cross-tab via `BroadcastChannel`
core-libs/core/src/auth/user-auth/facade/csrf-state.service.ts	`CsrfStateService`: condivide il CSRF fra guard e form

PATH	RUOLO
core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts	AuthInterceptor
core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts	TokenRevocationInterceptor
core-libs/core/src/auth/user-auth/http-interceptors/index.ts	array interceptors (multi HTTP_INTERCEPTORS)
core-libs/core/src/auth/user-auth/guards/auth.guard.ts	AuthGuard
core-libs/core/src/auth/user-auth/guards/not-auth.guard.ts	NotAuthGuard
core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts	CustomLoginGuard (CSRF + retry)
core-libs/core/src/auth/user-auth/guards/oauth-callback.guard.ts	OAuthCallbackGuard
core-libs/core/src/auth/user-auth/guards/federated-login.guard.ts	FederatedLoginGuard
core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts	UserAuthEventBuilder: LoginEvent, LogoutEvent
core-libs/core/src/auth/user-auth/store/actions/login-logout.action.ts	azioni [Auth] Login / [Auth] Logout
core-libs/core/src/auth/user-auth/models/auth-token.model.ts	AuthToken
core-libs/core/src/auth/user-auth/utils/oauth-constants.ts	OAUTH_REDIRECT_FLOW_KEY
core-libs/core/src/auth/client-auth/client-auth.module.ts	ClientAuthModule (store + ClientTokenInterceptor)
core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts	ClientTokenInterceptor
core-libs/core/src/auth/client-auth/services/client-token.service.ts	facade sullo store del client token
core-libs/core/src/auth/client-auth/services/client-authentication-token.service.ts	POST grant_type=client_credentials
core-libs/core/src/auth/client-auth/services/client-error-handling.service.ts	refresh client token + retry
core-libs/core/src/auth/client-auth/services/cross-site-request-forgery.service.ts	GET del token CSRF (custom login page)
core-libs/core/src/auth/client-auth/store/effects/client-token.effect.ts	ClientTokenEffect (exhaustMap)
core-libs/core/src/auth/client-auth/store/client-auth-store.module.ts	feature store client-auth
core-libs/core/src/federated-login/services/federated-login.service.ts	FederatedLoginService

PATH	RUOLO
core-libs/core/src/federated-login/services/federated-login-context-serializer.service.ts	contesto `ctx` in Base64 URL-safe
core-libs/core/src/federated-login/config/federated-login-config.ts	FederatedLoginConfig

3. Come funziona passo per passo

3.1 Wiring dei moduli e ordine degli interceptor

L'app demo importa `AuthModule.forRoot()` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L179-L179). `AuthModule` importa i due sottomoduli in un ordine preciso, motivato da un commento (core-libs/core/src/auth/auth.module.ts:L12-L24): il `ClientTokenInterceptor` deve essere registrato **dopo** l'`AuthInterceptor`, così è il primo a vedere i 401.

Perché "registrato dopo" significa "vede prima l'errore"? Angular adatta gli interceptor a classe (`HTTP_INTERCEPTORS`) con `reduceRight`: con l'array `[a, b, c]` la catena è `c(b(a(end)))` in costruzione, cioè in esecuzione `a` è il più esterno e `c` il più interno, quindi `c` è il primo a ricevere risposta ed errore — angular:packages/common/http/src/interceptor.ts:L233-L240. Gli interceptor utente sono `AuthInterceptor` e poi `TokenRevocationInterceptor`, entrambi `useExisting` (core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L12-L23); `ClientAuthModule.forRoot()` aggiunge il `ClientTokenInterceptor` (core-libs/core/src/auth/client-auth/client-auth.module.ts:L12-L29). Risultato: `AuthInterceptor` (esterno) → `TokenRevocationInterceptor` → `ClientTokenInterceptor` (interno). Approfondimento generale sugli interceptor: capitolo 06 e appendice A06.

`UserAuthModule.forRoot()` fornisce (core-libs/core/src/auth/user-auth/user-auth.module.ts:L102-L123):

- `provideDefaultConfigFactory(defaultAuthConfigFactory)` — la default config dipende dai feature toggle (§3.2);
- `provideConfigValidator(baseUrlConfigValidator)` — warning se mancano sia `authentication.baseUrl` sia `backend.occ.baseUrl`, salvo `requireHttps === false` (core-libs/core/src/auth/user-auth/config/base-url-config-validator.ts:L10-L19);
- un config initializer `AuthConfigInitializer`, ma solo con il toggle `asyncAuthConfigInitializer` (la factory altrimenti restituisce `null`) — core-libs/core/src/auth/user-auth/user-auth.module.ts:L108-L113;
- gli interceptor, il binding `OAuthStorage → AuthStorageService` (la libreria scrive e legge il token *attraverso* il servizio Spartacus) e due *location initializer*.

`FederatedLoginModule` invece arriva da `BaseCoreModule` (core-libs/core/src/base-core.module.ts:L44-L44) e fornisce solo la default config (core-libs/core/src/federated-login/federated-login.module.ts:L10-L17). `UserAuthEventModule` istanzia eagerly `UserAuthEventBuilder` iniettandolo nel costruttore (core-libs/core/src/auth/user-auth/events/user-auth-event.module.ts:L10-L15).

3.2 La configurazione di default dipende dai feature toggle

`defaultAuthConfig` descrive il caso "moderno" (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L48-L74): `client_id: 'mobile_android_public'`, endpoint `/oauth/token`, `/oauth/revoke`, `/oauth/authorize`, `responseType: 'code'`, `disablePKCE: false`, `redirectUri: '/oauth-callback'`, `customLoginPage` (`/csrf`, `/login`) e `initializerOptions 'auto'`. In `OAuthLibConfig` imposta inoltre `scope: ''`,

`customTokenParameters: ['token_type']`, `skipIssuerCheck: true`, `oidc: false` e altri flag — core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L54-L61.

`defaultAuthConfigFactory` legge tre toggle e "toglie pezzi" a quella config (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L76-L139):

AUTHORIZATIONCODEFLOWBYDEFAULT	ASYNCAUTHCONFIGINITIALIZER	OAUTHCALLBACKPAGE	RISULTATO
true	true	true	`defaultAuthConfig` invariato (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L83-L86)
true	true	false	senza `redirectUri` (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L87-L99)
true	false	—	senza `initializerOptions` e `redirectUri` (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L101-L115)
false	—	—	legacy ROPC: `mobile_android` + `client_secret: 'secret'`, `sendAuthHeaderOnRevoke: true`, `useClientTokens: true`, `disablePKCE: true`; rimossi `responseType`, `customLoginPage`, `initializerOptions`, `redirectUri` (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L117-L138)

I default dei toggle allo SHA analizzato sono: `authorizationCodeFlowByDefault: true` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029), `asyncAuthConfigInitializer: false` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1090-L1090), `oauthCallbackPage: false` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1111-L1111). Quindi, **con i default, si cade nella terza riga**: Authorization Code + PKCE, `client_id` `mobile_android_public`, custom login page abilitata, nessun `redirectUri` configurato (verrà calcolato a runtime, §3.4). Il JSDoc dichiara che il toggle `asyncAuthConfigInitializer` ha effetto solo con `authorizationCodeFlowByDefault` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L631-L637), ma `UserAuthModule` e `OAuthLibWrapperService` lo leggono da solo (core-libs/core/src/auth/user-auth/user-auth.module.ts:L108-L113, core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L56-L56); l'unico punto che richiede tutti e tre i toggle è `oauthCallbackPageEnabled()` (core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L160-L166). Sui feature toggle in generale: capitolo 42 e appendice A09.

`AuthConfigService` trasforma la config in URL: `getBaseUrl()` usa `authentication.baseUrl` o, in fallback, `backend.occ.baseUrl + '/authorizationserver'` (core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L54-L59); ogni endpoint passa per `prefixEndpoint`, che aggiunge `/` se manca e prefissa il base URL (core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L139-L145). `customLoginEnabled()` è vero se esiste l'oggetto `customLoginPage` (core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L105-L107). Il flow è dedotto da `responseType`: contiene `code` → `AuthorizationCode`; `token` → `ImplicitFlow`; altrimenti `ResourceOwnerPasswordFlow` (core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L151-L164).

3.3 Dove vive il token

`AuthStorageService` estende l'`OAuthStorage` della libreria ma non scrive in nessuno storage del browser: tiene l'intero token in un `BehaviorSubject<AuthToken>` inizializzato a `{}` (core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L18-L37). L'API per la libreria è sincrona: `getItem` legge il valore corrente con un subscribe/unsubscribe immediato; `setItem` e `removeItem` emettono un nuovo oggetto token (core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L85-L118). Le chiavi in `nonStringifiedOAuthLibKeys` (`PKCE_verifier`, `access_token`, `refresh_token`, `expires_at`, `nonce`…) passano in-

variate. Le altre arrivano dalla lib come stringhe JSON e vengono salvate nel token come oggetti (`JSON.parse` in `encode`, con fallback al valore grezzo), poi restituite alla lib come stringhe (`JSON.stringify` in `decode`): i nomi `encode`/`decode` sono invertiti rispetto a ciò che fanno (core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L22-L56, core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L111-L116).

Il modello `AuthToken` ha `access_token` e `access_token_stored_at` obbligatori; `refresh_token`, `expires_at`, `granted_scopes`, `token_type` opzionali (core-libs/core/src/auth/user-auth/models/auth-token.model.ts:L10-L35).

La persistenza vera la fa `AuthStatePersistenceService`: sincronizza con il browser storage, sotto la chiave `'auth'`, lo stato `{ userId, token, redirectUrl }` (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L19-L57), di default in `localStorage` (core-libs/core/src/state/services/state-persistence.service.ts:L44-L48). Il `refresh_token` **non** viene mai persistito "per minimizzare il rischio di hijacking" (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L77-L83). In lettura (`onRead`) ripristina token e redirect URL, imposta lo `userId` salvato o chiama `clearUserId()` (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L92-L104). Esiste anche un `isUserLoggedIn()` *sincrono*, che legge direttamente lo storage e serve al transfer state reducer (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L115-L121). Il meccanismo generico di persistenza è trattato nel capitolo 19.

`UserIdService` usa un `ReplaySubject<string>(1)`: finché nessuno chiama `setUserId`, `getUserId()` non emette (core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L24-L34). I valori notevoli sono `OCC_USER_ID_CURRENT = 'current'` e `OCC_USER_ID_ANONYMOUS = 'anonymous'` (core-libs/core/src/occ/utils/occ-user-ids.ts:L7-L8). `takeUserId(true)` lancia un `Error` se l'utente è anonimo (core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L57-L69); `isEmulated()` è vero quando lo userId non è né `anonymous` né `current`, cioè un customerId in emulazione ASM (core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L81-L88). Il JSDoc dichiara l'implementazione specifica per OCC (core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L15-L20).

3.4 Configurazione della libreria: `OAuthLibWrapperService`

Il costruttore chiama `initialize()` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L32-L53), che:

1. con `asyncAuthConfigInitializer` applica la config a ogni emissione di `configInitializerService.getStable('authentication')`; senza, la applica subito, sincrona (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L55-L64);
2. rilegge sempre il contesto (`detectContext()`, che agisce solo sul login domain — core-libs/core/src/federated-login/services/federated-login.service.ts:L62-L63) e, se il federated login è abilitato, riconfigura la lib con il `loginUrl` arricchito da `ctx=…` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L66-L73, core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L126-L139).

`generateBaseConfig` mappa la config Spartacus sui campi della lib (`tokenEndpoint`, `loginUrl`, `clientId`, `dummyClientSecret`, `revocationEndpoint`, `logoutUrl`, `userinfoEndpoint`, `issuer`, `redirectUri`) e poi spalma `OAuthLibConfig` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L96-L115). Se `redirectUri` non è configurato, `generateCustomerLoginConfig` lo calcola: `''` in SSR, l'origin federato sul login domain, altrimenti `location.origin` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L76-L94). `applyConfiguration` chiama `oAuthService.configure(config)` e, in modalità async, segnala `initialized` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L118-L123): tutti i metodi che parlano con la lib (`authorizeWithPasswordFlow`, `refreshToken`, `revokeAndLogout`,

`initLoginFlow`) attendono quel segnale in modalità async (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L157-L198, core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L239-L246).

Con l'initializer asincrono attivo, `AuthConfigInitializer` calcola `client_id` e `redirectUri` combinando il base site attivo e `authentication.initializerOptions` (core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L19-L53): il client id riceve il suffisso `_<baseSite>` (core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L55-L59); il redirect URI parte dal valore configurato o da `location.origin`, riceve `/<baseSite>` se `addBaseSiteToRedirectUri` lo consente e, con i tre toggle attivi (`oauthCallbackPage`, `authorizationCodeFlowByDefault`, `asyncAuthConfigInitializer`), un `redirectUri` relativo (default `/oauth-callback`) è accodato come path, mentre uno assoluto fa da base (core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L84-L113, core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L63-L63). Il valore `'auto'` vale `true` solo se `BASE_SITE_CONTEXT_ID` è fra i parametri di URL encoding del site context (core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L142-L146). Il meccanismo dei config initializer è nel capitolo 11.

3.5 Bootstrap: sincronizzazione e ritorno dall'Authorization Server

Il punto più delicato è *quando* Spartacus controlla se nell'URL c'è un `code` OAuth. Lo fa in un **location initializer**: i location initializer sono raccolti nel multi-token `LOCATION_INITIALIZED_MULTI` e risolti in un `Promise.all` fornito come `LOCATION_INITIALIZED` di Angular (core-libs/core/src/routing/routing.module.ts:L66-L71, core-libs/core/src/routing/routing.module.ts:L102-L105). Spartacus configura `initialNavigation: 'enabledBlocking'` (core-libs/storefront/router/app-routing.module.ts:L15-L18), che Angular mappa su `withEnabledBlockingInitialNavigation()` (angular:packages/router/src/router_module.ts:L237-L242): così il router attende quella promise prima della navigazione iniziale (angular:packages/router/src/provide_router.ts:L445-L452). Dettagli sull'ordine di bootstrap: capitolo 10.

`authInitializedFactory` (core-libs/core/src/auth/user-auth/user-auth.module.ts:L70-L80):

1. avvia **subito** `AuthStatePersistenceService.initSync()`, così token, userId e redirect URL letti dal `localStorage` sono in memoria; il commento spiega che altrimenti il `nonce` dell'Implicit Flow sarebbe `undefined`;
2. restituisce `checkOAuthParamsInUrl(...)`, che in browser attende `configInit.getStable()`, chiama `authService.refreshAuthConfig()` e poi `authService.checkOAuthParamsInUrl()`; in SSR risolve subito (core-libs/core/src/auth/user-auth/user-auth.module.ts:L39-L56).

`AuthService.refreshAuthConfig()` cambia il client id in `'asm_client'` se il code flow è default e ASM risulta abilitato, altrimenti riconfigura normalmente la lib (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L316-L327). ASM è considerato abilitato se `asm=true` è nella query, se `localStorage.asm_enabled === 'true'` o se il path contiene `assisted-service/emulate?` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L263-L284).

`AuthService.checkOAuthParamsInUrl()` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L90-L117):

1. chiama `oAuthLibWrapperService.tryLogin()`, che si iscrive all'evento `token_received` e invoca `oAuthService.tryLogin({ disableOAuth2StateCheck: true })`, risolvendo `{ result, tokenReceived }` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L258-L284); `tokenReceived` distingue il ritorno da una pagina di autorizzazione esterna dal token preso dallo storage (core-libs/core/src/auth/user-auth/models/oauth-try-login-response.ts:L7-L19);
2. imposta `userId = 'current'` solo se `result` è vero **e** c'è un `access_token` **e** l'utente non è emulato **e** non si usa il client ASM — il commento ricorda che nel code flow `result` è `true` anche senza login riuscito (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L101-L105);

3. solo se `tokenReceived` dispatcha `AuthActions.Login` e chiama `authRedirectService.redirect()` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L107-L114);
4. ogni errore è silenziato con `catch {}` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L116-L116).

In parallelo, `authNotificationInitializerFactory` avvia l'ascolto del `BroadcastChannel` solo con il toggle `propagateLogoutToAllTabs` (core-libs/core/src/auth/user-auth/user-auth.module.ts:L85-L92).

3.6 Avviare il login

- **Redirect (Authorization Code / Implicit)**: `AuthService.loginWithRedirect()` chiama `initLoginFlow()` e ritorna sempre `true` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L122-L125). `OAuthLibWrapperService.initLoginFlow()` scrive `oAuthRedirectCodeFlow='true'` in `localStorage` se il code flow non è default o se il federated login è abilitato (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L219-L225, chiave in core-libs/core/src/auth/user-auth/utils/oauth-constants.ts:L7-L7); sul login domain federato non avvia il flow ma rimanda al login dell'origin "perché il PKCE sia disponibile all'origin" (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L227-L237); altrimenti delega a `oAuthService.initLoginFlow()` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L239-L246). La chiave `oAuthRedirectCodeFlow` viene rimossa da `tryLogin()` se il token non arriva o in caso di errore (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L276-L288).
- **Credenziali (ROPC)**: `loginWithCredentials(uid, pwd)` decora lo uid con `|<baseSiteUid>` se il sito è `isolated` (core-libs/core/src/auth/user-auth/services/auth-multisite-isolation.service.ts:L16-L38), chiama `authorizeWithPasswordFlow` (→ `fetchTokenUsingPasswordFlow`), imposta `current`, dispatcha `Login` e fa redirect, inghiottendo gli errori (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L132-L153, core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L157-L168).
- **OTP**: `otpLoginWithCredentials(tokenId, tokenCode)` riusa il password flow con tokenId/tokenCode come credenziali, senza decorazione multisite (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L169-L186).

Custom login page. Col code flow default l'Authorization Server presenta un form di login servito da Spartacus stesso: la route `login` ha path `sign-in` e viene aggiunta la route `loginForm` con path `login` (core-libs/storefront/cms-structure/routing/default-routing-config.ts:L24-L33, core-libs/storefront/cms-structure/routing/default-routing-config.ts:L94-L111). Il componente CMS `ReturningCustomerLoginComponent` è protetto da `[NotAuthGuard, CustomLoginGuard]` (feature-libs/user/account/components/login-form/login-form.module.ts:L52-L54), come `VerifyOTPTokenComponent` (feature-libs/user/account/components/verification-token-form/verification-token-form.module.ts:L59-L61). `CustomLoginGuard` è un no-op se la custom login page non è configurata o in SSR (core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L61-L68); altrimenti chiede il token CSRF (GET con `withCredentials: true`, core-libs/core/src/auth/client-auth/services/cross-site-request-forgery.service.ts:L30-L35) come prova di una sessione valida dell'Authorization Server, lo salva in `CsrfStateService` e azzera il contatore dei redirect (core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L70-L75). Il form lo rilegge con il getter `csrf` (feature-libs/user/account/components/login-form/login-form-component.service.ts:L61-L63). In caso di errore CSRF il guard ritenta una sola volta (contatore in `localStorage` `login_redirect_count`, valido 15 s), poi porta a `home` con `authMessages.unrecoverableError` (core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L29-L38, core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L76-L101). `AuthService.getCsrfToken()` restituisce uno stream condiviso con `shareReplay({ bufferSize: 1, refCount: true })` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L67-L69), mentre `refreshCsrfToken()` rifà la GET (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L256-L258).

Pagina di callback. Con i tre toggle attivi esiste la route `oAuthCallback` (`oauth-callback`) (core-libs/storefront/cms-structure/routing/default-routing-config.ts:L94-L111) e `OAuthCallbackComponent` è mappato su `SpinnerComponent` con `OAuthCallbackGuard` (feature-libs/user/account/components/oauth-callback/oauth-callback.module.ts:L26-L37). Il guard in SSR lascia renderizzare la pagina; in browser, se l'utente è loggato chiama `redirect()` e ritorna `true`, altrimenti manda a `login` (core-libs/core/src/auth/user-auth/guards/oauth-callback.guard.ts:L32-L52).

3.7 Allegare il token alle richieste: `AuthInterceptor`

`AuthInterceptor` non contiene logica propria: "only hooks into request send/received events" e delega ad `AuthHttpHeaderService` (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L20-L29). Per ogni richiesta:

1. `shouldCatchError` è vero per URL OCC; `shouldAddAuthorizationHeader` per URL OCC **senza** un header `Authorization` già presente (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L175-L183). "URL OCC" significa semplicemente `url.includes(occEndpoints.getBaseUrl())` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L205-L207). `getBaseUrl()` include prefisso e base site attivo (core-libs/core/src/occ/services/occ-endpoints.service.ts:L93-L101): gli URL OCC costruiti con `baseSite: false`, come `baseSites` (core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L100-L100), non risultano OCC per l'interceptor.
2. Se serve l'header, il token arriva da `getStableToken().pipe(take(1))`: emette in modo sincrono, salvo refresh o logout in corso (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L40-L49). `getStableToken()` combina `token$`, `refreshInProgress$` e `logoutInProgress$`, filtra finché entrambi i flag sono `false` e usa `observeOn(queueScheduler)` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L337-L350). In pratica, **durante un refresh tutte le nuove richieste OCC restano in attesa**.
3. `alterRequest` aggiunge `Authorization: <token_type|Bearer> <access_token>` — ma non alle richieste `baseSites` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L188-L203). Se il token passato non ha `access_token`, `createAuthorizationHeader` rilegge in modo sincrono quello corrente; se non c'è restituisce `{}` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L220-L242).

`TokenRevocationInterceptor` (dedicato al server OAuth Hybris) passa la richiesta invariata se `sendAuthHeaderOnRevoke()` è falso; altrimenti aggiunge `Authorization` solo se l'URL è esattamente l'endpoint di revoca (core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L19-L58).

3.8 Gestione errori: 401, refresh e retry

Nel `catchError` dell'`AuthInterceptor` (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L54-L84):

- **401 + token scaduto + URL OCC** → `handleExpiredAccessToken(request, next, token)`. Il token è "scaduto" se `error.errors[0].type` è `InvalidBearerTokenError` o `InvalidTokenError` (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L106-L110).
- **401 dal token endpoint con** `error === 'invalid_token'` → `handleExpiredRefreshToken()` e `EMPTY` (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L65-L72). "Dal token endpoint" significa `errResponse.url` che contiene `getTokenEndpoint()` (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L90-L104).
- **400** `invalid_grant` **con** `grant_type === 'refresh_token'` → `handleExpiredRefreshToken()`, **ma l'errore viene comunque rilanciato** (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.

ts:L74-L83).

Il refresh è un piccolo motore reattivo dentro `AuthHttpHeaderService`:

1. `handleExpiredAccessToken` chiama `getValidToken(initialToken)` e ritenta la richiesta con il nuovo token; se il token è assente restituisce `EMPTY` "per evitare `Authorization: bearer undefined`" (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L247-L260).
2. `getValidToken` si iscrive a `tokenToRetryRequest$`; alla **prima** emissione, e solo se il token corrente è identico a quello della richiesta fallita, emette su `refreshTokenTrigger$` (guard per più chiamate parallele); poi salta i valori finché il token non cambia e prende il primo nuovo (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L356-L380).
3. `tokenToRetryRequest$` usa `using()` per tenere sottoscritto `refreshToken$` finché c'è qualcuno in attesa di `getStableToken()`, condiviso con `shareReplay({ refCount: true, bufferSize: 1 })` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L148-L151).
4. `refreshToken$` ignora i trigger se refresh o logout sono già in corso; se il token ha `refresh_token` chiama `oAuthLibWrapperService.refreshToken()` e imposta `refreshInProgress=true`, altrimenti chiama `handleExpiredRefreshToken()` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L125-L142). `refreshToken()` del wrapper è fire-and-forget (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L173-L181).
5. La libreria salva il nuovo token tramite `AuthStorageService.setItem` (core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L111-L118): la lib scrive `access_token`, `access_token_stored_at`, `expires_at` e `refresh_token` campo per campo via `this._storage.setItem` — deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L2094-L2111, deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L1605-L1609.
6. `stopProgress$` confronta con `pairwise` token precedente e nuovo e, quando l'`access_token` cambia, azzera sia `logoutInProgress` sia `refreshInProgress` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L109-L119). È sottoscritto nel costruttore per tutta la vita del servizio (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L166-L169).
7. `getStableToken()` ora emette il nuovo token: le richieste in coda partono, quella fallita viene ritentata con `createNewRequestWithNewToken` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L382-L392).

3.9 Refresh token scaduto o assente

`handleExpiredRefreshToken()` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L285):

- senza il toggle `enableExpiredRefreshTokenHandlers` va dritto al fallback;
- con il toggle (default `true`, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1072-L1072) esegue in sequenza (`concatMap`) gli `ExpiredRefreshTokenHandler` iniettati da `EXPIRED_REFRESH_TOKEN_HANDLERS` (opzionale, default `[]`, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L154-L155) e si ferma al primo che restituisce `true`; se nessuno gestisce, fallback. Un handler che non restituisce un Observable conta come "non gestito" (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L318-L328).

Il **fallback** salva l'URL della navigazione corrente, esegue `coreLogout()` e poi naviga a `login` mostrando `httpHandlers.sessionExpired` come messaggio d'errore (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L287-L310).

3.10 Logout

`AuthService.logout()` non revoca nulla: naviga alla cxRoute `logout` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L222-L224), dove `LogoutGuard` chiama `coreLogout()` (core-libs/storefront/cms-components/user/logout/logout.guard.ts:L87-L89). `coreLogout()` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L192-L207):

1. `logoutInProgress = true` (blocca le nuove richieste OCC in `getStableToken`);
2. `clearUserId()` → `anonymous`;
3. se `propagateLogoutToAllTabs`, invia `LOGOUT` sul `BroadcastChannel`;
4. `revokeAndLogout()`: `revokeTokenAndLogout(true)` della lib e, se la revoca fallisce, `logOut(true)` locale comunque (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L186-L198);
5. nel `finally`: dispatch `AuthActions.Logout` e `logoutInProgress = false`.

`LogoutEvent` non è legato all'azione NgRx ma alla transizione `true → false` di `isUserLoggedIn()` (core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L57-L63); `LoginEvent` invece è mappato sull'azione `AuthActions.LOGIN` via `StateEventService` (core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L40-L45). Entrambi sono emessi anche per l'emulazione cliente, non per l'autenticazione dell'agente ASM (core-libs/core/src/auth/user-auth/events/user-auth.events.ts:L9-L33). Sugli eventi in generale: capitolo 20 e appendice A07.

Nelle altre tab, `handleAuthNotification` riceve `LOGOUT` e, se non c'è già un logout in corso e l'utente è loggato, esegue a sua volta `coreLogout()` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L300-L314). Il canale si chiama `spartacus_auth_notification`, le notifiche sono isolate per base site e **non** passano per `EventService` (core-libs/core/src/auth/user-auth/facade/auth-notification.service.ts:L11-L28); la classe base apre il `BroadcastChannel` in `listen()` (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L93-L98).

3.11 Redirect dopo il login e guard di accesso

`AuthRedirectService` ricorda "la pagina precedente" oppure "la pagina che l'`AuthGuard` ha appena bloccato" (core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L31-L44):

- senza il toggle `redirectOnlyOnTrueNavigationEnd` (default `false`, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1078-L1078) salva `urlAfterRedirects` a ogni `NavigationEnd`; con il toggle scarta i `NavigationEnd` preceduti da un `NavigationCancel` di tipo `Redirect` (core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L62-L72, core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L122-L140), cioè da un guard che ha restituito una `UrlTree` (angular:packages/router/src/events.ts:L179-L183);
- `setRedirectUrl` ignora gli URL delle route `authFlow` e rimuove i parametri di site context (core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L112-L120); le route `authFlow` sono calcolate lazily dalla `RoutingConfig` (core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts:L23-L46, proprietà definita in core-libs/core/src/routing/configurable-routes/routes-config.ts:L49-L56);
- `saveCurrentNavigationUrl()` usa `router.currentNavigation()?.finalUrl` e non fa nulla se non c'è una navigazione in corso (core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L99-L107);
- `redirect()` naviga all'URL salvato o a `'/'` e azzera l'URL (core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L81-L93).

`AuthGuard`, se l'utente non è loggato, salva l'URL della navigazione corrente e restituisce la `UrlTree` di `login` (core-libs/core/src/auth/user-auth/guards/auth.guard.ts:L30-L42). `NotAuthGuard`, se l'utente è loggato, re-

stituisce la `UrlTree` di `home` (core-libs/core/src/auth/user-auth/guards/not-auth.guard.ts:L28-L40). Catalogo dei guard: appendice A05; route semantiche e protected routes: capitolo 14.

3.12 Client token (`client_credentials`)

1. Un adapter OCC marca la richiesta con l'header `cx-use-client-token` (`USE_CLIENT_TOKEN`, core-libs/core/src/occ/utils/interceptor-util.ts:L9-L9).
2. `ClientTokenInterceptor` rimuove **sempre** il marker, anche con i client token disabilitati; se `useClientTokens` è falso passa oltre (core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L46-L52). Il flag è letto una volta sola come field initializer (core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L32-L34).
3. Per richieste marcate chiede il token a `ClientTokenService.getClientToken()`, che dispatcha `LoadClientToken` se lo store non è né caricato né in loading (core-libs/core/src/auth/client-auth/services/client-token.service.ts:L31-L47). Per le altre restituisce `of(undefined)` (core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L90-L97).
4. `ClientTokenEffect` usa `exhaustMap` e chiama `ClientAuthenticationTokenService`, che fa POST al token endpoint con `client_id`, `client_secret`, `grant_type=client_credentials` (core-libs/core/src/auth/client-auth/store/effects/client-token.effect.ts:L21-L42, core-libs/core/src/auth/client-auth/services/client-authentication-token.service.ts:L31-L48). Lo stato è un `loaderReducer` nella feature `client-auth` (core-libs/core/src/auth/client-auth/store/reducers/index.ts:L13-L25, core-libs/core/src/auth/client-auth/store/client-auth-state.ts:L10-L19).
5. Il client token è aggiunto come `Authorization` se presente e se l'URL è OCC (core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L57-L68).
6. Su 401 scaduto di una richiesta marcata, `ClientErrorHandlingService.handleExpiredClientToken` forza `refreshClientToken()` e ritenta (core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L71-L85, core-libs/core/src/auth/client-auth/services/client-error-handling.service.ts:L30-L40). Essendo l'interceptor più interno, il 401 non arriva all'`AuthInterceptor` se viene gestito qui.

3.13 Federated login

Il federated login permette di eseguire il login su un **login domain** diverso dal dominio dello storefront ("origin"). Config di default: `enabled: false`, `contextParameterName: 'ctx'`, `loginHosts: []`, `originMap: {}` (core-libs/core/src/federated-login/config/default-federated-login-config.ts:L9-L18).

1. `FederatedLoginService.enabled` e `isLoginDomain` sono calcolati una volta alla creazione; `isLoginDomain` confronta `loginHosts` con l'host di `location.origin` (core-libs/core/src/federated-login/services/federated-login.service.ts:L24-L27, core-libs/core/src/federated-login/services/federated-login.service.ts:L81-L89).
2. Sull'origin, `getParameters()` produce `ctx=<serializzato>` con lingua attiva e origin (core-libs/core/src/federated-login/services/federated-login.service.ts:L40-L60); `serializeContext` è un Base64 URL-safe di `<originKey>:<language>`, stringa vuota se l'origin non è in `originMap` (core-libs/core/src/federated-login/services/federated-login-context-serializer.service.ts:L16-L31). Il wrapper lo appende al `loginUrl` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L126-L139).
3. Sul login domain, `detectContext()` fonde il contesto salvato e quello in URL e lo riscrive in `localStorage` chiave `federatedLogin` (core-libs/core/src/federated-login/services/federated-login.service.ts:L62-L79, core-libs/core/src/federated-login/services/federated-login-context-storage.ts:L17-

L23); `validateContext` azzera `origin` se non è fra i valori di `originMap` (core-libs/core/src/federated-login/services/federated-login.service.ts:L92-L105).

4. `FederatedLoginGuard` è registrato come `BEFORE_CMS_PAGE_GUARD` (core-libs/storefront/cms-structure/routing/routing.module.ts:L27-L36): sul login domain senza `origin` manda a `notFound` evitando loop, su route non `authFlow` rimanda all'origin via `location.href` e ritorna `false` (core-libs/core/src/auth/user-auth/guards/federated-login.guard.ts:L43-L57, core-libs/core/src/auth/user-auth/guards/federated-login.guard.ts:L74-L104).
5. Il `redirectUri` sul login domain è l'origin memorizzato (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L76-L94).

---

4. Diagrammi

4.1 Bootstrap e ritorno dall'Authorization Server (Authorization Code)

■ **TAVOLA 22.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/routing/routing.module.ts:L66-L71, core-libs/core/src/auth/user-auth/user-auth.module.ts:L39-L80, core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L77-L104, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L90-L117, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L316-L327, core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L258-L284.

4.2 Catena degli interceptor su una richiesta OCC

■ **TAVOLA 22.2** diagramma a tutta pagina, in fondo al capitolo

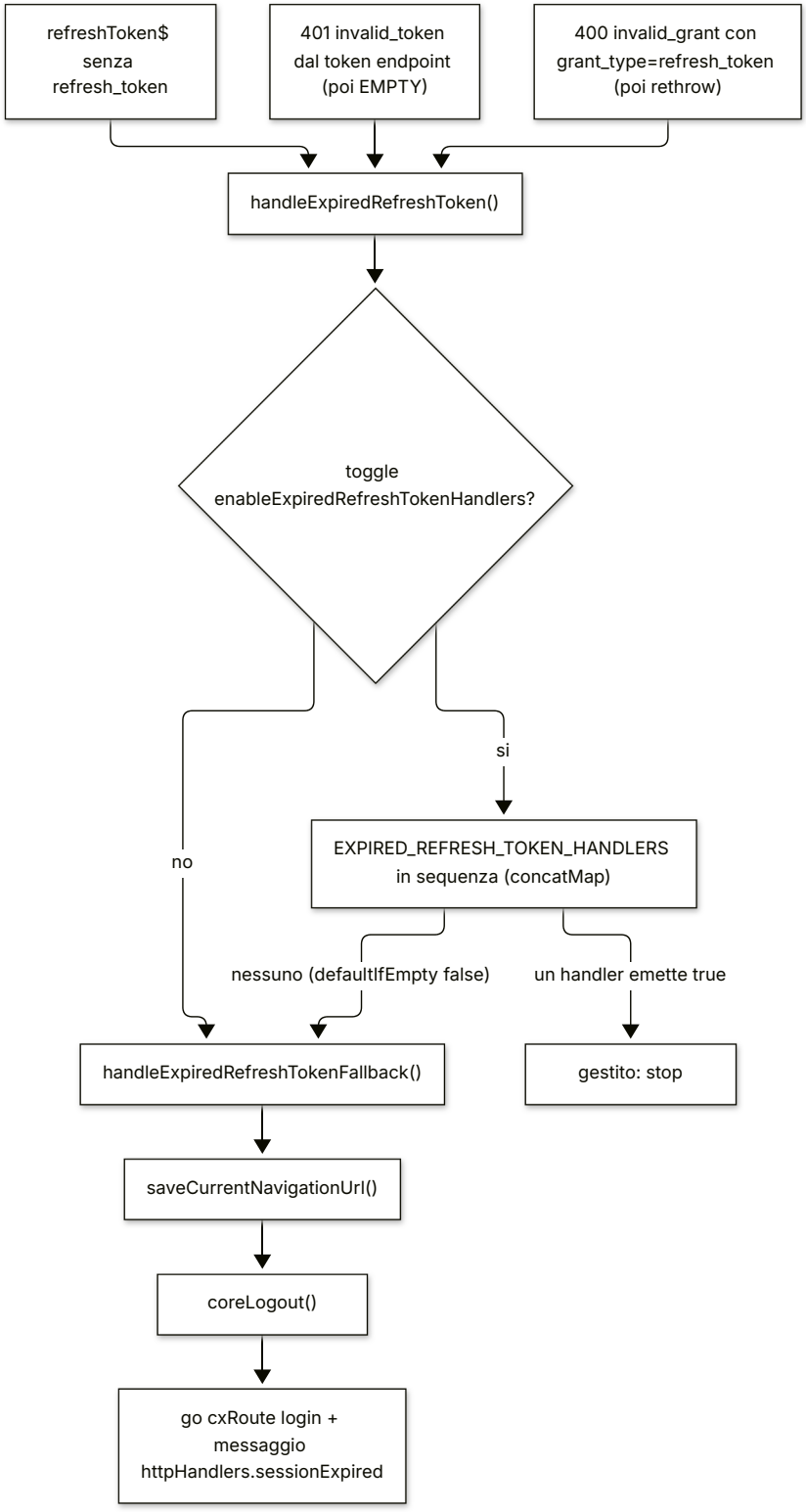
Riferimenti: angular:packages/common/http/src/interceptor.ts:L233-L240, core-libs/core/src/auth/auth.module.ts:L12-L24, core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L31-L49, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L175-L203, core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L29-L58, core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L42-L97.

4.3 401 per access token scaduto: refresh e retry

■ **TAVOLA 22.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L56-L64, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L109-L151, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L247-L260, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L337-L392, core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L173-L181.

4.4 Refresh token inutilizzabile: handler e fallback



Riferimenti: core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L125-L142, core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L65-L83, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L328.

5. Estratti di codice reale

5.1 L'ordine dei moduli è semantico

core-libs/core/src/auth/auth.module.ts:L12-L17

```ts
@NgModule({
  // ClientAuthModule should always be imported after UserAuthModule because the ClientTokenInterceptor must be imported after the AuthInterceptor.
  // This way, the ClientTokenInterceptor is the first to handle 401 errors and attempt to refresh the client token.
  // If the request is not for the client token, the AuthInterceptor handles the refresh.
  imports: [CommonModule, UserAuthModule.forRoot(), ClientAuthModule.forRoot()],
})
```

5.2 Location initializer: prima la sync dello storage, poi il controllo dell'URL

core-libs/core/src/auth/user-auth/user-auth.module.ts:L70-L80

```ts
const authInitializedFactory: LocationInitializerFactory = () => {
  const authService = inject(AuthService);
  const configInit = inject(ConfigInitializerService);
  const platformId = inject(PLATFORM_ID);
  const authStatePersistenceService = inject(AuthStatePersistenceService);

  //We need to call it before checkOAuthParamsInUrl to ensure that in-memory storage is in sync with browser storage.
  //In other case "nonce" value used in Implicit Flow will be undefined causing the error.
  authStatePersistenceFactory(authStatePersistenceService)();
  return checkOAuthParamsInUrl(authService, configInit, platformId);
};
```

5.3 Il `refresh_token` non finisce nel `localStorage`

core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L76-L84

```ts
      map(([authToken, userId, redirectUrl]) => {
        let token = authToken;
        if (token) {
          token = { ...token };
          // To minimize risk of user account hijacking we don't persist user refresh_token
          delete token.refresh_token;
        }
        return { token, userId, redirectUrl };
      })
```

5.4 Gestione degli errori in `AuthInterceptor`

core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L54-L84

```ts
          catchError((errResponse: any) => {
            switch (errResponse.status) {
              case 401: // Unauthorized
                if (this.isExpiredToken(errResponse) && shouldCatchError) {
                  // request failed because of the expired access token
                  // we should get refresh the token and retry the request, or logout if the refresh is
missing / expired
                  return this.authHttpHeaderService.handleExpiredAccessToken(
                    request,
                    next,
                    token
                  );
                } else if (
                  // Refresh the expired token
                  // Check if the OAuth endpoint was called and the error is because the refresh token
expired
                  this.errorIsInvalidToken(errResponse)
                ) {
                  this.authHttpHeaderService.handleExpiredRefreshToken();
                  return EMPTY;
                }
                break;
              case 400: // Bad Request
                if (
                  this.errorIsInvalidGrant(errResponse) &&
                  request.body.get('grant_type') === 'refresh_token'
                ) {
                  this.authHttpHeaderService.handleExpiredRefreshToken();
                }
                break;
            }
            throw errResponse;
          })
```

5.5 Il cuore del refresh: trigger, stop e retry

core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L125-L151

```ts
  protected refreshToken$ = this.refreshTokenTrigger$.pipe(
    withLatestFrom(
      this.authService.refreshInProgress$,
      this.authService.logoutInProgress$
    ),
    filter(
      ([, refreshInProgress, logoutInProgress]) =>
```

```
      !refreshInProgress && !logoutInProgress
  ),
  tap(([token]) => {
    if (token?.refresh_token) {
      this.oAuthLibWrapperService.refreshToken();
      this.authService.setRefreshProgress(true);
    } else {
      this.handleExpiredRefreshToken();
    }
  })
);

/**
 * Kicks of the process by listening to the new token and refresh token processes.
 * This token should be used when retrying the failed http request.
 */
protected tokenToRetryRequest$ = using(
  () => this.refreshToken$.subscribe(),
  () => this.getStableToken()
).pipe(shareReplay({ refCount: true, bufferSize: 1 }));
```

core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L356-L380

```ts
protected getValidToken(
  requestToken: AuthToken | undefined
): Observable<AuthToken | undefined> {
  return defer(() => {
    // flag to only refresh token only on first emission
    let refreshTriggered = false;
    return this.tokenToRetryRequest$.pipe(
      tap((token) => {
        // we want to refresh the access token only when it is old.
        // this is a guard for the case when there are multiple parallel http calls
        if (
          token?.access_token === requestToken?.access_token &&
          !refreshTriggered
        ) {
          this.refreshTokenTrigger$.next(token as AuthToken);
        }
        refreshTriggered = true;
      }),
      skipWhile(
        (token) => token?.access_token === requestToken?.access_token
      ),
      take(1)
    );
  });
}
```

5.6 Handler per il refresh token scaduto

core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts:L17-L23

```ts
export interface ExpiredRefreshTokenHandler {
  handleExpiredRefreshTokenIfApplicable?(): Observable<boolean>;
}

export const EXPIRED_REFRESH_TOKEN_HANDLERS = new InjectionToken<
  ExpiredRefreshTokenHandler[]
>('EXPIRED_REFRESH_TOKEN_HANDLERS');
```

5.7 `coreLogout()`

core-libs/core/src/auth/user-auth/facade/auth.service.ts:L192-L207

```ts
  coreLogout(): Promise<void> {
    this.setLogoutProgress(true);
    this.userIdService.clearUserId();
    if (this.featureToggles.propagateLogoutToAllTabs) {
      this.authNotificationService.sendNotification(
        AuthNotificationType.LOGOUT
      );
    }
    return new Promise((resolve) => {
      this.oAuthLibWrapperService.revokeAndLogout().finally(() => {
        this.store.dispatch(new AuthActions.Logout());
        this.setLogoutProgress(false);
        resolve();
      });
    });
  }
```

5.8 Default legacy (senza code flow)

core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L117-L138

```ts
  } else {
    const config = {
      authentication: {
        ...defaultAuthConfig.authentication,
        client_id: 'mobile_android',
        client_secret: 'secret',
        sendAuthHeaderOnRevoke: true,
        useClientTokens: true,
        OAuthLibConfig: {
          ...defaultAuthConfig.authentication?.OAuthLibConfig,
          disablePKCE: true,
```

```
      },
    },
  } satisfies AuthConfig;

  delete config.authentication.OAuthLibConfig.responseType;
  delete config.authentication.customLoginPage;
  delete config.authentication.initializerOptions;
  delete config.authentication.OAuthLibConfig?.redirectUri;

  return config;
}
```

5.9 `ClientTokenInterceptor`: marker sempre rimosso

core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L42-L52

```ts
intercept(
  request: HttpRequest<any>,
  next: HttpHandler
): Observable<HttpEvent<any>> {
  const isClientTokenRequest = this.isClientTokenRequest(request);
  if (isClientTokenRequest) {
    request = InterceptorUtil.removeHeader(USE_CLIENT_TOKEN, request);
  }
  if (!this.enableClientToken) {
    return next.handle(request);
  }
```

5.10 `CustomLoginGuard`: CSRF e un solo retry

core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L70-L87

```ts
  return this.authService.getCsrfToken().pipe(
    tap((token) => {
      this.csrfStateService.set(token);
      this.clearRedirectCount();
    }),
    map(() => true),
    catchError(() => {
      const currentCount = this.getRedirectCount();
      // check if retry limit is met
      if (currentCount >= totalRetries) {
        // go to homepage with message
        this.clearRedirectCount();
        this.showErrorMessage({ key: 'authMessages.unrecoverableError' });
        return this.createRoute('home');
      }
```

```
        // retry until limit met
        this.setRedirectCount(currentCount + 1);
```

6. Configurazione

Tutte le chiavi `authentication.*` sono definite nella classe astratta `AuthConfig` (core-libs/core/src/auth/user-auth/config/auth-config.ts:L32-L143), fusa in `Config` via module augmentation (core-libs/core/src/auth/user-auth/config/auth-config.ts:L145-L147). Il catalogo generale è in appendice A02.

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`authentication.client_id`	string	`mobile_android_public` (code flow) / `mobile_android` (legacy)	core-libs/core/src/auth/user-auth/config/auth-config.ts:L38-L41; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L50-L50, core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L121-L121	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L38-L40; core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L55-L59
`authentication.client_secret`	string	assente / `secret` (legacy)	core-libs/core/src/auth/user-auth/config/auth-config.ts:L42-L45; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L122-L122	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L47-L49
`authentication.baseUrl`	string	assente → `backend.occ.baseUrl + '/authorizationserver'`	core-libs/core/src/auth/user-auth/config/auth-config.ts:L46-L49	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L54-L59; core-libs/core/src/auth/user-auth/config/base-url-config-validator.ts:L10-L19
`authentication.tokenEndpoint`	string	`/oauth/token`	core-libs/core/src/auth/user-auth/config/auth-config.ts:L50-L53; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L51-L51	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L64-L67
`authentication.revokeEndpoint`	string	`/oauth/revoke`	core-libs/core/src/auth/user-auth/config/auth-config.ts:L54-L57; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L52-L52	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L97-L100

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
authentication.sendAuthHeaderOnRevoke	boolean	assente / true (legacy)	core-libs/core/src/auth/user-auth/config/auth-config.ts:L59-L66; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L123-L123	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L112-L114 → core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L33-L36
authentication.useClientTokens	boolean	assente (→ false) / true (legacy)	core-libs/core/src/auth/user-auth/config/auth-config.ts:L67-L76; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L124-L124	core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L33-L34
authentication.loginUrl	string	/oauth/authorize	core-libs/core/src/auth/user-auth/config/auth-config.ts:L77-L80; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L53-L53	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L72-L75
authentication.logoutUrl	string	assente	core-libs/core/src/auth/user-auth/config/auth-config.ts:L81-L84	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L119-L122
authentication.userinfoEndpoint	string	assente	core-libs/core/src/auth/user-auth/config/auth-config.ts:L85-L88	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L127-L130
authentication.OAuthLibConfig	AuthLibConfig	vedi §3.2	core-libs/core/src/auth/user-auth/config/auth-config.ts:L89-L92; core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L54-L64	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L135-L137 → core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L96-L115
authentication.OAuthLibConfig.responseType	string	code / rimosso (legacy)	core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L62-L62, core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L132-L132	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L151-L164

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`authentication.OAuthLibConfig.disablePKCE`	boolean	`false` / `true` (legacy)	`core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L59-L59`, `core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L127-L127`	lib angular-oauth2-oidc: `deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L1996-L2006` (login URL, `code_challenge` e `PKCE_verifier`), `deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L2231-L2245` (`code_verifier` nello scambio del code)
`authentication.OAuthLibConfig.redirectUri`	string	`/oauth-callback` solo con i tre toggle, altrimenti rimosso	`core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L63-L63`	`core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L79-L114`; `core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L80-L91`
`authentication.OAuthLibConfig.requireHttps`	boolean	assente	(campo della lib)	`core-libs/core/src/auth/user-auth/config/base-url-config-validator.ts:L15-L15`
`authentication.customLoginPage.csrfEndpoint`	string	`/csrf` (rimosso in legacy)	`core-libs/core/src/auth/user-auth/config/auth-config.ts:L96-L105`; `core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L65-L68`	`core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L89-L92`
`authentication.customLoginPage.loginFormEndpoint`	string	`/login` (rimosso in legacy)	idem	`core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L80-L84`
`authentication.initializerOptions.addBaseSiteToRedirectUri`	`boolean | 'auto'`	`'auto'` (rimosso senza async initializer)	`core-libs/core/src/auth/user-auth/config/auth-config.ts:L107-L123`; `core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L69-L72`	`core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L132-L140`
`authentication.initializerOptions.baseSiteSuffix`	`boolean | 'auto'`	`'auto'` (idem)	`core-libs/core/src/auth/user-auth/config/auth-config.ts:L125-L140`	`core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L119-L127`
`federatedLogin.enabled`	boolean	`false`	`core-libs/core/src/federated-login/config/federated-login-config.ts:L15-L16`; `core-libs/core/src/federated-login/config/default-federated-login-config.ts:L12-L12`	`core-libs/core/src/federated-login/services/federated-login.service.ts:L24-L24`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
federatedLogin.contextParameterName	string	ctx	core-libs/core/src/federated-login/config/federated-login-config.ts:L18-L19	core-libs/core/src/federated-login/services/federated-login.service.ts:L62-L79
federatedLogin.loginHosts	string[]	[]	core-libs/core/src/federated-login/config/federated-login-config.ts:L21-L31	core-libs/core/src/federated-login/services/federated-login.service.ts:L81-L89
federatedLogin.originMap	Record<string, string>	{}	core-libs/core/src/federated-login/config/federated-login-config.ts:L33-L45	core-libs/core/src/federated-login/services/federated-login-context-serializer.service.ts:L61-L71; core-libs/core/src/federated-login/services/federated-login.service.ts:L92-L105
toggle authorizationCodeFlowByDefault	boolean	true	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L161-L161; default core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029	core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L76-L83; core-libs/core/src/auth/user-auth/facade/auth.service.ts:L316-L320; core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L219-L225
toggle asyncAuthConfigInitializer	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L637-L637; default core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1090-L1090	core-libs/core/src/auth/user-auth/user-auth.module.ts:L108-L113; core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L55-L64
toggle oauthCallbackPage	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L914-L914; default core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1111-L1111	core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L85-L85; core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L160-L166

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `propagateLogoutToAllTabs`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L610-L610`; default `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1088-L1088`	`core-libs/core/src/auth/user-auth/user-auth.module.ts:L85-L92`; `core-libs/core/src/auth/user-auth/facade/auth.service.ts:L195-L199`
toggle `enableExpiredRefreshTokenHandlers`	boolean	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L464-L464`; default `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1072-L1072`	`core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L269`
toggle `redirectOnlyOnTrueNavigationEnd`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L509-L509`; default `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1078-L1078`	`core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L62-L72`
storage `auth` (localStorage)	—	—	`core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L44-L44`	`core-libs/core/src/state/services/state-persistence.service.ts:L44-L48`
storage `oAuthRedirectCodeFlow`	—	—	`core-libs/core/src/auth/user-auth/utils/oauth-constants.ts:L7-L7`	`core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L224-L224`, `core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L278-L286`
storage `login_redirect_count`	—	—	`core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L29-L29`	`core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L106-L123`
storage `federatedLogin`	—	—	`core-libs/core/src/federated-login/services/federated-login-context-storage.ts:L17-L17`	`core-libs/core/src/federated-login/services/federated-login-context-storage.ts:L26-L40`
storage `asm_enabled`	—	—	`core-libs/core/src/auth/user-auth/facade/auth.service.ts:L265-L265`	`core-libs/core/src/auth/user-auth/facade/auth.service.ts:L289-L295`

Due regole di compatibilità con il backend vengono dal JSDoc: `sendAuthHeaderOnRevoke` e `useClientTokens` vanno messi a `true` con l'Authorization Server legacy (precedente a settembre 2025) e a `false` con quello ricostruito (core-libs/core/src/auth/user-auth/config/auth-config.ts:L59-L76).

Il token `USE_AUTHORIZATION_CODE_FLOW_BY_DEFAULT` è deprecato dalla 221121.1 a favore del toggle (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L11-L19); vedi appendice A10.

---

7. Commenti e note notevoli dal sorgente

- `// TODO: Remove platformId dependency in 4.0` — TODO mai risolto, sopravvissuto a molte major (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L45-L45).
- `// TODO(#9638): Use logout route when it will support passing redirect url` — il fallback del refresh scaduto usa `coreLogout()` direttamente perché la route `logout` non sa conservare la redirect URL (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L299-L299).
- Token temporanei "to preserve legacy behavior of CXPSA-12514 fix", da rimuovere a rollout completato di `enableExpiredRefreshTokenHandlers` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L51-L54).
- `// When removing this feature toggle, switch to provideConfigInitializer` (core-libs/core/src/auth/user-auth/user-auth.module.ts:L109-L109).
- Il commento che spiega perché serve controllare anche l'`access_token`: nel code flow `result` è `true` anche senza login riuscito, con link al sorgente di angular-oauth2-oidc (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L101-L103).
- `// We don't load discovery document, because it doesn't contain revoke endpoint information` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L273-L273).
- `// To minimize risk of user account hijacking we don't persist user refresh_token` (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L80-L80).
- Il commento in testa ad `auth-config.ts` spiega perché alcuni campi della lib sono omessi: `siletRefreshTimeout` deprecato (con typo) e endpoint sotto il controllo di Spartacus (core-libs/core/src/auth/user-auth/config/auth-config.ts:L11-L19).
- JSDoc di `ClientAuthModule`: il pattern del client token "should not be used in the frontend apps" e il modulo "can be safely removed" dopo i miglioramenti OCC (core-libs/core/src/auth/client-auth/client-auth.module.ts:L12-L18).
- JSDoc di `getClientSecret`: "OAuth server shouldn't require it from web apps (but Hybris OAuth server requires)" (core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L42-L46).
- JSDoc di `coreLogout`: "To perform logout it is best to use `logout` method. Use this method with caution." (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L188-L191).
- JSDoc di `AuthNotificationService`: non fa da ponte verso `EventService`; ogni tab esegue il proprio `coreLogout()` (core-libs/core/src/auth/user-auth/facade/auth-notification.service.ts:L20-L25).
- JSDoc di `AuthRedirectService.init`: coppie `[NavigationEnd, NavigationEnd]` salvate, `[NavigationCancel, NavigationEnd]` scartate per non far sovrascrivere l'URL dal redirect post-logout verso `/` (core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L55-L61).
- JSDoc del toggle `oauthCallbackPage`, che descrive cosa serve lato CMS (`OauthCallbackComponent` con `SpinnerComponent`) (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L902-L914).

8. Extension point

EXTENSION POINT	COME	DOVE	RICETTA
`AuthConfig`	`provideConfig({ authentication: {...} });` augmentation di `Config`	`core-libs/core/src/auth/user-auth/config/auth-config.ts:L32-L147`	ricetta 50 (override di configurazione)
`AuthConfigInitializer`	sottoclasse con override dei metodi `protected` (`generateClientId`, `generateRedirectUri`…)	`core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L19-L167`	ricetta 50
`AuthService`, `AuthStorageService` (e `AuthHttpHeaderService`)	sostituzione via DI (`provide: AuthService, useExisting: ...`): ASM li sostituisce con `AsmAuthService`, `AsmAuthStorageService` e `AsmAuthHttpHeaderService`; `UserIdService` è sostituibile allo stesso modo, ma ASM non lo fa	`feature-libs/asm/root/asm-root.module.ts:L65-L76`	ricetta 50 (override di service/facade via DI)
`AuthHttpHeaderService`	sottoclasse via DI; attenzione ai conflitti fra librerie (ASM, Punchout) e ai token `DEFAULT_AUTH_HTTP_HEADER_SERVICE` / `DELEGATED_AUTH_HTTP_HEADER_SERVICE`	`core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L51-L82`; `integration-libs/punchout/root/punchout.root.module.ts:L51-L71`	ricetta 50
`EXPIRED_REFRESH_TOKEN_HANDLERS`	provider multi con un `ExpiredRefreshTokenHandler`: alternativa *non invasiva* all'override del servizio	`core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts:L17-L23`	ricetta 50 (interceptor e DI)
`AuthInterceptor`, `TokenRevocationInterceptor`	registrati `useExisting`: sostituibili con `{ provide: AuthInterceptor, useClass: ... }`	`core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L12-L23`	ricetta 50 (interceptor custom)
Guard (`AuthGuard`, `NotAuthGuard`, `CustomLoginGuard`, `FederatedLoginGuard`, `OAuthCallbackGuard`)	uso nei `guards` dei componenti CMS o override via DI (metodi `protected`)	`core-libs/core/src/auth/user-auth/guards/index.ts:L7-L11`	ricetta 51 (route custom e guard)
`RouteConfig.authFlow`	marcare route custom come parte del percorso di login (esclusa dal redirect, ammessa sul login domain)	`core-libs/core/src/routing/configurable-routes/routes-config.ts:L49-L56`	ricetta 51
`LoginEvent`, `LogoutEvent`	`EventService.get(LoginEvent)` per reagire a login/logout	`core-libs/core/src/auth/user-auth/events/user-auth.events.ts:L9-L33`	ricetta 50 (eventi)
`USE_CLIENT_TOKEN` + `InterceptorUtil`	marcare richieste di un adapter custom per usare il client token	`core-libs/core/src/occ/utils/interceptor-util.ts:L9-L9`	ricetta 52 (adapter custom)

EXTENSION POINT	COME	DOVE	RICETTA
FederatedLoginConfig	provideConfig({ federatedLogin: {...} })	core-libs/core/src/federated-login/config/federated-login-config.ts:L10-L47	ricetta 50

Esempio minimo di handler (solo simboli esportati: `ExpiredRefreshTokenHandler` ed `EXPIRED_REFRESH_TOKEN_HANDLERS` sono nel barrel dei servizi `user-auth`, riesportato da `@spartacus/core` — core-libs/core/src/auth/user-auth/services/index.ts:L9-L10, core-libs/core/src/auth/user-auth/index.ts:L20-L20, core-libs/core/src/auth/index.ts:L7-L9, core-libs/core/public_api.ts:L11-L11). È uno schizzo illustrativo, non codice del repository:

```ts
// esempio illustrativo
@Injectable({ providedIn: 'root' })
export class MyExpiredRefreshTokenHandler implements ExpiredRefreshTokenHandler {
  handleExpiredRefreshTokenIfApplicable(): Observable<boolean> {
    // true = gestito: il fallback (logout + redirect a login) NON viene eseguito
    return of(false);
  }
}

// providers:
{ provide: EXPIRED_REFRESH_TOKEN_HANDLERS, useExisting: MyExpiredRefreshTokenHandler, multi: true }
```

Funziona solo con il toggle `enableExpiredRefreshTokenHandlers` attivo (default `true`) — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L285. Punchout usa esattamente questo meccanismo (integration-libs/punchout/root/punchout.root.module.ts:L51-L71); il capitolo 36 e il capitolo 29 su ASM mostrano gli override reali.

Da evitare (vedi capitolo 54): importare simboli non esportati dai barrel. Non sono pubblici, per esempio, l'array `interceptors` e le factory del modulo (core-libs/core/src/auth/user-auth/index.ts:L13-L22), `provideAuthorizationCodeFlowByDefault` (core-libs/core/src/auth/user-auth/index.ts:L9-L9), effects e reducers del client auth (core-libs/core/src/auth/client-auth/index.ts:L7-L13), la costante `totalRetries` (core-libs/core/src/auth/user-auth/guards/index.ts:L8-L8).

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **L'ordine** `UserAuthModule` → `ClientAuthModule` **è semantico.** Invertirlo farebbe gestire i 401 dei client token all'`AuthInterceptor` — core-libs/core/src/auth/auth.module.ts:L13-L16, angular:packages/common/http/src/interceptor.ts:L233-L240.
2. **Il login fallito non lancia.** `checkOAuthParamsInUrl`, `loginWithCredentials`, `otpLoginWithCredentials` inghiottono ogni errore con `catch {}`; il chiamante deve osservare `isUserLoggedIn()` — core-libs/core/src/auth/user-auth/facade/auth.service.ts:L116-L116, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L152-L152, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L185-L185.
3. **Dopo un reload non c'è refresh token.** Il `refresh_token` non è persistito; al primo 401 dopo un reload `refreshToken$` non trova `refresh_token` e chiama `handleExpiredRefreshToken()` → logout forzato e redi-

rect a login — core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L80-L81, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L134-L140.

4. `UserIdService` **può non emettere mai.** È un `ReplaySubject(1)`: senza `setUserId`/`clearUserId` (normalmente chiamati da `onRead`) `getUserId()` resta muto — core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L25-L25, core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L99-L103.
5. **"URL OCC" è un** `includes`**.** `isOccUrl` controlla `url.includes(getBaseUrl())` — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L205-L207. `getBaseUrl() = urlPathJoin(backend.occ.baseUrl ?? '', prefix, activeBaseSite)`: con baseUrl non configurato restano prefisso (default `/occ/v2/`) e base site, cioè `/occ/v2/<baseSite>` — core-libs/core/src/occ/services/occ-endpoints.service.ts:L93-L101, core-libs/core/src/occ/config/default-occ-config.ts:L12-L12. Il caso "ogni URL è OCC" richiede baseUrl, prefisso e base site tutti vuoti (capitolo 21).
6. **401 e 400 si comportano diversamente.** Su 401 `invalid_token` l'interceptor restituisce `EMPTY` (stream completato senza valore né errore); su 400 `invalid_grant` chiama lo stesso handler ma rilancia l'errore — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L65-L83.
7. `request.body.get(...)` **presuppone** `HttpParams`**.** Un 400 `invalid_grant` dal token endpoint con un body di altro tipo potrebbe lanciare un `TypeError` — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L76-L77.
8. **Retry che "sparisce".** Se dopo il refresh il token è `undefined` (logout in corso), `handleExpiredAccessToken` restituisce `EMPTY`: il componente che aveva fatto la chiamata non riceve né valore né errore — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L253-L258.
9. **Richieste in coda durante refresh/logout.** `getStableToken()` filtra finché uno dei due flag è vero: le richieste OCC che richiedono l'header aspettano — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L337-L350. I flag tornano `false` solo quando l'`access_token` cambia (`stopProgress$`) o nel `finally` di `coreLogout()` — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L109-L119, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L201-L205.
10. **Revoca con uguaglianza esatta.** `TokenRevocationInterceptor` confronta l'URL con `===` all'endpoint di revoca prefissato: query string o slash diversi disattivano l'header — core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L56-L58.
11. `useClientTokens` **letto una sola volta.** È un field initializer di `ClientTokenInterceptor`: un valore impostato da un config initializer asincrono successivo all'istanziazione non ha effetto — core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L33-L34.
12. **Store client token in errore = "caricato".** `isClientTokenLoaded` considera `success || error`; `getClientToken()` può emettere `undefined` e la richiesta parte senza header — core-libs/core/src/auth/client-auth/services/client-token.service.ts:L66-L68, core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L57-L68.
13. **Federated login statico.** `enabled` e `isLoginDomain` sono calcolati alla creazione del servizio — core-libs/core/src/federated-login/services/federated-login.service.ts:L24-L27. Inoltre se l'origin non è in `originMap` il parametro `ctx` è vuoto — core-libs/core/src/federated-login/services/federated-login-context-serializer.service.ts:L19-L22.
14. `saveCurrentNavigationUrl()` **è no-op fuori navigazione.** Se l'errore nasce da un'azione UI su pagina stabile, resta l'ultimo URL salvato dal listener `NavigationEnd` — core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L99-L103.
15. **Redirect sovrascritto dai guard.** Con il default `redirectOnlyOnTrueNavigationEnd: false`, i redirect dei guard (es. post-logout verso `/`) possono sovrascrivere l'URL salvato; è il problema che il toggle risolve — core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L506-L509, core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L55-L61.

16. **ASM "appiccicoso".** `refreshAuthConfig()` attiva `asm_client` anche solo per `asm_enabled` rimasto in `localStorage` da una sessione precedente — core-libs/core/src/auth/user-auth/facade/auth.service.ts:L263-L267, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L289-L291. E con client ASM o utente emulato, `checkOAuthParamsInUrl` non imposta `current` né fa redirect — core-libs/core/src/auth/user-auth/facade/auth.service.ts:L104-L104.
17. **Un solo retry CSRF in 15 s.** Oltre, `CustomLoginGuard` porta a home con `authMessages.unrecoverableError` — core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L29-L31, core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L79-L84.
18. **In legacy la custom login page sparisce.** `defaultAuthConfigFactory` elimina `customLoginPage`, quindi `customLoginEnabled()` è falso e `CustomLoginGuard` diventa no-op — core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L133-L133, core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L105-L107.
19. `provideAuthorizationCodeFlowByDefault` **usa** `inject()`**.** Va chiamata in un injection context, ed è comunque non esportata dal barrel — core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L31-L46, core-libs/core/src/auth/user-auth/index.ts:L9-L9.
20. `LogoutEvent` **non coincide con** `AuthActions.Logout`**.** L'evento nasce dalla transizione di `isUserLoggedIn()` (quando la lib rimuove il token), l'azione nel `finally` di `coreLogout()` — core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L57-L63, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L201-L205. `LogoutEvent` precede sempre `AuthActions.Logout`: la lib rimuove `access_token` in modo sincrono dentro `logOut()`, chiamato prima di `resolve()` (deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L3013-L3015, deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L2724-L2731); nello stesso istante `stopProgress$` riporta `logoutInProgress` a `false`, cioè prima del `finally` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L109-L119). Se la revoca fallisce, il `catch` del wrapper chiama `logOut(true)` prima del `finally` (core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L187-L191). Senza `access_token` la lib risolve subito, senza `logOut` e quindi senza `LogoutEvent` (deps:angular-oauth2-oidc-20.0.2/package/fesm2022/angular-oauth2-oidc.mjs:L2954-L2959).

---

10. Come fare debug

- **Il token c'è?** In DevTools → Application → Local Storage, chiave `auth`: contiene `token` (senza `refresh_token`), `userId`, `redirectUrl` (core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L19-L57). Ricorda che è solo la copia persistita: la fonte di verità è il `BehaviorSubject` di `AuthStorageService` (core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L18-L37).
- **Quale flow è attivo?** Inietta `AuthConfigService` e chiama `getOAuthFlow()` (core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L151-L164); verifica i toggle `authorizationCodeFlowByDefault`, `asyncAuthConfigInitializer`, `oauthCallbackPage` tramite la tabella di §3.2.
- **Endpoint sbagliati / warning in console.** Controlla `getBaseUrl()` e il warning di `baseUrlConfigValidator` (core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L54-L59, core-libs/core/src/auth/user-auth/config/base-url-config-validator.ts:L10-L19).
- **Header** `Authorization` **mancante.** Nel tab Network verifica che l'URL contenga il base URL OCC (`isOccUrl`) e non sia una richiesta `baseSites` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L188-L213). Se la richiesta non parte proprio, probabilmente `refreshInProgress$` o `logoutInProgress$` sono rimasti `true`: sottoscrivi entrambi da `AuthService` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L52-L62).

- **Logout inatteso con messaggio "session expired".** Metti un breakpoint in `handleExpiredRefreshTokenFallback` (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L287-L310) e risali lo stack: refresh token assente (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L138-L140, tipico dopo un reload), 401 `invalid_token` o 400 `invalid_grant` dal token endpoint (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L65-L83).
- **Il ritorno dall'Authorization Server non logga.** Breakpoint in `AuthService.checkOAuthParamsInUrl` sul `catch {}` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L90-L117); controlla `result`, `tokenReceived`, `isEmulated`, `isUsingASMClient`; verifica che non sia rimasto `asm_enabled` in `localStorage` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L263-L284) e la chiave `oAuthRedirectCodeFlow`.
- **Redirect post-login nella pagina sbagliata.** Osserva `redirectUrl` nel `localStorage` `auth` durante la navigazione; valuta il toggle `redirectOnlyOnTrueNavigationEnd` e la marcatura `authFlow` delle route (core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L62-L72, core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts:L23-L46).
- **Loop sulla pagina di login (custom login page).** Ispeziona `login_redirect_count` in `localStorage` e la risposta della GET CSRF (core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L29-L38, core-libs/core/src/auth/client-auth/services/cross-site-request-forgery.service.ts:L30-L35): la GET richiede i cookie dell'Authorization Server (`withCredentials: true`).
- **Client token.** Con lo store devtools di NgRx osserva le azioni `[Token] Load Client Token*` e la feature `client-auth` (core-libs/core/src/auth/client-auth/store/actions/client-token.action.ts:L12-L41, core-libs/core/src/auth/client-auth/store/client-auth-state.ts:L10-L19); ricorda che senza `useClientTokens: true` l'interceptor si limita a rimuovere il marker.
- **Logout cross-tab non propagato.** Verifica il toggle `propagateLogoutToAllTabs` e che le tab siano sullo stesso base site (canale `spartacus_auth_notification`, core-libs/core/src/auth/user-auth/facade/auth-notification.service.ts:L11-L28).

---

11. Domande di autoverifica

1. **Perché `ClientAuthModule` deve essere importato dopo `UserAuthModule`?** Perché gli interceptor a classe sono concatenati con `reduceRight`: l'ultimo registrato è il più interno e vede per primo gli errori; così `ClientTokenInterceptor` può gestire i 401 dei client token prima dell'`AuthInterceptor` — core-libs/core/src/auth/auth.module.ts:L12-L16, angular:packages/common/http/src/interceptor.ts:L233-L240.
2. **Con i feature toggle di default, quale flow OAuth usa Spartacus e con quale `client_id`?** Authorization Code con PKCE (`responseType: 'code'`, `disablePKCE: false`), `client_id` `mobile_android_public`, senza `initializerOptions` né `redirectUri` configurati (ramo code flow senza async initializer) — core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L101-L115, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1090-L1090.
3. **Perché dopo un reload della pagina un access token scaduto porta al logout invece che a un refresh?** Perché `AuthStatePersistenceService` cancella il `refresh_token` prima di salvare nel `localStorage`; dopo il reload `refreshToken$` non trova il `refresh_token` e chiama `handleExpiredRefreshToken()` — core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L77-L83, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L134-L140.
4. **Se tre chiamate OCC parallele ricevono 401, quanti refresh partono?** Uno: `getValidToken` emette sul trigger solo se il token corrente è ancora quello della richiesta fallita e solo alla prima emissione, e

`refreshToken$` scarta i trigger mentre `refreshInProgress` è vero — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L356-L380, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L125-L133.

5. **Cosa succede di default quando il refresh token è scaduto, e come si può intervenire senza sovrascrivere** `AuthHttpHeaderService`**?** Fallback: salva l'URL corrente, `coreLogout()`, navigazione a `login` con messaggio `httpHandlers.sessionExpired`. Si può registrare un `ExpiredRefreshTokenHandler` in `EXPIRED_REFRESH_TOKEN_HANDLERS` che restituisca `of(true)` (toggle `enableExpiredRefreshTokenHandlers` attivo) — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L310, core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts:L17-L23.
6. **Perché** `AuthService.logout()` **non chiama direttamente la revoca?** Naviga alla cxRoute `logout`, dove `LogoutGuard` invoca `coreLogout()`; il JSDoc di `coreLogout` raccomanda di usare `logout` — core-libs/core/src/auth/user-auth/facade/auth.service.ts:L188-L191, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L222-L224, core-libs/storefront/cms-components/user/logout/logout.guard.ts:L87-L89.
7. `LogoutEvent` **è emesso dall'azione** `AuthActions.Logout`**?** No: deriva dalla transizione `true → false` di `isUserLoggedIn()`; `LoginEvent` invece è mappato sull'azione `AuthActions.LOGIN` — core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L40-L63.
8. **In quale momento del bootstrap viene letto il** `code` **OAuth dall'URL, e perché prima viene sincronizzato lo storage?** In un location initializer, risolto prima della navigazione iniziale; la sincronizzazione con il `localStorage` avviene prima per avere in memoria valori come il `nonce` — core-libs/core/src/auth/user-auth/user-auth.module.ts:L70-L80, core-libs/core/src/routing/routing.module.ts:L66-L71, angular:packages/router/src/provide_router.ts:L445-L452.
9. **Perché una richiesta a** `baseSites` **non riceve l'header** `Authorization` **anche se l'utente è loggato?** Di norma l'URL di `baseSites` è costruito con `{ baseSite: false }` e quindi non contiene il base site attivo incluso in `getBaseUrl()`: `isOccUrl` è già falso (core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L100-L100, core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L21-L21, core-libs/core/src/occ/services/occ-endpoints.service.ts:L93-L101). In più `alterRequest` esclude esplicitamente le richieste il cui URL contiene l'endpoint `baseSites` — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L188-L213.
10. **Con** `useClientTokens: false`**, cosa fa** `ClientTokenInterceptor` **su una richiesta marcata** `cx-use-client-token`**?** Rimuove il marker e passa la richiesta invariata, senza chiedere alcun token — core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L46-L52.

TAVOLA 22.1 **4.1 Bootstrap e ritorno dall'Authorization Server (Authorization Code)**

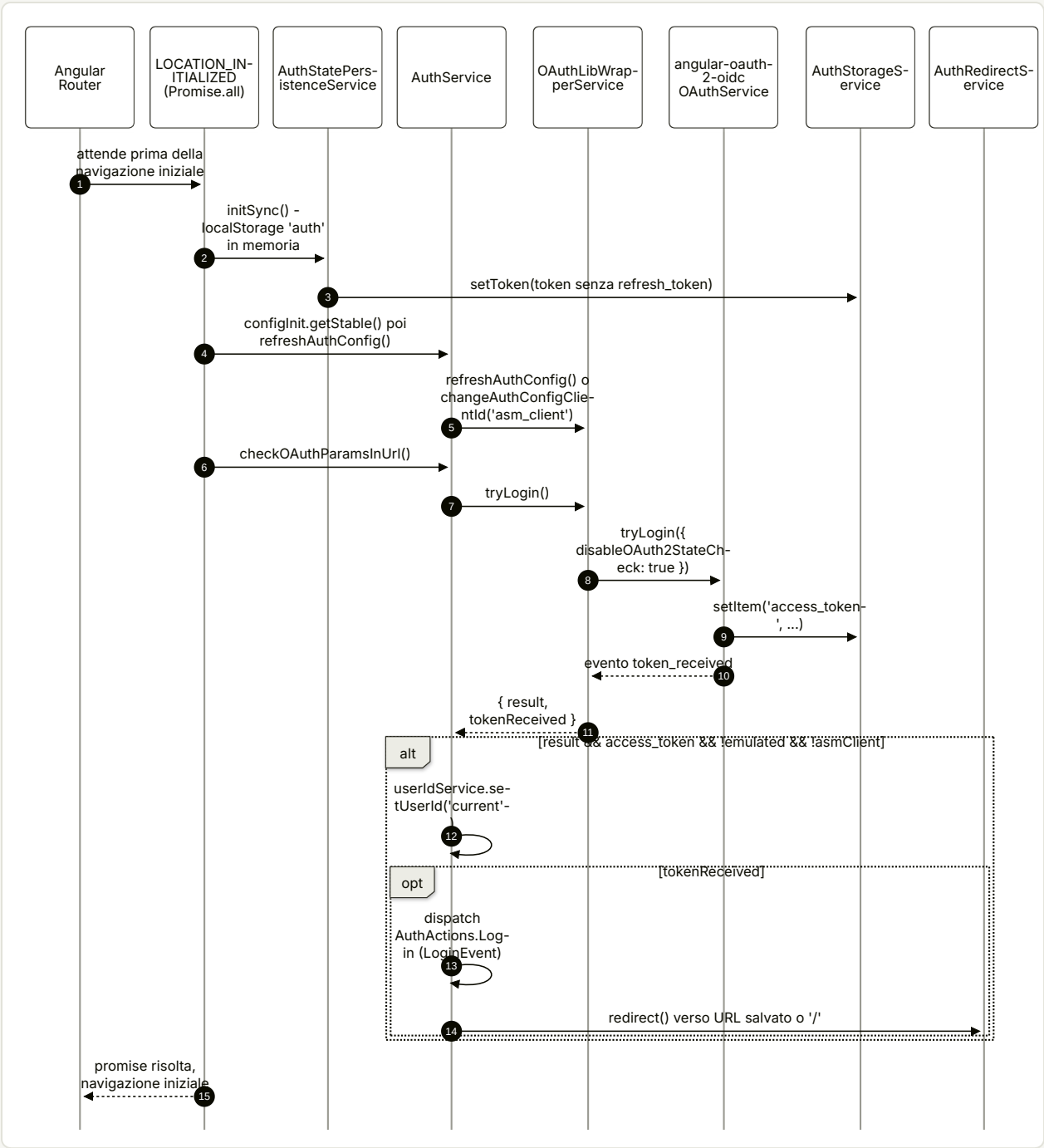


TAVOLA 22.2 **4.2 Catena degli interceptor su una richiesta OCC**

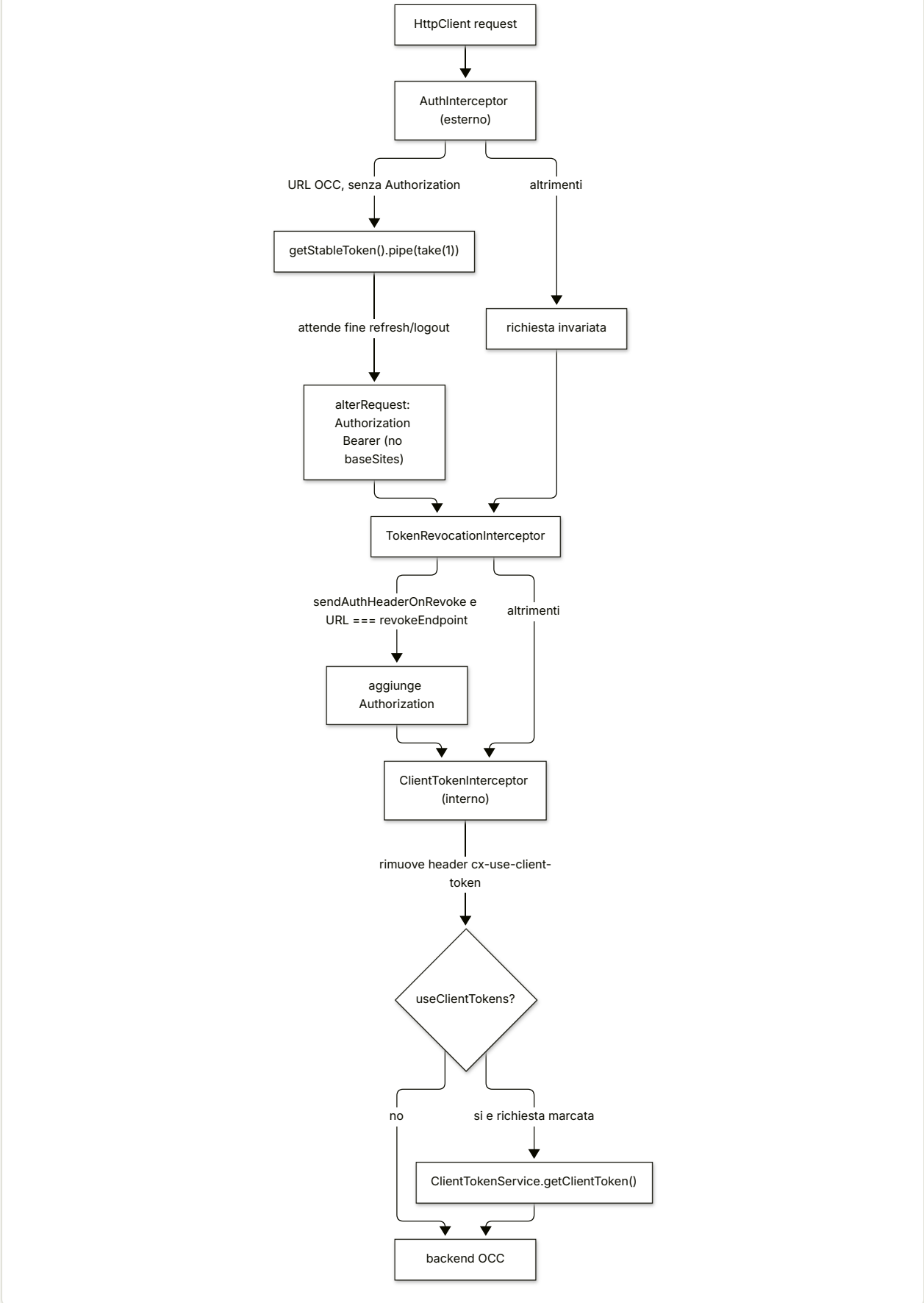
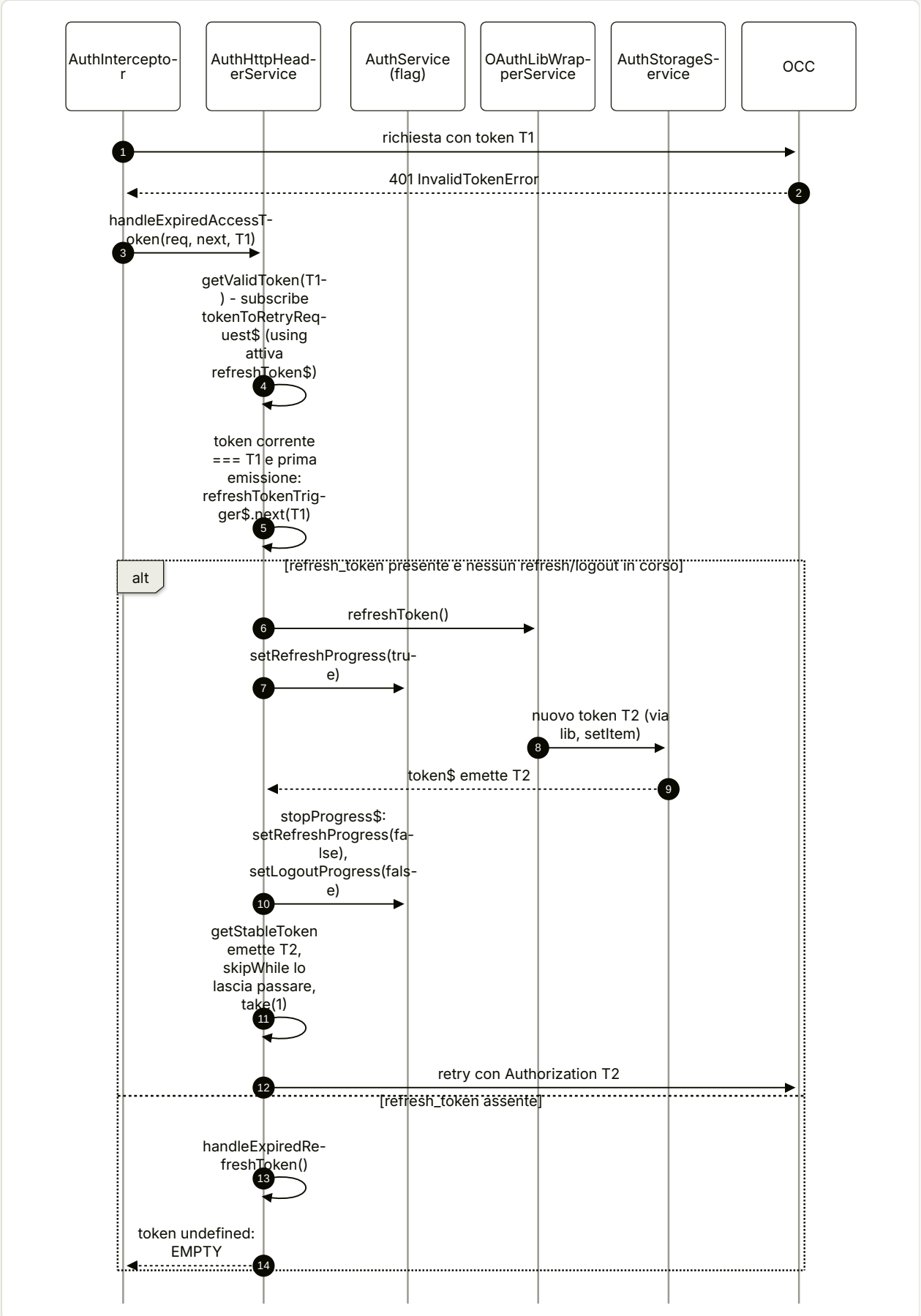


TAVOLA 22.3 **4.3 401 per access token scaduto: refresh e retry**



23

Utente, profilo, consensi

Un dominio diviso in tre strati di età e stile architetturale diversi.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

"L'utente" in Spartacus non è un unico modulo: è un dominio **spezzato in tre strati con età e stili architetturali diversi**, e la prima cosa da capire è dove sta ogni pezzo.

1. **Core utente "legacy" in** `@spartacus/core` (`core-libs/core/src/user`). È basato su NgRx: un feature store `user` con indirizzi, consensi del cliente registrato, paesi/regioni/città/distretti, metodi di pagamento, coupon cliente, preferenze di notifica, interessi prodotto e cost center B2B — `core-libs/core/src/user/store/user-state.ts:L52-L65`. Le facade (`UserAddressService`, `UserConsentService`, `UserPaymentService`, ...) sono classi `providedIn: 'root'` che fanno *fire-and-forget* sullo store — `core-libs/core/src/user/facade/user-address.service.ts:L44-L48`. Nell'app demo è registrato con `UserModule` + `UserOccModule`, moduli semplici, non `forRoot` — `projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L201-L217`.
2. **Consensi anonimi in** `@spartacus/core` (`core-libs/core/src/anonymous-consents`). Uno store separato (`anonymous-consents`), persistito in `localStorage`, più un interceptor HTTP che allega l'header `X-Anonymous-Consents` a ogni chiamata OCC — `core-libs/core/src/anonymous-consents/store/anonymous-consents-state.ts:L10-L24`, `core-libs/core/src/model/consent.model.ts:L21-L32`. Si attiva con `AnonymousConsentsModule.forRoot()` — `core-libs/core/src/anonymous-consents/anonymous-consents.module.ts:L14-L28`.
3. **Libreria** `@spartacus/user` (`feature-libs/user`), che riesporta due entry point — `feature-libs/user/public_api.ts:L19-L20`:
   - `@spartacus/user/account`: chi è l'utente loggato (GET `users/{userId}`), login form, mini-login, login OTP;
   - `@spartacus/user/profile`: registrazione, aggiornamento profilo/email/password, recupero password, chiusura account, rubrica indirizzi.

   Qui non c'è NgRx: lo stato è gestito con `Query` e `Command` (vedi capitolo 20) e la libreria è **lazy**, con il pattern `root` / `core` / `occ` / `components`. L'app importa solo i root module e mappa le feature su `import()` dinamici — `projects/storefrontapp/src/app/spartacus/features/user/user-feature.module.ts:L39-L78`.

A questi si aggiungono i **componenti storefront** in `core-libs/storefront/cms-components` (rotte `login`/`logout`, banner consensi anonimi, pagine My Account per consensi, coupon, notifiche, interessi, metodi di pagamento) — `projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L180-L181`, `:L201-L217`.

Perché questa frammentazione? Il sorgente stesso la presenta come una migrazione in corso: `UserRegisterService` di `@spartacus/user/profile` dispatcha ancora un'azione NgRx del core "as a compatibility mechanism only, to make anonymous consents management work properly in transitional period (when we move logic to separate libraries)" — `feature-libs/user/profile/core/facade/user-register.service.ts:L33-L43`. Allo stesso modo `UserProfileFacadeTransitionalToken` esiste perché codice del core possa leggere i titoli del profilo senza dipendere da `@spartacus/user` — `core-libs/core/src/user/user-transitional-tokens.ts:L10-L12`, `feature-libs/user/profile/root/user-profile-root.module.ts:L45-L53`.

Un'ultima nota di orientamento: `UserIdService`, che tutte le facade usano per sapere *per chi* chiamare OCC, **non sta nel modulo user** ma in `auth/user-auth/facade` — `core-libs/core/src/user/facade/user-address.service.ts:L11-L11`. È trattato nel capitolo 22; qui basta ricordare che emette `'current'` per un cliente loggato, `'anonymous'` senza login e il `customerId` durante l'emulazione ASM — `core-libs/core/src/auth/user-`

auth/facade/user-id.service.ts:L24-L48 — e che `takeUserId(true)` lancia errore se l'utente è anonimo — core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L50-L62.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/user/user.module.ts	`UserModule`: importa `UserStoreModule` + `UserEventModule`, fornisce `UserEventBuilder` e `defaultHierarchicalAddressConfig`
core-libs/core/src/user/user-transitional-tokens.ts	`UserProfileFacadeTransitionalToken` (solo `getTitles()`)
core-libs/core/src/user/index.ts	barrel pubblico (non esporta effects né reducers)
core-libs/core/src/user/config/hierarchical-address-config.ts	`HierarchicalAddressConfig`
core-libs/core/src/user/config/default-hierarchical-address-config.ts	default `['CN']`
core-libs/core/src/user/connectors/*/	connector `providedIn: 'root'` + adapter astratti + token converter per address, consent, cost-center, customer-coupon, interests, notification-preference, payment
core-libs/core/src/user/events/user-event.builder.ts	mapping azioni indirizzo → eventi
core-libs/core/src/user/events/user.events.ts	`AddUserAddressEvent`, `UpdateUserAddressEvent`, `DeleteUserAddressEvent`, ecc.
core-libs/core/src/user/facade/user-address.service.ts	indirizzi, paesi, regioni/città/distretti, verifica indirizzo (`Command`)
core-libs/core/src/user/facade/user-consent.service.ts	consensi utente registrato
core-libs/core/src/user/facade/consent.service.ts	`ConsentService`: facade unificata anonimi + registrati
core-libs/core/src/user/facade/user-payment.service.ts	metodi di pagamento e billing countries
core-libs/core/src/user/facade/customer-coupon.service.ts	coupon cliente
core-libs/core/src/user/facade/user-interests.service.ts	interessi prodotto
core-libs/core/src/user/facade/user-notification-preference.service.ts	canali di notifica
core-libs/core/src/user/facade/user-cost-center.service.ts	cost center B2B attivi
core-libs/core/src/user/store/user-state.ts	`USER_FEATURE`, ID di processo, entity type dei loader, `UserState`
core-libs/core/src/user/store/user-store.module.ts	`StoreModule.forFeature` + `EffectsModule.forFeature`
core-libs/core/src/user/store/reducers/index.ts	mappa reducer, `reducerToken`, meta-reducer `clearUserState`

PATH	RUOLO
core-libs/core/src/user/store/effects/index.ts	i 12 effect del feature `user`
core-libs/core/src/occ/adapters/user/user-occ.module.ts	`UserOccModule`: binding adapter OCC utente + consensi anonimi
core-libs/core/src/occ/adapters/user/default-occ-user-config.ts	endpoint OCC dell'area utente legacy
core-libs/core/src/anonymous-consents/anonymous-consents.module.ts	`AnonymousConsentsModule.forRoot()`
core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts	`AnonymousConsentsConfig`
core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts	`AnonymousConsentsService`
core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts	header `X-Anonymous-Consents` in richiesta e risposta
core-libs/core/src/anonymous-consents/services/anonymous-consents-state-persistence.service.ts	sync con `localStorage`
core-libs/core/src/anonymous-consents/store/anonymous-consents-store.module.ts	feature store + `APP_INITIALIZER` di persistenza
core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts	check versioni, caricamento template, trasferimento al login, consensi obbligatori
core-libs/core/src/occ/adapters/user/occ-anonymous-consent-templates.adapter.ts	GET template, HEAD per i consensi
feature-libs/user/account/user-account.module.ts	modulo lazy `UserAccountModule`
feature-libs/user/account/root/user-account-root.module.ts	`UserAccountRootModule`, CMS mapping lazy della feature `userAccount`
feature-libs/user/account/root/facade/user-account.facade.ts	`UserAccountFacade` lazy (`get`, `getById`)
feature-libs/user/account/root/facade/verification-token.facade.ts	`VerificationTokenFacade` lazy (OTP)
feature-libs/user/account/root/services/user-login-currency.service.ts	valuta utente al login / ripristino al logout
feature-libs/user/account/core/facade/user-account.service.ts	`UserAccountService` basato su `Query`
feature-libs/user/account/occ/adapters/occ-user-account.adapter.ts	`OccUserAccountAdapter`
feature-libs/user/account/components/login-form/login-form-component.service.ts	login ROPC o POST nativo (Authorization Code)
feature-libs/user/account/components/otp-login-form/otp-login-form.component.ts	primo passo del login OTP
feature-libs/user/account/components/verification-token-form/verification-token-form.component.ts	secondo passo del login OTP

PATH	RUOLO
feature-libs/user/profile/user-profile.module.ts	modulo lazy `UserProfileModule`
feature-libs/user/profile/root/user-profile-root.module.ts	root module, token transitorio, CMS mapping lazy `userProfile`
feature-libs/user/profile/root/facade/user-profile.facade.ts	`UserProfileFacade` lazy
feature-libs/user/profile/core/facade/user-profile.service.ts	update / close / titoli (`Command` + `Query`)
feature-libs/user/profile/core/facade/user-register.service.ts	registrazione, registrazione guest
feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts	`OccUserProfileAdapter`
feature-libs/user/profile/components/register/register.component.ts	form registrazione, checkbox newsletter → consenso anonimo
feature-libs/user/profile/components/address-book/address-form/address-form.component.ts	form indirizzo, verifica, formato gerarchico
core-libs/storefront/cms-components/user/login-route/login.guard.ts	`LoginGuard`: pagina CMS o redirect OAuth
core-libs/storefront/cms-components/user/logout/logout.guard.ts	`LogoutGuard`
core-libs/storefront/cms-components/anonymous-consent-management/anonymous-consent-management.module.ts	banner e dialog consensi anonimi
core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts	pagina consensi My Account
feature-libs/user/wishlist/occ/adapters/occ-user-wishlist.adapter.ts	adapter OCC wishlist (senza NgModule proprio)

File aggiuntivi

PATH	RUOLO
feature-libs/user/_index.scss	entry SCSS della lib: importa gli stili di account/profile, applica i placeholder `%cx-*` ai selettori (salvo `$skipComponentStyles`) e definisce `.user-form`
feature-libs/user/schematics/add-user/index.ts	rule `ng add` di `@spartacus/user` (`addUserFeatures`)
feature-libs/user/schematics/add-user/index_spec.ts	test Jest dello schematic: senza feature, User-Account, User-Profile, lazy/eager, snapshot di feature module, `user.scss` e `angular.json`

3. Come funziona passo per passo

3.1 Wiring all'avvio

1. `UserModule` importa `UserStoreModule` e `UserEventModule`, fornisce `UserEventBuilder` e la default config degli indirizzi gerarchici; il suo `forRoot()` restituisce solo `ngModule: UserModule`, senza provider aggiuntivi — core-libs/core/src/user/user.module.ts:L14-L27.
2. `UserStoreModule` registra il feature store `'user'` con `StoreModule.forFeature(USER_FEATURE, reducerToken, { metaReducers })` e `EffectsModule.forFeature(effects)` — core-libs/core/src/user/store/user-store.module.ts:L17-L27. La mappa dei reducer è fornita via `InjectionToken 'UserReducers'` con `useFactory: getReducers` — core-libs/core/src/user/store/reducers/index.ts:L91-L97.
3. `UserEventModule` inietta `UserEventBuilder` nel costruttore solo per istanziarlo — core-libs/core/src/user/events/user-event.module.ts:L10-L15.
4. `UserOccModule` collega con `useClass` tutti gli adapter OCC dell'area utente **e** quello dei template di consensi anonimi, registra i normalizer multi e la default config degli endpoint — core-libs/core/src/occ/adapters/user/user-occ.module.ts:L34-L72, core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L9-L34.
5. `AnonymousConsentsModule.forRoot()` aggiunge interceptor, `AnonymousConsentsService` e la default config; lo store module viene importato sempre — core-libs/core/src/anonymous-consents/anonymous-consents.module.ts:L14-L28. L'interceptor è registrato come `HTTP_INTERCEPTORS` multi `useExisting` — core-libs/core/src/anonymous-consents/http-interceptors/index.ts:L11-L17.
6. Lo store dei consensi anonimi registra un `APP_INITIALIZER` che chiama `AnonymousConsentsStatePersistenceService.initSync()` — core-libs/core/src/anonymous-consents/store/anonymous-consents-store.module.ts:L17-L43.
7. `UserAccountRootModule` importa `UserAccountEventModule` e fornisce la default config più una *config factory* con la mappa CMS lazy — feature-libs/user/account/root/user-account-root.module.ts:L68-L75. `UserAccountEventModule` istanzia eagerly `UserAccountEventListener` e `UserLoginCurrencyService` — feature-libs/user/account/root/events/user-account-event.module.ts:L11-L18.
8. `UserProfileRootModule` lega `UserProfileFacadeTransitionalToken` a `UserProfileFacade` e istanzia eagerly `UserCurrencyPreferenceSaverService` — feature-libs/user/profile/root/user-profile-root.module.ts:L45-L58.

3.2 Lo store `user` del core

- `UserState` ha 12 slice; quasi tutte sono `StateUtils.LoaderState<...>`, mentre `billingCountries` e `countries` sono reducer semplici — core-libs/core/src/user/store/user-state.ts:L52-L65, core-libs/core/src/user/store/reducers/index.ts:L47-L89. **Non** c'è lo slice "dettagli account": quello vive in `@spartacus/user/account` via `Query` — feature-libs/user/account/core/facade/user-account.service.ts:L26-L35.
- `user-state.ts` definisce anche gli **ID di processo** (`registerUser`, `giveConsent`, `withdrawConsent`, `updateNotificationPreferences`, …) e gli entity type dei loader (`'[User] User Addresses'`, …) — core-libs/core/src/user/store/user-state.ts:L22-L46.
- Il `loaderReducer` (vedi capitolo 19) si comporta in base a `action.meta.loader`: load → `loading: true`; error → `error: true, success: false`; success → `value = subReducer(state.value, action)` o `action.payload`; reset → stato iniziale — core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L69. Con un sub-reducer, questo viene chiamato sempre — core-libs/core/src/state/utils/loader/loader.reducer.ts:L80-L82.

- Le mutazioni indirizzo `AddUserAddress`, `UpdateUserAddress`, `DeleteUserAddress` estendono `LoaderLoadAction(USER_ADDRESSES)`, cioè **condividono il loader della lista** — core-libs/core/src/user/store/actions/user-addresses.action.ts:L56-L61, :L84-L91, :L114-L119. Il sub-reducer gestisce solo `LOAD_USER_ADDRESSES_FAIL` e `LOAD_USER_ADDRESSES_SUCCESS` — core-libs/core/src/user/store/reducers/user-addresses.reducer.ts:L12-L26.
- Il meta-reducer `clearUserState` azzera tutto lo stato `user` su `AuthActions.LOGOUT` — core-libs/core/src/user/store/reducers/index.ts:L99-L111.
- `UserActions` e `UsersSelectors` sono namespace (`import * as`) sui file "group" — core-libs/core/src/user/store/actions/index.ts:L7-L8, core-libs/core/src/user/store/selectors/index.ts:L7-L8; il feature selector è `createFeatureSelector<UserState>(USER_FEATURE)` — core-libs/core/src/user/store/selectors/feature.selector.ts:L10-L11.
- Gli effect sono 12 — core-libs/core/src/user/store/effects/index.ts:L20-L33. I più interessanti:
  - `ClearMiscsDataEffect`: `LANGUAGE_CHANGE`/`CURRENCY_CHANGE` → `ClearUserMiscsData` — core-libs/core/src/user/store/effects/clear-miscs-data.effect.ts:L17-L27. Le delivery countries si resettano nel reducer — core-libs/core/src/user/store/reducers/delivery-countries.reducer.ts:L20-L42; regioni, città e distretti invece tramite effect che emettono `LoaderResetAction` — core-libs/core/src/user/store/effects/regions.effect.ts:L50-L57, core-libs/core/src/user/store/effects/cities-districts.effect.ts:L50-L55, :L82-L87.
  - Paesi, regioni, città e distretti si caricano con `SiteConnector` (site-context), non con un connector utente — core-libs/core/src/user/store/effects/regions.effect.ts:L29-L37, core-libs/core/src/user/store/effects/delivery-countries.effect.ts:L26-L26, core-libs/core/src/user/store/effects/billing-countries.effect.ts:L26-L26.
  - Indirizzi: load con `switchMap`, add/update/delete con `mergeMap`, errori passati da `tryNormalizeHttpError` — core-libs/core/src/user/store/effects/user-addresses.effect.ts:L26-L70; un update fallito mostra `addressForm.invalidAddress` — core-libs/core/src/user/store/effects/user-addresses.effect.ts:L84-L94; dopo i success l'effect ricarica la lista chiamando la *facade* `UserAddressService.loadAddresses()` — core-libs/core/src/user/store/effects/user-addresses.effect.ts:L128-L177, :L179-L198.
  - Pagamenti: set default/delete emettono success e poi `LoadUserPaymentMethods` — core-libs/core/src/user/store/effects/payment-methods.effect.ts:L45-L97.
  - Consensi: `resetConsents$` su `LANGUAGE_CHANGE` — core-libs/core/src/user/store/effects/user-consents.effect.ts:L23-L29; `giveConsent$` gestisce `GIVE_USER_CONSENT` **e** `TRANSFER_ANONYMOUS_CONSENT` con `concatMap` — core-libs/core/src/user/store/effects/user-consents.effect.ts:L50-L65; un 409 sul trasferimento rimuove i messaggi d'errore — core-libs/core/src/user/store/effects/user-consents.effect.ts:L75-L85.
  - Interessi prodotto: dopo add/remove ricaricano la lista — core-libs/core/src/user/store/effects/product-interests.effect.ts:L58-L121.

3.3 Le facade del core

- Pattern comune: `takeUserId()` + `store.dispatch(...)` dentro `subscribe`, ritorno `void` — core-libs/core/src/user/facade/user-address.service.ts:L44-L48.
- `setAddressAsDefault` è un `UpdateUserAddress` con `address: { defaultAddress: true }` — core-libs/core/src/user/facade/user-address.service.ts:L69-L79; `getDefaultAddress()` cerca `defaultAddress` nella lista — core-libs/core/src/user/facade/user-address.service.ts:L116-L120.
- `getRegions(countryIsoCode)` contiene logica di caricamento *dentro* `map`: azzera, avvia `loadRegions`, ritorna `[]` durante il caricamento — core-libs/core/src/user/facade/user-address.service.ts:L188-L209;

`getCities`/`getDistricts` replicano lo schema — core-libs/core/src/user/facade/user-address.service.ts:L215-L240, :L261-L280.

- `verifyAddress` **non passa dallo store**: è un `Command` che chiama `UserAddressConnector.verify` con `takeUserId(false)`, quindi funziona anche per anonimi — core-libs/core/src/user/facade/user-address.service.ts:L301-L318. Lato OCC, per lo userId anonimo viene aggiunto l'header `USE_CLIENT_TOKEN` — core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L96-L105.
- `UserConsentService.getConsents(loadIfMissing)` con `iif`: se `true` carica i template quando mancano e non c'è già stato un success ("avoid infinite loop") — core-libs/core/src/user/facade/user-consent.service.ts:L60-L85.
- `UserConsentService.getConsent(templateId)` emette solo se `isUserLoggedIn()` **e** `userId === OCC_USER_ID_CURRENT` (soluzione dichiarata temporanea, CXSPA-2988) — core-libs/core/src/user/facade/user-consent.service.ts:L128-L154.
- Un consenso registrato è "dato" se ha `consentGivenDate` e nessuna `consentWithdrawnDate`; un consenso assente è considerato ritirato — core-libs/core/src/user/facade/user-consent.service.ts:L163-L178.
- Gli stati di give/withdraw si leggono dal feature `process` (`getProcessLoadingFactory(GIVE_CONSENT_PROCESS_ID)`, …) — core-libs/core/src/user/facade/user-consent.service.ts:L200-L222, perché le azioni sono `EntityLoadAction` su `PROCESS_FEATURE` — core-libs/core/src/user/store/actions/user-consents.action.ts:L71-L83.
- `ConsentService` unifica: `getConsent` fa `merge` del flusso registrato e di quello anonimo — core-libs/core/src/user/facade/consent.service.ts:L14-L35; distingue i tipi per forma (`templateCode` vs `code`) — core-libs/core/src/user/facade/consent.service.ts:L106-L125.
- `UserCostCenterService.getActiveCostCenters()` usa `observeOn(queueScheduler)` e auto-carica solo per utenti loggati — core-libs/core/src/user/facade/user-cost-center.service.ts:L31-L62; `CustomerCouponService.getCustomerCoupons` e `UserInterestsService.getAndLoadProductInterests` usano lo schema "attemptedLoad" — core-libs/core/src/user/facade/customer-coupon.service.ts:L66-L86, core-libs/core/src/user/facade/user-interests.service.ts:L81-L127.

3.4 Connector, adapter, endpoint (core)

Ogni area ha un connector `providedIn: 'root'` che delega a un adapter astratto — core-libs/core/src/user/connectors/address/user-address.connector.ts:L12-L37, core-libs/core/src/user/connectors/address/user-address.adapter.ts:L10-L27 (pattern generale nel capitolo 21). Particolarità:

- `UserConsentAdapter.withdrawConsent` ha un terzo parametro opzionale `consentId` — core-libs/core/src/user/connectors/consent/user-consent.adapter.ts:L19-L23 — che l'implementazione OCC ignora, usando `consentCode` come `consentId` dell'URL — core-libs/core/src/occ/adapters/user/occ-user-consent.adapter.ts:L69-L78.
- `CustomerCouponConnector.claimCustomerCoupon` usa la variante con codice nel body — core-libs/core/src/user/connectors/customer-coupon/customer-coupon.connector.ts:L42-L47.
- Il barrel `core-libs/core/src/user/connectors/index.ts` non esporta `notification-preference` — core-libs/core/src/user/connectors/index.ts:L7-L12.

3.5 Consensi anonimi

1. **Persistenza.** `initSync()` chiama `StatePersistenceService.syncWithStorage` con chiave `'anonymous-consents'` e l'intero stato del feature — core-libs/core/src/anonymous-consents/services/anonymous-consents-state-persistence.service.ts:L42-L63. Il default è `localStorage` con contesto vuoto, quindi la chiave effettiva è `spartacus⁉️⁉️anonymous-consents` — core-libs/core/src/state/services/state-persistence.service.ts:L45-L48, :L124-L131. In lettura (`onRead`) reidrata i template solo se `templates`.

success, poi i consensi e i flag UI — core-libs/core/src/anonymous-consents/services/anonymous-consents-state-persistence.service.ts:L69-L91.

2. **Stato.** templates (loader), consents (array di {templateCode, templateVersion, consentState}), ui. {bannerDismissed, updated} — core-libs/core/src/anonymous-consents/store/anonymous-consents-state.ts:L10-L24, core-libs/core/src/anonymous-consents/store/reducers/index.ts:L27-L36. Give/withdraw cambiano lo stato del solo consenso con quel templateCode; SET_ANONYMOUS_CONSENTS sostituisce l'array — core-libs/core/src/anonymous-consents/store/reducers/anonymous-consents.reducer.ts:L15-L62. Su LOGOUT o LANGUAGE_CHANGE un meta-reducer azzera solo templates — core-libs/core/src/anonymous-consents/store/reducers/index.ts:L49-L65.
3. **Banner.** Il componente è caricato con DeferLoadingStrategy.INSTANT — core-libs/storefront/cms-components/anonymous-consent-management/anonymous-consent-management.module.ts:L32-L42. isBannerVisible() ritorna false per utenti loggati; altrimenti è visibile se non dismesso o se i template sono cambiati, e a ogni emissione dispatcha AnonymousConsentCheckUpdatedVersions — core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L281-L309.
4. **Check versioni.** checkConsentVersions$ chiama loadAnonymousConsents() e, se le versioni differiscono, dispatcha LoadAnonymousConsentTemplates — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L36-L87. L'adapter OCC fa una **HEAD** sull'endpoint dei template e legge l'header X-Anonymous-Consents — core-libs/core/src/occ/adapters/user/occ-anonymous-consent-templates.adapter.ts:L48-L62. Se il backend non restituisce consensi, in dev mode viene loggato un warning — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L49-L55.
5. **Caricamento template.** loadAnonymousConsentTemplates$ emette LoadAnonymousConsentTemplatesSuccess e ToggleAnonymousConsentTemplatesUpdated(updated), con updated calcolato solo se esistevano template — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L89-L129; il confronto (detectUpdatedTemplates) guarda lunghezza e version per posizione — core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L316-L333.
6. **Interceptor.** Per ogni richiesta prende una volta consensi e stato login, ignora URL non OCC — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L36-L48, :L110-L112; aggiunge l'header con i consensi serializzati (JSON + encodeURIComponent) — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L93-L108; sulla risposta dell'endpoint template, per utenti non loggati, decodifica l'header, forza a GIVEN i requiredConsents e fa setConsents solo se cambiati — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L51-L91, :L114-L135. Il confronto consentsUpdated è tra stringhe serializzate — core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L369-L376.
7. **Al login.** Due effect reagiscono a AuthActions.LOGIN:
   - transferAnonymousConsentsToUser$: solo se in precedenza si è visto REGISTER_USER_SUCCESS (withLatestFrom), trasforma i consensi anonimi GIVEN non obbligatori in TransferAnonymousConsent — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L133-L184;
   - giveRequiredConsentsToUser$: carica i consensi utente se mancano e dà quelli obbligatori risultati ritirati — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L195-L247.

3.6 @spartacus/user/account

1. UserAccountModule (lazy) = UserAccountCoreModule + UserAccountOccModule + UserAccountComponentsModule — feature-libs/user/account/user-account.module.ts:L12-L19.

2. La root dichiara le feature `userAccount` e `userAccountCore` — feature-libs/user/account/root/feature-name.ts:L7-L8 — e, nella config factory, elenca i CMS component e mappa la core sulla feature dei componenti ("by default core is bundled together with components") — feature-libs/user/account/root/user-account-root.module.ts:L23-L48. `OAuthCallbackComponent` viene tolto se non sono attivi insieme `oauthCallbackPage`, `asyncAuthConfigInitializer`, `authorizationCodeFlowByDefault` — feature-libs/user/account/root/user-account-root.module.ts:L50-L63.
3. `UserAccountFacade` è una facade lazy creata con `facadeFactory` sulla feature `userAccountCore` (metodi `get`, `getById`) — feature-libs/user/account/root/facade/user-account.facade.ts:L13-L25 (meccanismo nel capitolo 12). Il core la lega a `UserAccountService` con `useExisting` — feature-libs/user/account/core/facade/facade-providers.ts:L15-L26.
4. `UserAccountService.get()` è una `Query` che chiama `users/{userId}` solo per utenti loggati, si ricarica su `UserAccountChangedEvent` e si resetta su `LoginEvent`/`LogoutEvent` — feature-libs/user/account/core/facade/user-account.service.ts:L26-L35. `getById` bypassa la query — feature-libs/user/account/core/facade/user-account.service.ts:L43-L49.
5. Endpoint: `user: 'users/${userId}'`, `createVerificationToken: 'users/anonymous/verificationToken'`, aggiunti a `OccEndpoints` via module augmentation — feature-libs/user/account/occ/adapters/config/default-occ-user-account-endpoint.config.ts:L9-L18, feature-libs/user/account/occ/adapters/config/occ-user-account-endpoint.model.ts:L20-L22.
6. Il modello `User` di account è minimale e aumentabile — feature-libs/user/account/root/model/user.model.ts:L7-L15, feature-libs/user/account/root/public_api.ts:L17-L19; `@spartacus/user/profile/root` lo aumenta con currency, language, defaultAddress, roles, … — feature-libs/user/profile/root/model/user-profile.model.ts:L9-L26.
7. Su `LogoutEvent` il listener mostra `authMessages.signedOutSuccessfully` — feature-libs/user/account/root/events/user-account-event.listener.ts:L35-L44.

Login form. `ReturningCustomerLoginComponent` → `LoginFormComponent` con guard `NotAuthGuard`, `CustomLoginGuard` e `LoginFormComponentService` fornito a livello di componente — feature-libs/user/account/components/login-form/login-form.module.ts:L50-L64. Il componente è standalone, OnPush, con `ngSkipHydration` — feature-libs/user/account/components/login-form/login-form.component.ts:L32-L65.

- Con `authorizationCodeFlowByDefault` (default `true`, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029) il costruttore del servizio chiama `initCustomLogin()`: form `POST` nativo verso `getCustomLoginFormEndpoint()` con un controllo `csrf` obbligatorio — feature-libs/user/account/components/login-form/login-form-component.service.ts:L90-L93, :L275-L280.
- Con `authorizationCodeFlowByDefaultCsrfTokenRefresh` il CSRF è rinfrescato subito prima del submit, che avviene *prima* di `busy$.next(true)` — feature-libs/user/account/components/login-form/login-form-component.service.ts:L101-L131. Se il refresh fallisce, l'errore va in `sessionStorage` e si naviga "hard" a `/login` — feature-libs/user/account/components/login-form/login-form-component.service.ts:L133-L166.
- Senza code flow (ROPC), `login()` chiama `auth.loginWithCredentials(userId.toLowerCase(), password)` — feature-libs/user/account/components/login-form/login-form-component.service.ts:L198-L213.
- Al rientro, `handleCustomLoginError` legge l'errore da `sessionStorage` (`cx_login_error`) o dal query param `error`, e lo traduce in chiavi i18n tramite `resolveLoginErrorKey` — feature-libs/user/account/components/login-form/login-form-component.service.ts:L217-L262.

Login OTP. `OneTimePasswordLoginFormComponent` chiama `createVerificationToken` e naviga a `verifyToken` passando loginId, password, tokenId, expiresIn nello `state` — feature-libs/user/account/components/otp-login-form/otp-login-form.component.ts:L132-L173. `VerificationTokenService` è un `Command` — feature-libs/user/account/core/facade/verification-token.service.ts:L22-L38; l'adapter invia POST con `USE_CLIENT_TOKEN` — feature-libs/user/account/occ/adapters/occ-user-account.adapter.ts:L54-L79. Il form

di verifica legge lo stato da `history.state` e lo sovrascrive con `pushState` — `feature-libs/user/account/components/verification-token-form/verification-token-form.component.ts:L110-L124`, poi usa `otpLoginWithCredentials` (ROPC) o il submit nativo (code flow) — `feature-libs/user/account/components/verification-token-form/verification-token-form-component.service.ts:L64-L87`.

3.7 `@spartacus/user/profile`

1. `UserProfileModule` (lazy) = Core + Occ + Components + `CaptchaModule` — `feature-libs/user/profile/user-profile.module.ts:L13-L21`. I CMS component della feature `userProfile` sono elencati nella root — `feature-libs/user/profile/root/user-profile-root.module.ts:L21-L43`.
2. Quattro facade lazy sulla feature `userProfileCore`: `UserEmailFacade`, `UserPasswordFacade`, `UserProfileFacade`, `UserRegisterFacade` — `feature-libs/user/profile/root/facade/user-profile.facade.ts:L14-L22`, `feature-libs/user/profile/root/facade/user-register.facade.ts:L14-L22`; implementazioni collegate con `useExisting` — `feature-libs/user/profile/core/facade/facade-providers.ts:L18-L39`.
3. `UserProfileService.get()` delega a `UserAccountFacade.get()`: il profilo non ha stato proprio — `feature-libs/user/profile/core/facade/user-profile.service.ts:L79-L81`.
4. `updateCommand` (strategia `Queue`) fa PATCH e dispatcha `UserAccountChangedEvent` — `feature-libs/user/profile/core/facade/user-profile.service.ts:L31-L48`; `closeCommand` fa `remove(uid)` e poi `authService.logout()` — `feature-libs/user/profile/core/facade/user-profile.service.ts:L50-L60`; `titleQuery` si ricarica su `LanguageSetEvent` — `feature-libs/user/profile/core/facade/user-profile.service.ts:L62-L67`.
5. `registerCommand` chiama il connector e in `tap` dispatcha `UserActions.RegisterUserSuccess()` nello store del core — `feature-libs/user/profile/core/facade/user-register.service.ts:L33-L43`. `registerGuestCommand` fa login con credenziali in ROPC o naviga a `login` con code flow — `feature-libs/user/profile/core/facade/user-register.service.ts:L45-L62`.
6. Adapter OCC: `update` (PATCH) e `close` (DELETE) usano gli endpoint opzionali `userUpdateProfile`/`userCloseAccount` se configurati, altrimenti `user` — `feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L49-L60`, `:L164-L174`; `register` invia POST con `USE_CLIENT_TOKEN` e captcha token — `feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L62-L96`; `updateEmail` POST a `users/${userId}/login` con `{ newLoginId, password }` — `feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L124-L142`.
7. Componenti: la registrazione collega la checkbox `newsletter` al consenso anonimo `registerConsent` — `feature-libs/user/profile/components/register/register.component.ts:L204-L209`, `:L272-L283`; dopo la registrazione naviga a `login` — `feature-libs/user/profile/components/register/register.component.ts:L260-L270`. Cambio email e cambio password terminano con `coreLogout()` e navigazione a `login` — `feature-libs/user/profile/components/update-email/update-email-component.service.ts:L77-L104`, `feature-libs/user/profile/components/update-password/update-password-component.service.ts:L92-L112`. Il modal di chiusura account chiama `coreLogout()` e va a `home` — `feature-libs/user/profile/components/close-account/components/close-account-modal/close-account-modal.component.ts:L85-L100`.
8. Rubrica: `AddressBookComponentService` è un sottile wrapper sopra `UserAddressService` del core — `feature-libs/user/profile/components/address-book/address-book.component.service.ts:L11-L42`. Il form indirizzo, per i paesi a formato gerarchico (solo con toggle `enableHierarchicalAddressFormat`), salta la verifica OCC — `feature-libs/user/profile/components/address-book/address-form/address-form.component.ts:L413-L420`, `:L499-L518`.

3.8 Rotte `login`/`logout` e My Account (storefront)

- `LoginRouteModule` registra `cxRoute: 'login'` con `LoginGuard` — core-libs/storefront/cms-components/user/login-route/login-route.module.ts:L15-L36. `LoginGuard` lascia passare alla pagina CMS se il flow è ROPC o l'utente è già loggato, altrimenti chiama `loginWithRedirect()` e ritorna `EMPTY` — core-libs/storefront/cms-components/user/login-route/login.guard.ts:L34-L65.
- `LogoutModule` registra `cxRoute: 'logout'` con `[LogoutGuard, CmsPageGuard]` e default `logout.redirectRoute: 'home'` — core-libs/storefront/cms-components/user/logout/logout.module.ts:L18-L33, core-libs/storefront/cms-components/user/logout/config/default-logout-config.ts:L9-L13. Con `useConfigurableLogoutRedirect` il guard esegue `coreLogout()` e reindirizza secondo la priorità login (storefront protetto) → `redirectRoute` → home — core-libs/storefront/cms-components/user/logout/logout.guard.ts:L71-L110; senza toggle controlla se esiste una content page `logout` — core-libs/storefront/cms-components/user/logout/logout.guard.ts:L46-L69.
- La pagina consensi carica i consensi se mancano, nasconde i template anonimi se `showAnonymousConsents` è false o quelli in `hideConsents` — core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L127-L192; dà consenso con `template.id + version`, lo ritira con `currentConsent.code + template.id` — core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L230-L245; dopo un withdraw riuscito ricarica — core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L203-L222.
- Coupon: rotta `couponClaim` con `AuthGuard + CmsPageGuard` — core-libs/storefront/cms-components/myaccount/my-coupons/my-coupons.module.ts:L48-L81; `CouponClaimComponent` legge `couponCode` dai parametri e fa il claim — core-libs/storefront/cms-components/myaccount/my-coupons/coupon-claim/coupon-claim.component.ts:L29-L52.
- Notifiche e metodi di pagamento chiamano le facade core in `ngOnInit` — core-libs/storefront/cms-components/myaccount/notification-preference/notification-preference.component.ts:L42-L47, core-libs/storefront/cms-components/myaccount/payment-methods/payment-methods.component.ts:L52-L67.

3.9 Stili e schematic della libreria

`feature-libs/user/_index.scss` importa gli stili di `account` e `profile`, poi per ogni selettore della lista `$selectors` (login, register, address book, OTP, my-account-v2…) fa `@extend %<selettore>` e `%<selettore>__body` sul `body`, saltando quelli elencati in `$skipComponentStyles` (feature-libs/user/_index.scss:L3-L32); definisce inoltre la classe globale `.user-form`, un layout flex che da `md` in su occupa il 50% e rende `form`/`fieldset` `display: contents` (feature-libs/user/_index.scss:L34-L71).

Lo schematic `addUserFeatures` valida l'installazione di Spartacus, calcola le feature con `analyzeCrossFeatureDependencies` e concatena `analyzeApplication`, `addFeatures`, `addPackageJsonDependenciesForLibrary(peerDependencies)` e `finalizeInstallation` (feature-libs/user/schematics/add-user/index.ts:L25-L43). Il suo spec verifica che senza feature non venga creato il feature module ma siano installate le `peerDependencies` (saltando `SPARTACUS_SCHEMATICS`, aggiunta solo dalla Angular CLI) (feature-libs/user/schematics/add-user/index_spec.ts:L46-L84), e confronta con snapshot il feature module lazy/eager, `src/styles/spartacus/user.scss` e `angular.json` per User-Account e User-Profile (feature-libs/user/schematics/add-user/index_spec.ts:L86-L181). Gli schematics in generale sono nel capitolo 43.

4. Diagrammi

4.1 Architettura: dove vive ogni pezzo

■ TAVOLA 23.1 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: wiring app projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L180-L217, projects/storefrontapp/src/app/spartacus/features/user/user-feature.module.ts:L39-L78; moduli lazy feature-libs/user/account/user-account.module.ts:L12-L19, feature-libs/user/profile/user-profile.module.ts:L13-L21; ponti verso il core feature-libs/user/profile/core/facade/user-register.service.ts:L33-L43, feature-libs/user/profile/components/address-book/address-book.component.service.ts:L11-L42.

4.2 Consensi anonimi: avvio, banner, header HTTP

■ TAVOLA 23.2 diagramma a tutta pagina, in fondo al capitolo

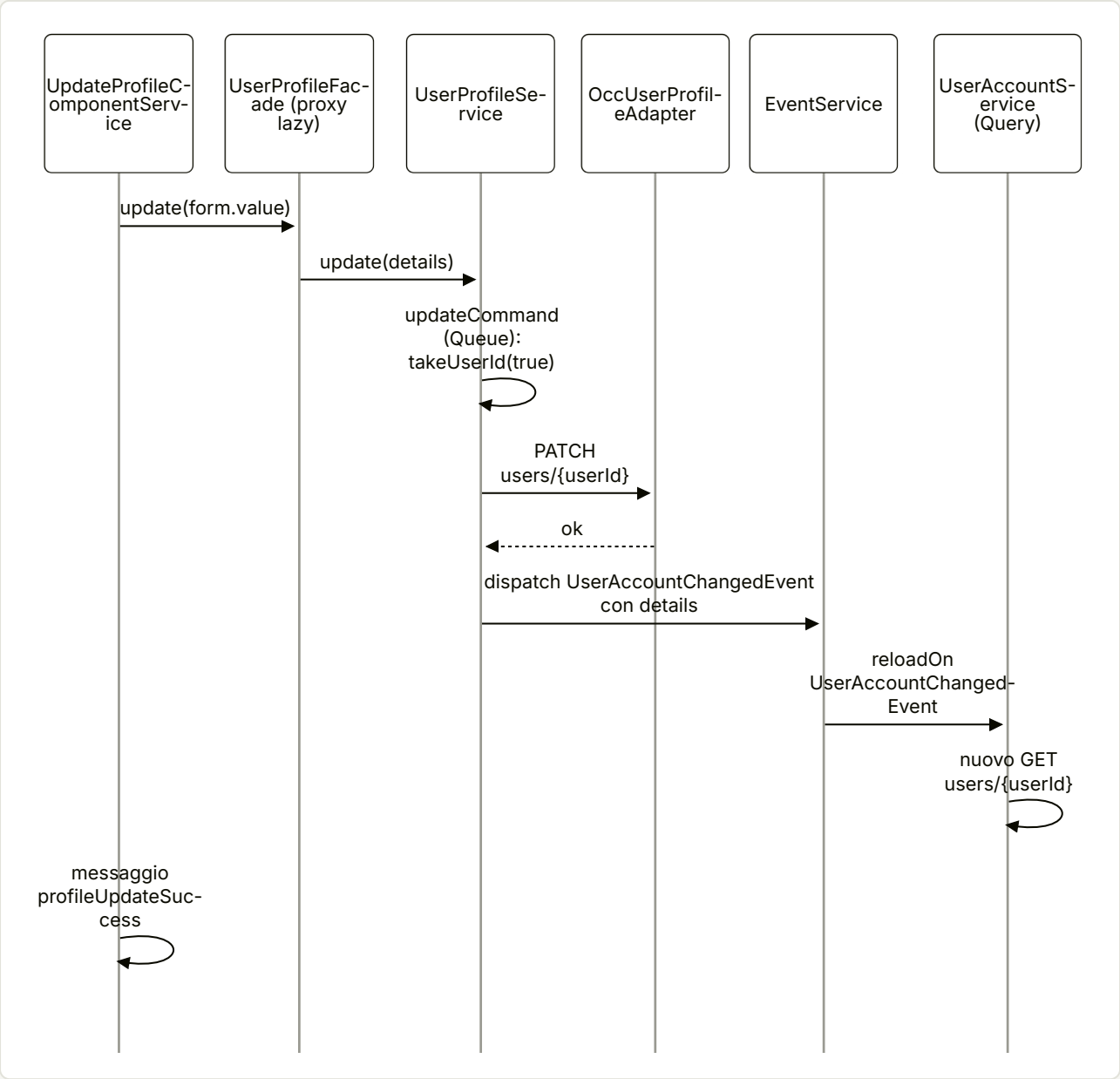
Riferimenti: core-libs/core/src/anonymous-consents/store/anonymous-consents-store.module.ts:L17-L43, core-libs/core/src/anonymous-consents/services/anonymous-consents-state-persistence.service.ts:L42-L91, core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L281-L309, core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L36-L129, core-libs/core/src/occ/adapters/user/occ-anonymous-consent-templates.adapter.ts:L48-L62, core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L36-L135.

4.3 Registrazione e trasferimento dei consensi anonimi

■ TAVOLA 23.3 diagramma a tutta pagina, in fondo al capitolo

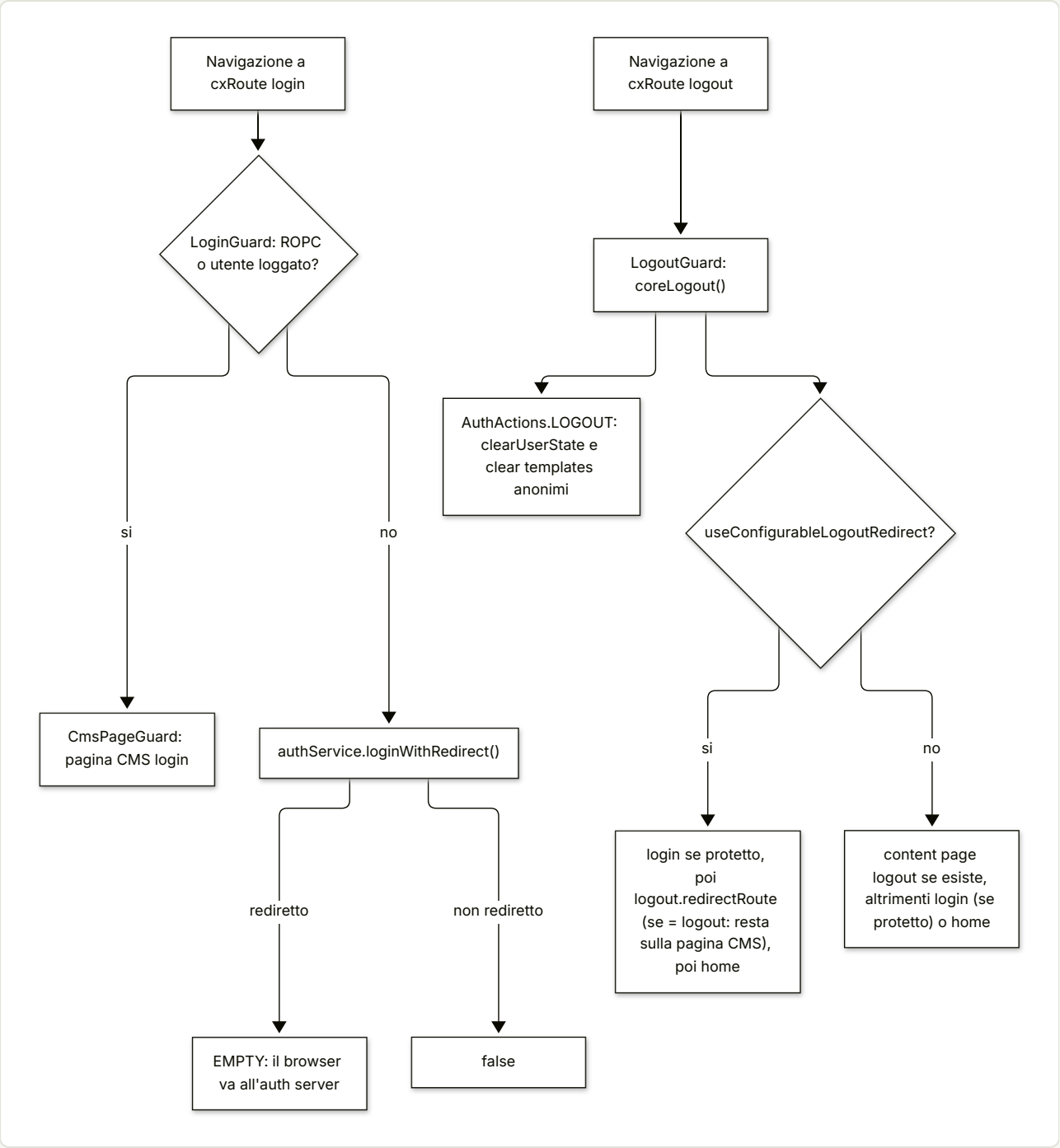
Riferimenti: feature-libs/user/profile/components/register/register.component.ts:L204-L209, :L260-L283, feature-libs/user/profile/core/facade/user-register.service.ts:L33-L43, feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L62-L96, core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L133-L247, core-libs/core/src/user/store/effects/user-consents.effect.ts:L50-L90.

4.4 Aggiornamento profilo e reload della `Query` utente



Riferimenti: `feature-libs/user/profile/components/update-profile/update-profile-component.service.ts:L34-L77`, `feature-libs/user/profile/root/facade/user-profile.facade.ts:L14-L22`, `feature-libs/user/profile/core/facade/user-profile.service.ts:L31-L48`, `feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L49-L60`, `feature-libs/user/account/core/facade/user-account.service.ts:L26-L35`.

4.5 Rotte `login` e `logout`



Riferimenti: `core-libs/storefront/cms-components/user/login-route/login.guard.ts:L34-L65`, `core-libs/storefront/cms-components/user/logout/logout.guard.ts:L46-L110`, `core-libs/core/src/user/store/reducers/index.ts:L99-L111`, `core-libs/core/src/anonymous-consents/store/reducers/index.ts:L49-L65`.

---

5. Estratti di codice reale

Il meta-reducer che svuota lo store `user` al logout — `core-libs/core/src/user/store/reducers/index.ts:L99-L111`

```ts
export function clearUserState(
  reducer: ActionReducer<any>
): ActionReducer<any> {
  return function (state, action) {
    if (action.type === AuthActions.LOGOUT) {
      state = undefined;
    }

    return reducer(state, action);
  };
}

export const metaReducers: MetaReducer<any>[] = [clearUserState];
```

Facade legacy: fire-and-forget e `Command` fuori store — core-libs/core/src/user/facade/user-address.service.ts:L44-L48 e :L301-L318

```ts
  loadAddresses(): void {
    this.userIdService.takeUserId().subscribe((userId) => {
      this.store.dispatch(new UserActions.LoadUserAddresses(userId));
    });
  }
// ...
  verifyAddress(address: Address): Observable<AddressValidation> {
    return this.userAddressVerificationCommand.execute({ address });
  }

  protected userAddressVerificationCommand: Command<
    {
      address: Address;
    },
    AddressValidation
  > = this.command.create((payload) =>
    this.userIdService
      .takeUserId(false)
      .pipe(
        switchMap((userId) =>
          this.userAddressConnector.verify(userId, payload.address)
        )
      )
  );
```

Effetti collaterali dentro `map` — core-libs/core/src/user/facade/user-address.service.ts:L188-L209

```ts
  getRegions(countryIsoCode: string): Observable<Region[]> {
    return this.store.pipe(
      select(UsersSelectors.getRegionsDataAndLoading),
      map(({ regions, country, loading, loaded }) => {
```

```
        if (!countryIsoCode && (loading || loaded)) {
          this.clearRegions();
          return [];
        } else if (loading && !loaded) {
          // don't interrupt loading
          return [];
        } else if (!loading && countryIsoCode !== country && countryIsoCode) {
          // country changed - clear store and load new regions
          if (country) {
            this.clearRegions();
          }
          this.loadRegions(countryIsoCode);
          return [];
        }
        return regions;
      })
    );
  }
```

getConsents(loadIfMissing) **e la guardia contro il loop —** core-libs/core/src/user/facade/user-consent.service.ts:L60-L85

TS

```
  getConsents(loadIfMissing = false): Observable<ConsentTemplate[]> {
    return iif(
      () => loadIfMissing,
      (<Store<StateWithUser>>this.store).pipe(
        select(UsersSelectors.getConsentsValue),
        withLatestFrom(
          this.getConsentsResultLoading(),
          this.getConsentsResultSuccess()
        ),
        filter(([_templates, loading, _success]) => !loading),
        tap(([templates, _loading, success]) => {
          if (!templates || templates.length === 0) {
            // avoid infinite loop - if we've already attempted to load templates and we got an empty
array as the response
            if (!success) {
              this.loadConsents();
            }
          }
        }),
        filter(([templates, _loading]) => Boolean(templates)),
        map(([templates, _loading]) => templates)
      ),
      (<Store<StateWithUser>>this.store).pipe(
        select(UsersSelectors.getConsentsValue)
```

```
      )
    );
  }
```

La facade unificata dei consensi — core-libs/core/src/user/facade/consent.service.ts:L14-L35

```ts
/**
 * Unified facade for both anonymous and registered user consents.
 */
@Injectable({ providedIn: 'root' })
export class ConsentService {
  constructor(
    protected anonymousConsentsService: AnonymousConsentsService,
    protected userConsentService: UserConsentService
  ) {}

  /**
   * Returns either anonymous consent or registered consent as they are emitted.
   * @param templateCode for which to return either anonymous or registered consent.
   */
  getConsent(
    templateCode: string
  ): Observable<AnonymousConsent | Consent | undefined> {
    return merge(
      this.userConsentService.getConsent(templateCode),
      this.anonymousConsentsService.getConsent(templateCode)
    );
  }
```

L'interceptor dei consensi anonimi — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L36-L69 e :L110-L112

```ts
  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    return combineLatest([
      this.anonymousConsentsService.getConsents(),
      this.authService.isUserLoggedIn(),
    ]).pipe(
      take(1),
      switchMap(([consents, isUserLoggedIn]) => {
        if (!this.isOccUrl(request.url)) {
          return next.handle(request);
        }

        const clonedRequest = this.handleRequest(consents, request);
        return next.handle(clonedRequest).pipe(
```

```
          tap((event) => {
            if (
              event instanceof HttpResponse &&
              (event.url ?? '').startsWith(
                this.occEndpoints.buildUrl('anonymousConsentTemplates')
              )
            ) {
              this.handleResponse(
                isUserLoggedIn,
                event.headers.get(ANONYMOUS_CONSENTS_HEADER),
                consents
              );
            }
          })
        );
      })
    );
  }
// ...
  private isOccUrl(url: string): boolean {
    return url.includes(this.occEndpoints.getBaseUrl());
  }
```

Banner e check versioni a ogni emissione — core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L281-L299

```ts
  isBannerVisible(): Observable<boolean> {
    return this.authService
      .isUserLoggedIn()
      .pipe(
        switchMap((isLoggedIn) =>
          isLoggedIn ? of(false) : this.getAnonymousBannerVisible()
        )
      );
  }

  private getAnonymousBannerVisible(): Observable<boolean> {
    return combineLatest([
      this.isBannerDismissed(),
      this.getTemplatesUpdated(),
    ]).pipe(
      tap(() => this.checkConsentVersions()),
      map(([dismissed, updated]) => !dismissed || updated)
    );
  }
```

Trasferimento dei consensi al login (estratto) — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L131-L146

```ts
  // TODO(#9416): This won't work with flow different than `Resource Owner Password Flow` which involves redirect (maybe in popup in will work)

  transferAnonymousConsentsToUser$: Observable<
    UserActions.TransferAnonymousConsent | Observable<never>
  > = createEffect(() =>
    this.actions$.pipe(
      ofType<AuthActions.Login>(AuthActions.LOGIN),
      filter(() => Boolean(this.anonymousConsentsConfig.anonymousConsents)),
      withLatestFrom(
        this.actions$.pipe(
          ofType<UserActions.RegisterUserSuccess>(
            UserActions.REGISTER_USER_SUCCESS
          )
        )
      ),
      filter(([, registerAction]) => Boolean(registerAction)),
```

Il ponte di compatibilità dalla libreria lazy allo store del core — `feature-libs/user/profile/core/facade/user-register.service.ts:L33-L43`

```ts
  protected registerCommand: Command<{ user: UserSignUp }, User> =
    this.command.create(({ user }) =>
      this.userConnector.register(user).pipe(
        tap(() => {
          // this is a compatibility mechanism only, to make anonymous consents
          // management work properly in transitional period (when we move logic
          // to separate libraries)
          this.store.dispatch(new UserActions.RegisterUserSuccess());
        })
      )
    );
```

La `Query` **dell'utente corrente —** `feature-libs/user/account/core/facade/user-account.service.ts:L24-L49`

```ts
@Injectable()
export class UserAccountService implements UserAccountFacade {
  protected userQuery: Query<User> = this.query.create(
    () =>
      this.userIdService
        .takeUserId(true)
        .pipe(switchMap((userId) => this.userAccountConnector.get(userId))),
    {
      reloadOn: [UserAccountChangedEvent],
      resetOn: [LoginEvent, LogoutEvent],
    }
  );
// ...
```

```
  /**
   * Returns the user according the userId
   * no use query for userId can change every time
   */
  getById(userId: string): Observable<User | undefined> {
    return this.userAccountConnector.get(userId);
  }
```

Update e close del profilo — feature-libs/user/profile/core/facade/user-profile.service.ts:L31-L60

```ts
  protected updateCommand: Command<{ details: User }> = this.command.create(
    (payload) =>
      this.userIdService.takeUserId(true).pipe(
        switchMap((uid) =>
          this.userProfileConnector.update(uid, payload.details).pipe(
            tap(() => {
              this.eventService.dispatch(
                { user: payload.details },
                UserAccountChangedEvent
              );
            })
          )
        )
      ),
    {
      strategy: CommandStrategy.Queue,
    }
  );

  protected closeCommand: Command = this.command.create(() =>
    this.userIdService
      .takeUserId(true)
      .pipe(
        switchMap((uid) =>
          this.userProfileConnector
            .remove(uid)
            .pipe(tap(() => this.authService.logout()))
        )
      )
  );
```

Facade lazy — feature-libs/user/profile/root/facade/user-profile.facade.ts:L14-L23

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: UserProfileFacade,
```

```
      feature: USER_PROFILE_CORE_FEATURE,
      methods: ['get', 'update', 'close', 'getTitles'],
    }),
})
export abstract class UserProfileFacade {
```

LoginGuard — core-libs/storefront/cms-components/user/login-route/login.guard.ts:L34-L65

```ts
  canActivate(
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot
  ): Observable<GuardResult> {
    return this.shouldRenderCMSPage().pipe(
      take(1),
      switchMap((shouldRenderCMSPage) => {
        if (shouldRenderCMSPage) {
          return this.cmsPageGuard.canActivate(route, state);
        } else {
          // This method can trigger redirect to OAuth server that's why we don't return anything in
this case
          const redirected = this.authService.loginWithRedirect();
          if (!redirected) {
            return of(false);
          }
          return EMPTY;
        }
      })
    );
  }

  protected shouldRenderCMSPage(): Observable<boolean> {
    return this.authService.isUserLoggedIn().pipe(
      take(1),
      map((isUserLoggedIn) => {
        return (
          this.authConfigService.getOAuthFlow() ===
            OAuthFlow.ResourceOwnerPasswordFlow || isUserLoggedIn
        );
      })
    );
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
hierarchicalAddress.countries UsingHierarchicalAddressFormat	string[]	['CN']	core-libs/core/src/user/config/hierarchical-address-config.ts:L15-L22, default core-libs/core/src/user/config/default-hierarchical-address-config.ts:L9-L13	feature-libs/user/profile/components/address-book/address-form/address-form.component.ts:L417-L420
anonymousConsents.register Consent	string	'MARKETING_NEWSLETTER'	core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L17-L20, default core-libs/core/src/anonymous-consents/config/default-anonymous-consents-config.ts:L9-L19	feature-libs/user/profile/components/register/register.component.ts:L272-L283
anonymousConsents.showLegal DescriptionInDialog	boolean	true	core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L21-L24	core-libs/storefront/shared/components/anonymous-consents-dialog/anonymous-consent-dialog.component.ts:L100-L100
anonymousConsents.required Consents	string[]	[]	core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L25-L28	core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L114-L135; core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L186-L193; core-libs/storefront/cms-components/myaccount/consent-management/consent-management-component.service.ts:L13-L24
anonymousConsents.consent ManagementPage.showAnonymous Consents	boolean	true	core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L32-L36	core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L166-L175
anonymousConsents.consent ManagementPage.hideConsents	string[]	[]	core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L37-L40	core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L177-L186
userAccount.enableUserCurrency Sync	boolean	false	feature-libs/user/account/root/config/user-account-config.ts:L14-L23, default feature-libs/user/account/root/config/default-user-account-config.ts:L9-L13	feature-libs/user/account/root/services/user-login-currency.service.ts:L38-L75; feature-libs/user/profile/root/services/user-currency-preference-saver.service.ts:L34-L57
logout.redirectRoute	string	'home'	core-libs/storefront/cms-components/user/logout/config/logout-config.ts:L14-L25, default core-libs/storefront/cms-components/user/logout/config/default-logout-config.ts:L9-L13	core-libs/storefront/cms-components/user/logout/logout.guard.ts:L91-L110 (solo con useConfigurableLogoutRedirect)

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
Endpoint OCC utente legacy (`paymentDetailsAll`, `paymentDetail`, `anonymousConsentTemplates`, `consentTemplates`, `consents`, `consentDetail`, `addresses`, `addressDetail`, `addressVerification`, `customerCoupons`, `claimCoupon`, `claimCustomerCoupon`, `couponNotification`, `notificationPreference`, `productInterests`, `getProductInterests`)	`string`	vedi file	`core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L9-L34`	adapter in `core-libs/core/src/occ/adapters/user/`
Endpoint OCC account (`user`, `createVerificationToken`)	`string`	`users/${userId}`, `users/anonymous/verificationToken`	`feature-libs/user/account/occ/adapters/config/default-occ-user-account-endpoint.config.ts:L9-L18`	`feature-libs/user/account/occ/adapters/occ-user-account.adapter.ts:L54-L79`
Endpoint OCC profile (`userRegister`, `userForgotPassword`, `userRestoreToken`, `userResetPassword`, `userUpdateLoginId`, `userUpdatePassword`, `titles`; opzionali `userUpdateProfile`, `userCloseAccount`)	`string`	vedi file (opzionali non definiti)	`feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L9-L23`, modello `feature-libs/user/profile/occ/adapters/config/occ-user-profile-endpoint.model.ts:L13-L53`	`feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L49-L174`
Endpoint OCC wishlist (`getWishlists`, …)	`string` (castati `as any`)	vedi file	`feature-libs/user/wishlist/occ/adapters/config/default-occ-user-wishlist-endpoint.config.ts:L9-L21`	`feature-libs/user/wishlist/occ/adapters/occ-user-wishlist.adapter.ts:L25-L50`
Token `USE_MY_ACCOUNT_V2_PROFILE` / `USE_MY_ACCOUNT_V2_EMAIL` / `USE_MY_ACCOUNT_V2_PASSWORD`	`InjectionToken<boolean>`	`false`	`feature-libs/user/profile/components/update-profile/use-my-account-v2-profile.ts:L9-L12`, `feature-libs/user/profile/components/update-email/use-my-account-v2-email.ts:L9-L11`, `feature-libs/user/profile/components/update-password/use-my-account-v2-password.ts:L9-L11`	`feature-libs/user/profile/components/update-profile/update-profile.module.ts:L36-L83`, `feature-libs/user/profile/components/update-email/update-email.module.ts:L91-L91`, `feature-libs/user/profile/components/update-password/update-password.module.ts:L90-L90`, `feature-libs/user/profile/components/update-password/update-password-component.service.ts:L52-L52`
Token `USE_MY_ACCOUNT_V2_CONSENT` / `USE_MY_ACCOUNT_V2_NOTIFICATION_PREFERENCE`	`InjectionToken<boolean>`	`false`	`core-libs/storefront/cms-components/myaccount/my-account-v2/use-my-account-v2-consent-notification-perference.ts:L9-L18`	`core-libs/storefront/cms-components/myaccount/consent-management/consent-management.module.ts:L26-L60`
Feature toggle `enableHierarchicalAddressFormat`	`boolean`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1081-L1081`	`feature-libs/user/profile/components/address-book/address-form/address-form.component.ts:L413-L420`, `:L499-L518`
Feature toggle `useConfigurableLogoutRedirect`	`boolean`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1089-L1089`	`core-libs/storefront/cms-components/user/logout/logout.guard.ts:L46-L110`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
Feature toggle `authorizationCodeFlowByDefault`	`boolean`	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029`	`feature-libs/user/account/components/login-form/login-form-component.service.ts:L90-L93`; `feature-libs/user/profile/core/facade/user-register.service.ts:L45-L62`; `feature-libs/user/profile/components/register/register.component.ts:L260-L270`; `feature-libs/user/account/components/guards/login-as-guest.guard.ts:L30-L44`
Feature toggle `authorizationCodeFlowByDefaultCsrfTokenRefresh`	`boolean`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1030-L1030`	`feature-libs/user/account/components/login-form/login-form-component.service.ts:L101-L131`
Feature toggle `siteIsolationForCustomLoginPage`	`boolean`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1091-L1091`	`feature-libs/user/account/components/login-form/login-form-component.service.ts:L294-L324`
Feature toggle `oauthCallbackPage` / `asyncAuthConfigInitializer`	`boolean`	`false` / `false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1111-L1111`, `:L1090-L1090`	`feature-libs/user/account/components/oauth-callback/oauth-callback.module.ts:L19-L37`; `feature-libs/user/account/root/user-account-root.module.ts:L50-L63`
`featureModules.userAccount` / `userAccountCore`	`CmsConfig`	lista componenti; core → components	`feature-libs/user/account/root/user-account-root.module.ts:L23-L48`	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L80-L100` (via `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L49`)
`featureModules.userProfile` / `userProfileCore`	`CmsConfig`	lista componenti; core → components	`feature-libs/user/profile/root/user-profile-root.module.ts:L21-L43`	idem

Chiavi di storage usate dal dominio: `spartacus⁉️⁉️anonymous-consents` (localStorage, `core-libs/core/src/state/services/state-persistence.service.ts:L124-L131`), `spartacus⁉️⁉️pre-login-currency` (localStorage, `feature-libs/user/account/root/services/user-login-currency-persistence.service.ts:L10-L37`), `cx_login_error` e `cx_otp_login_state` (`feature-libs/user/account/components/user-account-constants.ts:L7-L10`), `isGuestUserCheckout` (`core-libs/storefront/cms-components/user/utils/user-constants.ts:L7-L7`).

Per i meccanismi generali (`provideDefaultConfig`, config factory, merge) vedi il capitolo 11; per i toggle il capitolo 42.

7. Commenti e note notevoli dal sorgente

- "avoid infinite loop - if we've already attempted to load templates and we got an empty array as the response" — `core-libs/core/src/user/facade/user-consent.service.ts:L72-L72`. Spiega perché `getConsents(true)` guarda anche `success`.
- "NOTE: This is a temporary solution and the issue should be solved in the roots. Here is the ticket to track the issue: https://jira.tools.sap/browse/CXSPA-2988" — `core-libs/core/src/user/facade/user-consent.`

service.ts:L136-L137.

- "TODO: for future releases, refactor this part to thrown errors" — core-libs/core/src/user/facade/user-cost-center.service.ts:L37-L37.
- "TODO(#9416): This won't work with flow different than `Resource Owner Password Flow` which involves redirect (maybe in popup in will work)" — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L131-L131. Ammissione esplicita di un limite funzionale.
- "using the endpoint that doesn't set caching headers" — core-libs/core/src/occ/adapters/user/occ-anonymous-consent-templates.adapter.ts:L49-L49. Motiva la HEAD.
- "this is a compatibility mechanism only, to make anonymous consents management work properly in transitional period (when we move logic to separate libraries)" — feature-libs/user/profile/core/facade/user-register.service.ts:L37-L39.
- "TODO: Inline this factory when we start releasing Ivy compiled libraries" — feature-libs/user/account/root/user-account-root.module.ts:L22-L22, feature-libs/user/profile/root/user-profile-root.module.ts:L20-L20.
- "by default core is bundled together with components" — feature-libs/user/account/root/user-account-root.module.ts:L45-L45.
- "no use query for userId can change every time" — feature-libs/user/account/core/facade/user-account.service.ts:L45-L45.
- "TODO: consider dropping toLowerCase as this should not be part of the UI, as it's too opinionated and doesn't work with other AUTH services" — feature-libs/user/account/components/login-form/login-form-component.service.ts:L203-L204.
- "CXSPA-13213: refresh the CSRF token immediately before the native form submit. The auth server rotates per-request, so a stale page-load token would 403." — feature-libs/user/account/components/login-form/login-form-component.service.ts:L102-L109.
- "Submit BEFORE flipping busy$ to true … the browser excludes disabled inputs from native form submissions — resulting in an empty POST body and a 403." — feature-libs/user/account/components/login-form/login-form-component.service.ts:L122-L125. Una lezione di DOM poco ovvia.
- Lungo commento sul perché usare `window.location.href` e `sessionStorage` invece del router e dei query param — feature-libs/user/account/components/login-form/login-form-component.service.ts:L136-L158.
- "TODO(#9638): Use logout route when it will support passing redirect url" — feature-libs/user/profile/components/update-email/update-email-component.service.ts:L93-L93, feature-libs/user/profile/components/update-password/update-password-component.service.ts:L108-L108.
- "This method can trigger redirect to OAuth server that's why we don't return anything in this case" — core-libs/storefront/cms-components/user/login-route/login.guard.ts:L44-L44.
- `@deprecated since 221121.21` sul parametro `cms` di `LogoutGuard`, rimozione "in the major release of 2028" — core-libs/storefront/cms-components/user/logout/logout.guard.ts:L37-L39.
- "TODO(#9385): Use CMS page guard here." — core-libs/storefront/cms-components/user/logout/logout.guard.ts:L63-L63, :L79-L79.
- JSDoc di `RegisterComponentService.getAdditionalConsents`: "In integration scenarios, eg: cdc, this method will be overridden" — feature-libs/user/profile/components/register/register-component.service.ts:L53-L58.
- JSDoc `UserIdService`: implementazione OCC-specifica, "provide your own implementation" — core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L15-L20.
- "'default' is only valid for mutations (POST/DELETE), not for GET entries." — feature-libs/user/wishlist/occ/adapters/occ-user-wishlist.adapter.ts:L43-L44.

- Commento nell'app: `LogoutModule` e `LoginRouteModule` "will be come part of auth package" — `projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L180-L181`.

8. Extension point

EXTENSION POINT	COME SI ESTENDE	RIFERIMENTO	RICETTA
Adapter core (`UserAddressAdapter`, `UserConsentAdapter`, `UserPaymentAdapter`, `CustomerCouponAdapter`, `UserInterestsAdapter`, `UserCostCenterAdapter`)	fornire una propria classe al posto dell'adapter OCC (binding `useClass` in `UserOccModule`)	`core-libs/core/src/user/connectors/address/user-address.adapter.ts:L10-L27`, `core-libs/core/src/occ/adapters/user/user-occ.module.ts:L34-L72`	Ricette OCC
Token converter (`ADDRESS_NORMALIZER`, `ADDRESS_SERIALIZER`, `ADDRESS_VALIDATION_NORMALIZER`, ..., `ANONYMOUS_CONSENT_NORMALIZER`)	converter multi aggiuntivi	`core-libs/core/src/user/connectors/address/converters.ts:L12-L26`	Ricette OCC
Endpoint OCC (core, account, profile; opzionali `userUpdateProfile`, `userCloseAccount`)	`provideConfig` con `backend.occ.endpoints`	`core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L9-L34`, `feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L49-L60`	Ricette OCC
Facade core `providedIn: 'root'` (`UserAddressService`, `UserConsentService`, `ConsentService`, `AnonymousConsentsService`, ...)	sostituzione via DI (`{ provide: X, useClass: MyX }`)	`core-libs/core/src/user/facade/consent.service.ts:L14-L35`, `core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L46-L65`	Ricette config/DI
`UserIdService`	"provide your own implementation"	`core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L15-L24`	Ricette config/DI
Facade lazy (`UserAccountFacade`, `UserProfileFacade`, `UserRegisterFacade`, `UserEmailFacade`, `UserPasswordFacade`, `VerificationTokenFacade`)	sostituire l'implementazione nel core lazy (`useExisting`) o consumarle senza importare il core	`feature-libs/user/account/core/facade/facade-providers.ts:L15-L26`, `feature-libs/user/profile/core/facade/facade-providers.ts:L18-L39`	Ricette feature lazy
Adapter lazy (`UserAccountAdapter`, `UserProfileAdapter`) + token (`USER_ACCOUNT_NORMALIZER`, `LOGIN_FORM_SERIALIZER`, `USER_SIGN_UP_SERIALIZER`, `TITLE_NORMALIZER`, ...)	come per il core	`feature-libs/user/account/core/connectors/converters.ts:L15-L29`, `feature-libs/user/profile/core/connectors/user-profile.adapter.ts:L11-L42`, `feature-libs/user/profile/core/connectors/converters.ts:L12-L30`	Ricette OCC

EXTENSION POINT	COME SI ESTENDE	RIFERIMENTO	RICETTA
Modello `User` aumentabile	module augmentation di `@spartacus/user/account/root`	feature-libs/user/account/root/model/user.model.ts:L7-L15, feature-libs/user/profile/root/model/user-profile.model.ts:L9-L26	Ricette OCC
Eventi (`UserAccountChangedEvent`, `AddUserAddressEvent`, `UpdateUserAddressEvent`, `DeleteUserAddressEvent`, `LoginEvent`, `LogoutEvent`)	`EventService.get(...)`	feature-libs/user/account/root/events/user-account.events.ts:L10-L16, core-libs/core/src/user/events/user-event.builder.ts:L25-L53	Ricette eventi
`RegisterComponentService` (`getAdditionalConsents`, `generateAdditionalConsentsFormControl`, `collectDataFromRegisterForm`)	sostituzione del servizio (lo fa l'integrazione CDC)	feature-libs/user/profile/components/register/register-component.service.ts:L53-L85	Ricette CMS
Servizi di componente (`LoginFormComponentService`, `UpdateProfileComponentService`)	`providers` nella `CmsConfig` del componente	feature-libs/user/account/components/login-form/login-form.module.ts:L50-L64, feature-libs/user/profile/components/update-profile/update-profile.module.ts:L36-L83	Ricette CMS
`ConsentManagementComponentService.getRequiredConsents`	override (parametro pensato per CDC)	core-libs/storefront/cms-components/myaccount/consent-management/consent-management-component.service.ts:L13-L24	Ricette CMS
Mapping CMS (`ReturningCustomerLoginComponent`, `RegisterCustomerComponent`, `ConsentManagementComponent`, `AnonymousConsentManagementBannerComponent`, …)	`provideConfig({ cmsComponents: ... })`	feature-libs/user/account/root/user-account-root.module.ts:L23-L48, core-libs/storefront/cms-components/anonymous-consent-management/anonymous-consent-management.module.ts:L32-L42	Ricette CMS
Token `USE_MY_ACCOUNT_V2_*`	`{ provide: USE_MY_ACCOUNT_V2_PROFILE, useValue: true }`	projects/storefrontapp/src/app/spartacus/features/user/user-feature.module.ts:L79-L90	Ricette config/DI
`LogoutConfig`, `LogoutGuard`, `LoginGuard`	config / sostituzione guard	core-libs/storefront/cms-components/user/logout/config/logout-config.ts:L14-L25, core-libs/storefront/cms-components/user/logout/logout.guard.ts:L71-L110	Ricette route

EXTENSION POINT	COME SI ESTENDE	RIFERIMENTO	RICETTA
Config `HierarchicalAddressConfig`, `AnonymousConsentsConfig`, `UserAccountConfig`	`provideConfig`	`core-libs/core/src/user/config/hierarchical-address-config.ts:L10-L27`, `core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L11-L43`, `feature-libs/user/account/root/config/user-account-config.ts:L14-L23`	Ricette config/DI

Consigli di upgrade-safety (vedi capitolo 54): preferire le facade astratte di `@spartacus/user/*/root` e gli eventi agli internals dello store `user`, i cui reducer ed effect **non** sono esportati dal barrel — `core-libs/core/src/user/index.ts:L7-L15`.

> Nota: `UserNotificationPreferenceAdapter` e il suo connector sono esportati solo dal barrel locale `core-libs/core/src/user/connectors/notification-preference/index.ts:L8-L9`, che a sua volta non è riesportato da `core-libs/core/src/user/connectors/index.ts:L7-L12`; fuori da `core-libs/core/src/user/` gli unici import sono relativi (`core-libs/core/src/occ/adapters/user/user-occ.module.ts:L19-L19`, `core-libs/core/src/occ/adapters/user/occ-user-notification-preference.adapter.ts:L19-L20`). Quindi il simbolo non è importabile da `@spartacus/core` e il token astratto, registrato con `useClass: OccUserNotificationPreferenceAdapter` (`core-libs/core/src/occ/adapters/user/user-occ.module.ts:L50-L51`), non è sovrascrivibile tramite API pubblica (è esportata solo la classe OCC, `core-libs/core/src/occ/adapters/user/index.ts:L13-L13`).

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Mutazioni indirizzo e loader condiviso.** Add/Update/Delete usano il loader `USER_ADDRESSES`: durante una mutazione `getAddressesLoading()` è `true`, e un update fallito imposta `error: true` sull'intera lista, mentre il valore resta perché il sub-reducer non gestisce quei fail — `core-libs/core/src/user/store/actions/user-addresses.action.ts:L56-L137`, `core-libs/core/src/state/utils/loader/loader.reducer.ts:L45-L52`, `core-libs/core/src/user/store/reducers/user-addresses.reducer.ts:L16-L25`.
2. **Dispatch dentro** `map`**.** `getRegions/getCities/getDistricts` dispatchano clear/load nel `map` di un selettore: ogni sottoscrittore in più può ripeterli, e durante il caricamento ritornano `[]` (non il valore precedente) — `core-libs/core/src/user/facade/user-address.service.ts:L188-L280`.
3. `consentId` **ignorato.** L'adapter OCC ignora il `consentId` di `withdrawConsent`, che il componente consensi valorizza con `template.id` — `core-libs/core/src/occ/adapters/user/occ-user-consent.adapter.ts:L69-L78`, `core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L239-L243`.
4. **Withdraw senza reducer.** Il reducer dei consensi non ha un case per il withdraw success: lo stato resta "dato" finché qualcuno non chiama `loadConsents()` (la pagina lo fa) — `core-libs/core/src/user/store/reducers/user-consents.reducer.ts:L16-L33`, `core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L203-L222`. Se costruisci una UI tua, devi ricaricare.
5. **Trasferimento consensi solo dopo una registrazione nella stessa sessione app.** `withLatestFrom` su `REGISTER_USER_SUCCESS`: un login senza registrazione precedente non trasferisce nulla; il TODO dichiara che con flow a redirect non funziona — `core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L131-L146`.

6. **Code flow di default e trasferimento.** `authorizationCodeFlowByDefault` è `true` e dopo la registrazione si va a `login` (redirect al server OAuth) — feature-libs/user/profile/components/register/register.component.ts:L260-L262, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029. Combinato con il punto 5, il trasferimento potrebbe non avvenire. DA VERIFICARE a runtime: dipende da se lo store sopravvive al redirect (dossier T6/Q3).
7. `templates: {}` **dopo logout/cambio lingua.** Il meta-reducer anonimo non usa lo stato iniziale del loader: i flag loading/success/error diventano `undefined` — core-libs/core/src/anonymous-consents/store/reducers/index.ts:L58-L61.
8. **Una HEAD per ogni emissione del banner.** `isBannerVisible()` dispatcha il check versioni a ogni emissione del `combineLatest`, e sia il banner sia il link "open dialog" creano uno stream — core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L291-L299, core-libs/storefront/cms-components/anonymous-consent-management/banner/anonymous-consent-management-banner.component.ts:L23-L24, core-libs/storefront/cms-components/anonymous-consent-management/open-dialog/anonymous-consent-open-dialog.component.ts:L27-L28.
9. `giveAllConsents()` **è lazy e ripetitivo.** Agisce solo se sottoscritto e ri-dispatcha a ogni nuova emissione dei template finché la sottoscrizione è viva — core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L169-L215, core-libs/storefront/cms-components/anonymous-consent-management/banner/anonymous-consent-management-banner.component.ts:L44-L51.
10. **Confronti sensibili all'ordine.** `consentsUpdated` confronta stringhe JSON; `detectUpdatedTemplates` confronta per indice: un riordino è un "aggiornamento" — core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L316-L333, :L369-L376.
11. **Mutazione in place.** `giveRequiredConsents` copia l'array ma modifica gli oggetti (`consent.consentState = GIVEN`) — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L117-L131.
12. `includes` **invece di** `startsWith`**.** L'interceptor considera OCC ogni URL che *contiene* il base URL — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L110-L112.
13. **Captcha "enabled" sempre vero.** In `appendCaptchaToken` la variabile `isCaptchaEnabled` riceve il valore di ritorno di `.subscribe(...)`, una `Subscription` sempre truthy: conta solo `provider.getToken()` — feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L187-L203.
14. **Doppio logout alla chiusura account.** `closeCommand` chiama `authService.logout()` e il modal chiama anche `coreLogout()` — feature-libs/user/profile/core/facade/user-profile.service.ts:L55-L57, feature-libs/user/profile/components/close-account/components/close-account-modal/close-account-modal.component.ts:L97-L99.
15. **Evento con dati parziali.** `UserAccountChangedEvent` porta solo i dettagli inviati, non l'utente completo — feature-libs/user/profile/core/facade/user-profile.service.ts:L36-L41.
16. **Sync valuta "a ping-pong".** Con `enableUserCurrencySync` il login applica la valuta utente con `setActive`, che genera `CurrencySetEvent`, e `UserCurrencyPreferenceSaverService` reagisce con una PATCH della stessa valuta — feature-libs/user/account/root/services/user-login-currency.service.ts:L38-L75, core-libs/core/src/site-context/events/site-context-event.builder.ts:L58-L71, feature-libs/user/profile/root/services/user-currency-preference-saver.service.ts:L34-L57. DA VERIFICARE: catena dedotta dal codice, non eseguita.
17. **Password nello** `history.state`**.** Nel login OTP la password in chiaro viaggia nello state di navigazione verso `verifyToken` ed è sovrascritta solo dal componente di arrivo — feature-libs/user/account/components/otp-login-form/otp-login-form.component.ts:L132-L173, feature-libs/user/account/components/verification-token-form/verification-token-form.component.ts:L110-L124.
18. `history` **globale in SSR.** `VerificationTokenFormComponent.ngOnInit` usa `history` direttamente, non `WindowRef` — feature-libs/user/account/components/verification-token-form/verification-token-

form.component.ts:L110-L124. DA VERIFICARE il comportamento in SSR (in Node `history` non esiste; dipende da come la pagina viene renderizzata lato server, vedi capitolo 7).

19. **Coupon claim e navigazione immediata.** `CouponClaimComponent` sottoscrive un risultato che può emettere subito `false` e navigare a `coupons` prima del successo — core-libs/storefront/cms-components/myaccount/my-coupons/coupon-claim/coupon-claim.component.ts:L29-L52. DA VERIFICARE a runtime se il messaggio di successo vada perso.
20. **Eventi "comando" non legati allo store.** `LoadUserAddressesEvent` e `LoadUserPaymentMethodsEvent` (core-libs/core/src/user/events/user.events.ts:L30-L36) non sono mappati su azioni dal `UserEventBuilder` (core-libs/core/src/user/events/user-event.builder.ts:L25-L53). Sono eventi di richiesta: li emette il checkout dopo aver creato un indirizzo o un metodo di pagamento (feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L90-L92, feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L49-L51). `CheckoutLegacyStoreEventListener` li traduce in `LoadUserAddresses`/`LoadUserPaymentMethods` (feature-libs/checkout/base/root/events/checkout-legacy-store-event.listener.ts:L38-L66). Senza la feature checkout nessuno li emette né li ascolta.
21. **Eventi sulla richiesta, non sul success.** `UpdateUserAddressEvent` & co. sono legati a `UPDATE_USER_ADDRESS`: arrivano anche se la chiamata poi fallisce — core-libs/core/src/user/events/user-event.builder.ts:L25-L53.
22. **ASM e consensi.** `UserConsentService.getConsent` richiede `userId === 'current'`: in emulazione ASM (userId = customerId) non emette — core-libs/core/src/user/facade/user-consent.service.ts:L128-L154, core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L24-L48 (vedi capitolo 29).
23. `toLowerCase()` **opinionato.** Il login ROPC e la registrazione abbassano username/email — feature-libs/user/account/components/login-form/login-form-component.service.ts:L198-L213, feature-libs/user/profile/components/register/register-component.service.ts:L53-L85.
24. **Refusi nell'API pubblica** (non "correggibili" senza breaking change): `getProdutInterestsLoading`, `removeProdutInterest`, `ToggleAnonymousConsentsBannerDissmissed`, `OneTimePasswordLoginFormModeule`, il file `use-my-account-v2-consent-notification-perference.ts` — core-libs/core/src/user/facade/user-interests.service.ts:L81-L127, core-libs/core/src/anonymous-consents/store/actions/anonymous-consents.action.ts:L91-L113, feature-libs/user/account/components/user-account-component.module.ts:L16-L27. Anche le costanti action degli interessi non hanno il prefisso `[User]` e quelle del claim coupon sono troncate — core-libs/core/src/user/store/actions/product-interests.actions.ts:L32-L48, core-libs/core/src/user/store/actions/customer-coupon.action.ts:L58-L66.
25. `UserComponentModule` **inutilizzato dall'app demo**, che importa direttamente `LogoutModule` e `LoginRouteModule` — core-libs/storefront/cms-components/user/user.module.ts:L14-L23, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L180-L181.
26. **Wishlist senza modulo.** `@spartacus/user/wishlist` non registra provider: oggi lo fa il modulo `wish-list-v2` di cart — feature-libs/user/wishlist/public_api.ts:L7-L9, feature-libs/cart/wish-list/wish-list-v2.module.ts:L36-L46 (vedi capitolo 25). L'adapter, se non trova wishlist, ritorna `{ entries: [] }` — feature-libs/user/wishlist/occ/adapters/occ-user-wishlist.adapter.ts:L25-L50.
27. `getById` **senza cache.** Ogni chiamata è una GET — feature-libs/user/account/core/facade/user-account.service.ts:L43-L49.

---

10. Come fare debug

- **Quale "utente" stai guardando?** Prima di tutto chiediti se il dato vive nello store NgRx `user` (indirizzi, consensi registrati, pagamenti, coupon, ...) — core-libs/core/src/user/store/user-state.ts:L52-L65 — oppure in una `Query` di `@spartacus/user` (dettagli account, titoli) — feature-libs/user/account/core/facade/

user-account.service.ts:L26-L35, feature-libs/user/profile/core/facade/user-profile.service.ts:L62-L67. Solo il primo si vede nello state tree.

- **State tree.** Cerca le chiavi `user` (core-libs/core/src/user/store/user-state.ts:L22-L46), `anonymous-consents` (core-libs/core/src/anonymous-consents/store/anonymous-consents-state.ts:L10-L24) e `process` per gli esiti di give/withdraw consent (core-libs/core/src/user/store/actions/user-consents.action.ts:L71-L83). Le action hanno type parlanti (`'[User] …'`), con le eccezioni del punto 24 della sezione 9.
- **Consensi anonimi.** Ispeziona in `localStorage` la chiave `spartacus⁓⁓anonymous-consents` — core-libs/core/src/state/services/state-persistence.service.ts:L124-L131 — e nel tab Network l'header `X-Anonymous-Consents` sulle richieste OCC e sulla risposta di `users/anonymous/consenttemplates` — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L51-L108. Le richieste `HEAD` sullo stesso endpoint sono il check versioni — core-libs/core/src/occ/adapters/user/occ-anonymous-consent-templates.adapter.ts:L48-L62. Se in dev mode vedi il warning "No consents were loaded", il backend non restituisce l'header — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L49-L55.
- **Feature lazy non caricata.** Se un componente come `ReturningCustomerLoginComponent` non appare, verifica che sia nella lista CMS della feature (feature-libs/user/account/root/user-account-root.module.ts:L23-L48) e che l'app abbia mappato `USER_ACCOUNT_FEATURE` sul wrapper module (projects/storefrontapp/src/app/spartacus/features/user/user-feature.module.ts:L39-L78). Per `OAuthCallbackComponent` controlla i tre toggle — feature-libs/user/account/root/user-account-root.module.ts:L50-L63. Meccanismo nel capitolo 12.
- **La query utente non si aggiorna.** Deve passare un `UserAccountChangedEvent` (`reloadOn`) oppure un `LoginEvent`/`LogoutEvent` (`resetOn`) — feature-libs/user/account/core/facade/user-account.service.ts:L26-L35. Puoi sottoscrivere `EventService.get(UserAccountChangedEvent)` per vedere se viene emesso (feature-libs/user/profile/core/facade/user-profile.service.ts:L36-L41).
- **Login form che dà 403 in code flow.** Ricordati del CSRF ruotato per richiesta e del submit prima di `busy$` — feature-libs/user/account/components/login-form/login-form-component.service.ts:L101-L131; in caso di errore cerca `cx_login_error` in `sessionStorage` — feature-libs/user/account/components/login-form/login-form-component.service.ts:L133-L166.
- **Regioni che "spariscono".** Il cambio lingua/valuta emette `ClearUserMiscsData` e resetta paesi, regioni, città e distretti — core-libs/core/src/user/store/effects/clear-miscs-data.effect.ts:L17-L27, core-libs/core/src/user/store/effects/regions.effect.ts:L50-L57.
- **Consensi utente "fermi".** Dopo un cambio lingua il loader dei consensi è resettato (core-libs/core/src/user/store/effects/user-consents.effect.ts:L23-L29); dopo un withdraw serve un reload (sezione 9, punto 4).
- **Test.** I test di `@spartacus/user` girano con Vitest in `jsdom` — feature-libs/user/vitest.config.ts:L33-L35; comando e linee guida nel capitolo 44.

---

11. Domande di autoverifica

1. **Dove sta `UserIdService` e quali valori emette?** In `core-libs/core/src/auth/user-auth/facade/`, non nel modulo user; emette `'current'`, `'anonymous'` o il `customerId` in emulazione ASM — core-libs/core/src/user/facade/user-address.service.ts:L11-L11, core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L24-L48.
2. **Lo stato NgRx `user` contiene i dati anagrafici del cliente loggato?** No: contiene indirizzi, consensi, paesi, pagamenti, coupon, notifiche, interessi, cost center. I dettagli account sono in una `Query` di

UserAccountService — core-libs/core/src/user/store/user-state.ts:L52-L65, feature-libs/user/account/core/facade/user-account.service.ts:L26-L35.

3. **Perché** `verifyAddress` **funziona anche per un utente anonimo?** Perché è un `Command` fuori store che usa `takeUserId(false)`, e l'adapter OCC aggiunge `USE_CLIENT_TOKEN` per lo userId anonimo — core-libs/core/src/user/facade/user-address.service.ts:L301-L318, core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L96-L105.
4. **Come arrivano al backend i consensi di un visitatore anonimo?** Tramite l'header `X-Anonymous-Consents` (JSON + `encodeURIComponent`) che `AnonymousConsentsInterceptor` aggiunge a ogni richiesta OCC — core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L93-L108, core-libs/core/src/model/consent.model.ts:L21-L32.
5. **In quali condizioni i consensi anonimi vengono trasferiti all'utente dopo il login?** Su `AuthActions.LOGIN`, se esiste la config `anonymousConsents` e se in precedenza è stato visto `REGISTER_USER_SUCCESS`; vengono trasferiti solo i consensi `GIVEN` non obbligatori — core-libs/core/src/anonymous-consents/store/effects/anonymous-consents.effect.ts:L133-L184.
6. **Chi dispatcha** `RegisterUserSuccess`**, visto che la registrazione è in una libreria senza NgRx?** `UserRegisterService` di `@spartacus/user/profile`, nel `tap` del `registerCommand`, come meccanismo di compatibilità — feature-libs/user/profile/core/facade/user-register.service.ts:L33-L43.
7. **Dopo** `UserProfileFacade.update()`**, come si aggiorna il nome nel mini-login?** `updateCommand` dispatcha `UserAccountChangedEvent`; la `Query` di `UserAccountService` ha `reloadOn: [UserAccountChangedEvent]` e rifà la GET; `LoginComponent` legge `userAccount.get()` — feature-libs/user/profile/core/facade/user-profile.service.ts:L31-L48, feature-libs/user/account/core/facade/user-account.service.ts:L26-L35, feature-libs/user/account/components/login/login.component.ts:L45-L62.
8. **Cosa succede allo store** `user` **e ai consensi anonimi al logout?** `clearUserState` azzera tutto lo stato `user`; il meta-reducer anonimo azzera solo `templates`, conservando consensi e flag UI — core-libs/core/src/user/store/reducers/index.ts:L99-L111, core-libs/core/src/anonymous-consents/store/reducers/index.ts:L49-L65.
9. **Quando** `LoginGuard` **mostra la pagina CMS di login invece di redirigere?** Quando il flow è Resource Owner Password o l'utente è già loggato; altrimenti chiama `loginWithRedirect()` e ritorna `EMPTY` — core-libs/storefront/cms-components/user/login-route/login.guard.ts:L34-L65.
10. **Perché dopo aver ritirato un consenso con una UI custom lo stato può risultare ancora "dato"?** Perché il reducer dei consensi gestisce solo il give success; il withdraw richiede un `loadConsents()` esplicito, come fa `ConsentManagementComponent` — core-libs/core/src/user/store/reducers/user-consents.reducer.ts:L16-L33, core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L203-L222.

TAVOLA 23.1 **4.1 Architettura: dove vive ogni pezzo**

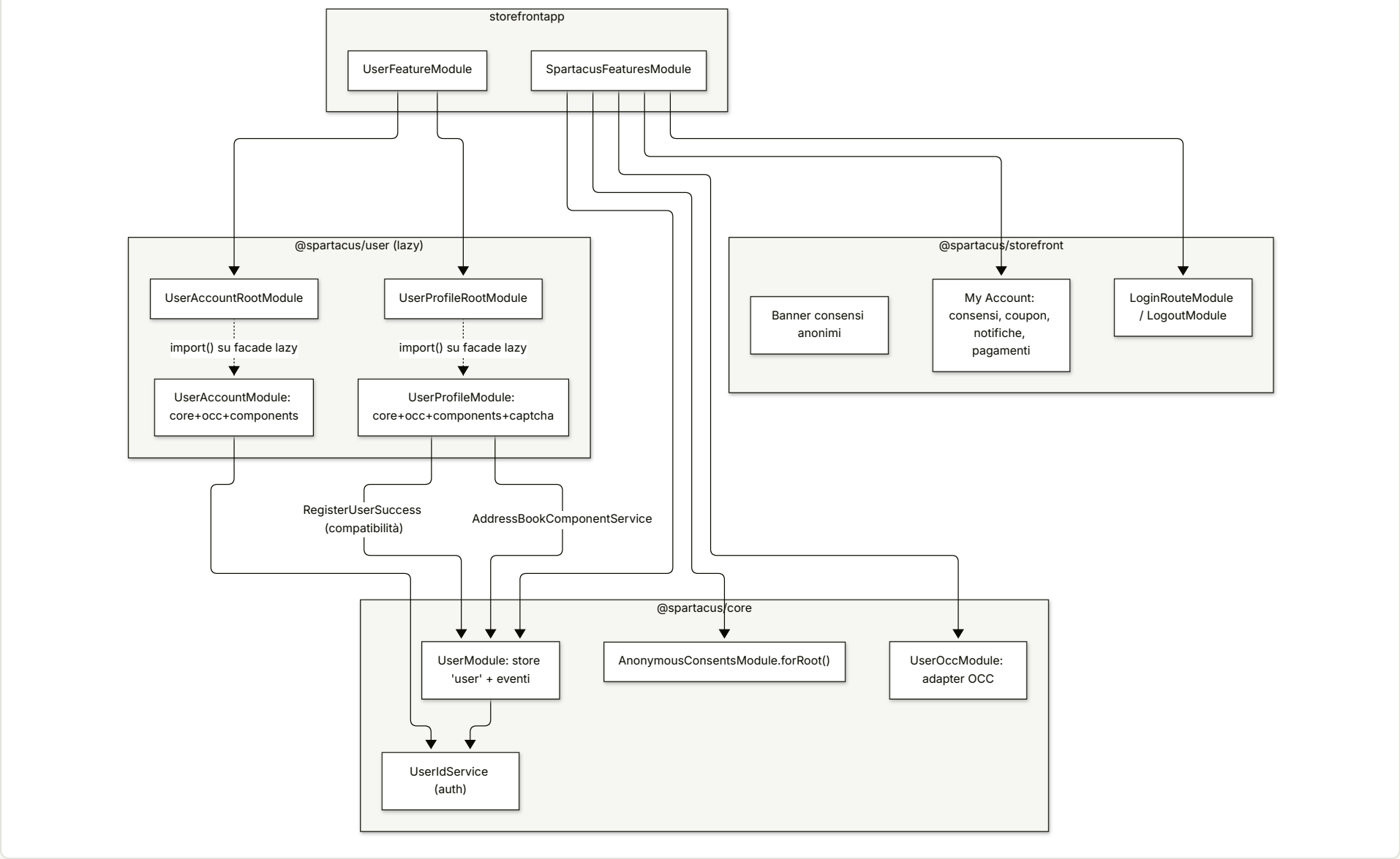


TAVOLA 23.2 **4.2 Consensi anonimi: avvio, banner, header HTTP**

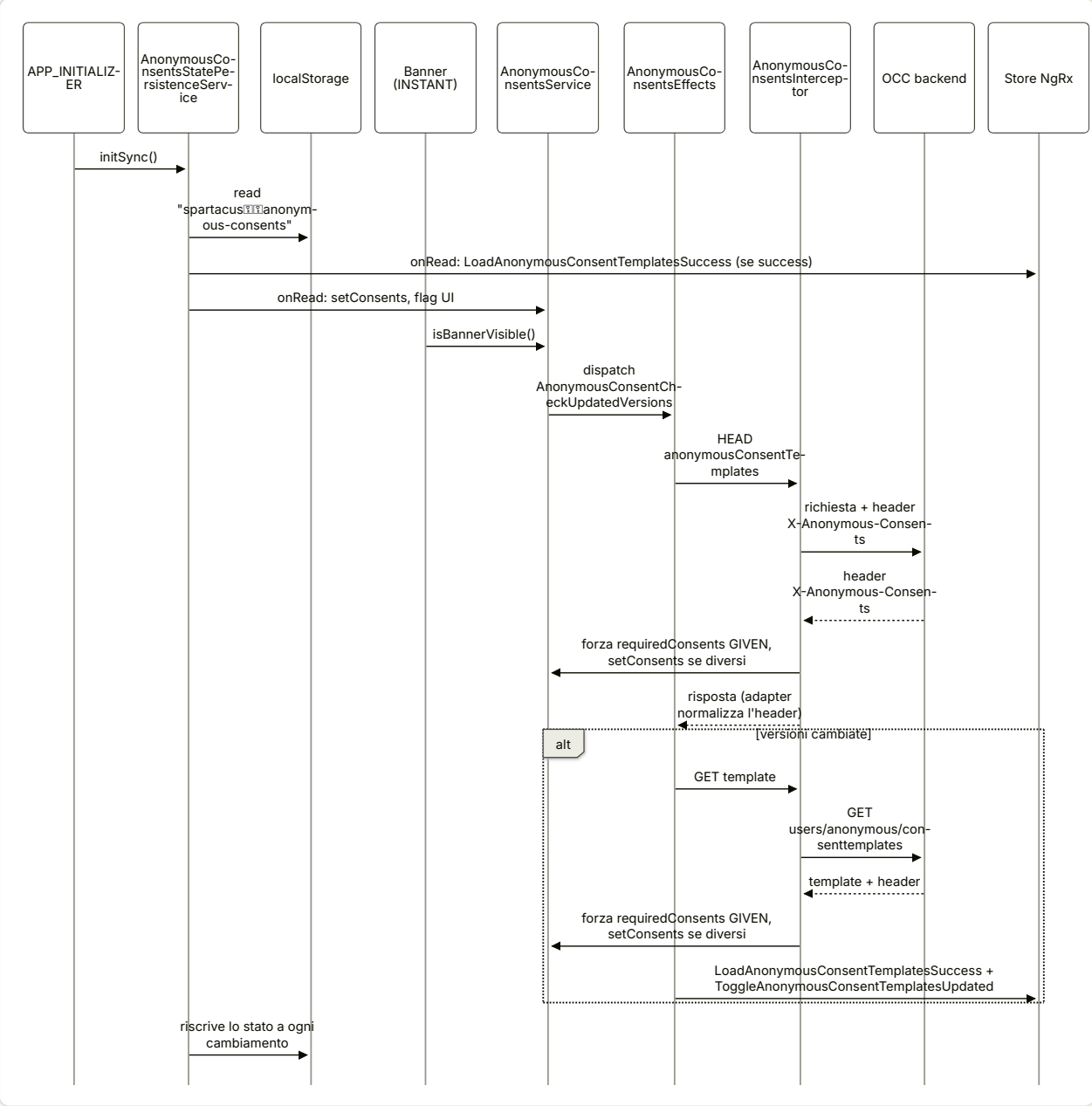
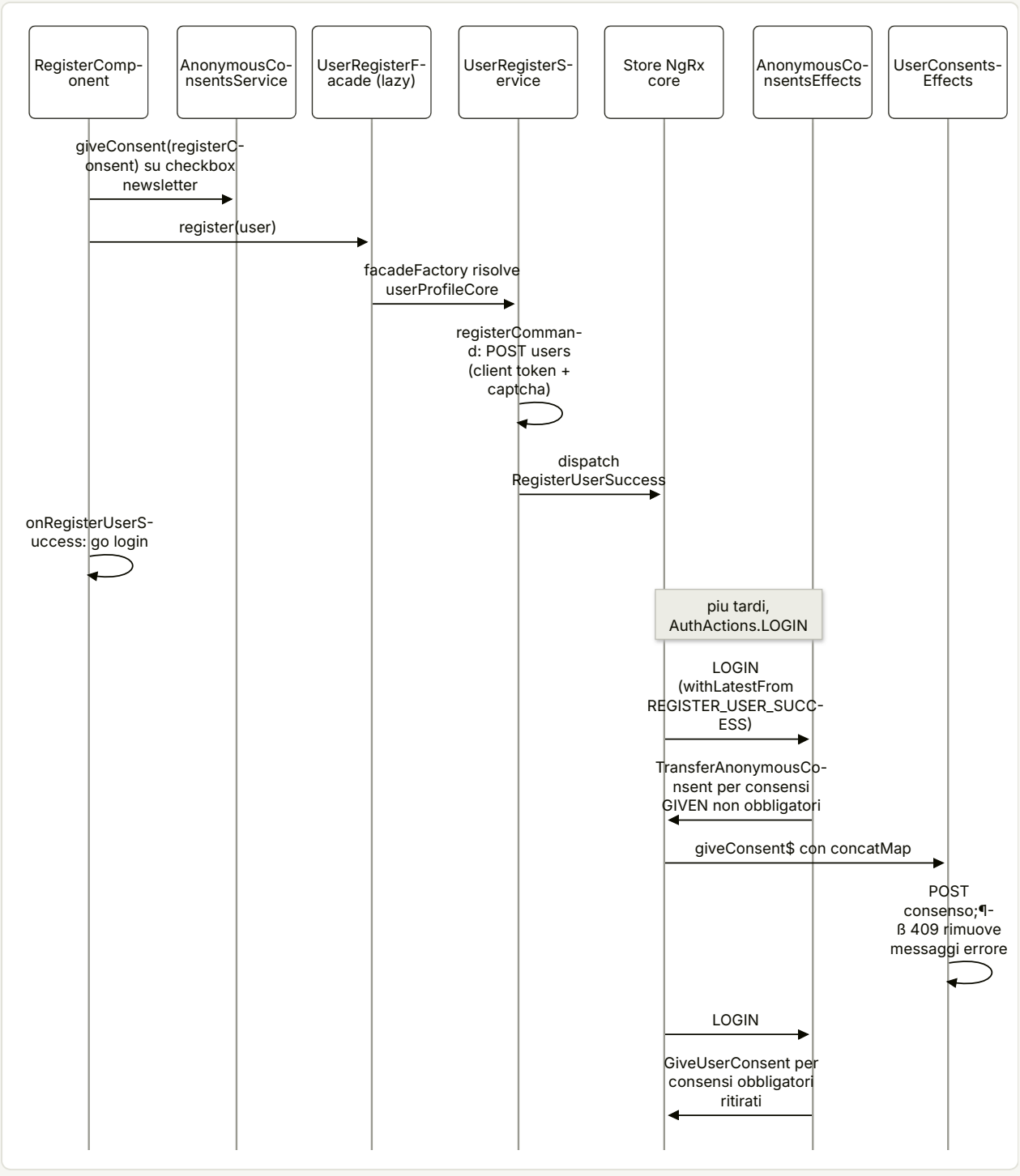


TAVOLA 23.3 **4.3 Registrazione e trasferimento dei consensi anonimi**



24

Prodotto e ricerca

Caricamento "a pezzi" del prodotto con gli scope, ricerca PLP e search box.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Il dominio "prodotto" è il cuore di qualsiasi storefront: la pagina di dettaglio (PDP), la pagina di elenco/ricerca (PLP), il search box con autocompletamento, i carousel, le recensioni e i prodotti correlati leggono tutti dati di prodotto. In Spartacus questo dominio è distribuito su tre livelli:

1. **Core non-UI** (`core-libs/core/src/product`): facade (`ProductService`, `ProductSearchService`, `SearchboxService`, `ProductSearchByCodeService`, `ProductReviewService`, `ProductReferenceService`, `ProductAvailabilityService`), un servizio di orchestrazione dei caricamenti per *scope* (`ProductLoadingService`), lo slice NgRx `product`, connector/adapter astratti, page meta resolver ed eventi. `ProductModule` importa store ed eventi; solo `forRoot()` aggiunge i quattro `PageMetaResolver` (prodotto, categoria, ricerca, coupon-search) (`core-libs/core/src/product/product.module.ts:L16-L49`).
2. **Implementazione OCC** (`core-libs/core/src/occ/adapters/product`): `ProductOccModule` lega gli adapter astratti alle classi OCC e registra i normalizer multi-provider (`core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L76`).
3. **UI** (`core-libs/storefront/cms-components/product` e `cms-pages`): componenti CMS di PDP e PLP, facet navigation, carousel, references; più le feature lib opzionali `@spartacus/product` (variants, bulk-pricing, future-stock, image-zoom) e `@spartacus/product-multi-dimensional`.

L'app demo mette insieme tutto importando `ProductModule.forRoot()` + `ProductOccModule` ("Product Core") e i moduli UI di prodotto (`projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L219-L234`).

Il problema di fondo che questo codice risolve è **caricare un prodotto "a pezzi"**. Un prodotto OCC ha decine di campi; una lista ne serve pochi (nome, prezzo, immagine), la PDP molti di più, la tab "specifiche" solo le `classifications`. Spartacus introduce quindi gli **scope** (`ProductScope`: `list`, `details`, `attributes`, `variants`, `price`, `stock`, …) (`core-libs/core/src/product/model/product-scope.ts:L7-L21`): ogni scope ha un proprio stato nello store, un proprio endpoint/`fields` OCC, e può essere invalidato separatamente. Sopra gli scope, una serie di ottimizzazioni (buffer delle azioni simultanee, fusione delle richieste che differiscono solo per `fields`) evita che "caricare a pezzi" significhi "fare N richieste HTTP".

La ricerca segue un modello diverso: non è un'entità indicizzata per codice ma un singolo risultato corrente, con due "canali" separati — risultati di pagina (PLP) e risultati ausiliari (search box) — più i suggerimenti (`core-libs/core/src/product/store/product-state.ts:L35-L39`). La PLP è **guidata dall'URL**: facet, ordinamento e paginazione sono navigazioni, e la ricerca riparte dal cambio di route.

Capitoli collegati: lo state management generico (loader/entity/scoped) è nel 19; facade ed eventi nel 20; endpoint, converter e `OccRequestsOptimizerService` nel 21; il caricamento lazy delle feature nel 12; gli outlet nel 17; il rendering CMS nel 16; il trasferimento di stato SSR nel 07.

2. Mappa dei file

PATH	RUOLO
`core-libs/core/src/product/product.module.ts`	`ProductModule` (+`forRoot` con 4 `PageMetaResolver`)

PATH	RUOLO
core-libs/core/src/product/model/product-scope.ts	enum ProductScope
core-libs/core/src/product/model/search-config.ts	SearchConfig (pageSize, currentPage, sort, filters, query)
core-libs/core/src/product/facade/product.service.ts	ProductService: get per scope, stati loader
core-libs/core/src/product/services/product-loading.service.ts	ProductLoadingService: espansione scope, cache observable, maxAge, reloadOn
core-libs/core/src/product/services/product-catalog.service.ts	ProductCatalogService.isProductInCatalog
core-libs/core/src/product/services/*-page-meta.resolver.ts	resolver SEO di prodotto, categoria, ricerca, coupon
core-libs/core/src/product/facade/product-search.service.ts	ProductSearchService (risultati PLP)
core-libs/core/src/product/facade/searchbox.service.ts	SearchboxService (risultati ausiliari + suggerimenti)
core-libs/core/src/product/facade/product-search-by-code.service.ts	load per codice (carousel)
core-libs/core/src/product/facade/product-search-by-category.service.ts	load per categoria
core-libs/core/src/product/facade/product-review.service.ts, product-reference.service.ts	reviews e references
core-libs/core/src/product/facade/product-availability.service.ts	stock in tempo reale via CommandService
core-libs/core/src/product/store/product-state.ts	shape dello slice product e costanti entità
core-libs/core/src/product/store/product-store.module.ts	feature NgRx + config SSR transfer
core-libs/core/src/product/store/actions/*.ts	azioni product, search, by-code, by-category, reviews, references
core-libs/core/src/product/store/effects/*.ts	ProductEffects, ProductsSearchEffects, ProductSearchByCodeEffects, ProductSearchByCategoryEffects, ProductReviewsEffects, ProductReferencesEffects
core-libs/core/src/product/store/reducers/*.ts	mappa reducer, meta-reducer clearProductsState, search/reviews/references
core-libs/core/src/product/store/selectors/*.ts	selector per entità/scope, search, reviews, references
core-libs/core/src/product/connectors/**	ProductAdapter, ProductConnector, ProductSearchAdapter, ProductReviewsAdapter, ProductReferencesAdapter, ProductAvailabilityAdapter, token di conversione
core-libs/core/src/product/event/*.ts	FacetChangedEvent e ProductEventBuilder
core-libs/core/src/occ/adapters/product/product-occ.module.ts	binding OCC + normalizer
core-libs/core/src/occ/adapters/product/default-occ-product-config.ts	endpoint prodotto/ricerca e loadingScopes.product

PATH	RUOLO
core-libs/core/src/occ/adapters/product/occ-product.adapter.ts	OccProductAdapter (load, loadMany)
core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts	ricerca, by codes (chunk 100), by category, suggerimenti, keyword redirect
core-libs/core/src/occ/adapters/product/converters/*.ts	ProductImageNormalizer, ProductNameNormalizer, OccProductSearchPageNormalizer, OccProductReferencesListNormalizer
core-libs/core/src/occ/services/loading-scopes.service.ts	LoadingScopesService (expand, getMaxAge, getReloadTriggers)
core-libs/storefront/cms-components/product/current-product.service.ts	CurrentProductService: prodotto dalla route
core-libs/storefront/cms-components/product/product-outlets.model.ts	ProductDetailOutlets, ProductListOutlets
core-libs/storefront/cms-components/product/product-summary/, product-intro/, product-images/, product-tabs/, stock-notification/	componenti PDP
core-libs/storefront/cms-components/product/product-list/	ProductListModule, ProductListComponent, ProductListComponentService, scroll, grid/list item, facet navigation
core-libs/storefront/cms-components/product/carousel/	ProductCarouselComponent, ProductReferencesComponent, ProductCarouselService
core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts	route PDP
core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts	route search/brand/category
feature-libs/product/variants/**	selettori varianti, guard, endpoint variants
feature-libs/product/bulk-pricing/**	scope bulkPrices, tabella prezzi a scaglioni
feature-libs/product/future-stock/**	facade lazy future stock
feature-libs/product/image-zoom/**	sostituzione di ProductImagesComponent con zoom
feature-libs/product-multi-dimensional/**	selector multi-dimensionale e range prezzo in lista

File aggiuntivi

PATH	RUOLO
feature-libs/product-multi-dimensional/_index.scss	entry SCSS: importa solo gli stili di selector
feature-libs/product-multi-dimensional/public_api.ts	entry point primario: riesporta il solo product-multi-dimensional-selector.module
feature-libs/product-multi-dimensional/schematics/add-product-multi-dimensional/index.ts	rule ng add (addSpartacusProductMultiDimensional)
feature-libs/product-multi-dimensional/schematics/add-product-multi-dimensional/index_spec.ts	test Jest dello schematic: feature selector e list, lazy/eager, snapshot
feature-libs/product/_index.scss	entry SCSS: mixin visible-focus/button-reset e import di bulk-pricing, variants, image-zoom, future-stock

PATH	RUOLO
feature-libs/product/schematics/add-product/index.ts	rule `ng add` di `@spartacus/product` (`addSpartacusProduct`)
feature-libs/product/schematics/add-product/index_spec.ts	test Jest dello schematic: BulkPricing, Variants, ImageZoom, FutureStock, config B2B
feature-libs/product/vitest.config.ts	configurazione Vitest della lib `product`: alias per helper di test, coverage v8 con soglie

3. Come funziona passo per passo

3.1 Wiring: store, effects, SSR

`ProductStoreModule` registra la feature NgRx `product` con `metaReducers` e gli effects, e fornisce via factory una `StateConfig` che marca l'intero slice `product` come `TRANSFER_STATE`: lo stato prodotto calcolato in SSR viene trasferito al browser (core-libs/core/src/product/store/product-store.module.ts:L20-L43). La factory serve solo a poter usare la costante `PRODUCT_FEATURE` come chiave calcolata (commento a core-libs/core/src/product/store/product-store.module.ts:L21-L21).

Lo slice contiene sei sotto-stati (core-libs/core/src/product/store/product-state.ts:L14-L33):

- `details`: `EntityScopedLoaderState<Product>` — prodotti per codice e per scope;
- `search`: `{ results, suggestions, auxResults }`;
- `searchByCode`: `EntityScopedLoaderState<Product>` (carousel);
- `searchByCategory`: `EntityScopedLoaderState<Product[]>`;
- `reviews` e `references`: stato "singolo prodotto" `{ productCode, list }` (core-libs/core/src/product/store/product-state.ts:L41-L49).

I reducer di `details`, `searchByCode`, `searchByCategory` sono creati con `entityScopedLoaderReducer`; la mappa è fornita via `reducerToken`/`reducerProvider` (core-libs/core/src/product/store/reducers/index.ts:L22-L43). Gli effects sono registrati nell'ordine `ProductsSearchEffects`, `ProductSearchByCodeEffects`, `ProductSearchByCategoryEffects`, `ProductEffects`, `ProductReviewsEffects`, `ProductReferencesEffects` (core-libs/core/src/product/store/effects/index.ts:L14-L21). Il barrel pubblico esporta azioni, stato e selector ma **non** effects né reducers (core-libs/core/src/product/index.ts:L7-L15).

3.2 Gli scope e `ProductService.get`

`ProductService.get(code, scopes = DEFAULT_SCOPE)` restituisce `of(undefined)` se il codice è falsy; altrimenti normalizza gli scope in array e delega a `ProductLoadingService.get` (core-libs/core/src/product/facade/product.service.ts:L39-L46). `DEFAULT_SCOPE` vale `'default'` (core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L7-L7). Il JSDoc garantisce che il prodotto venga caricato una sola volta anche con osservatori paralleli e invita a chiedere solo gli scope necessari (core-libs/core/src/product/facade/product.service.ts:L26-L38).

Attenzione: `isLoading`, `isSuccess`, `hasError` hanno scope di default `''`, non `'default'` (core-libs/core/src/product/facade/product.service.ts:L51-L88).

3.3 `ProductLoadingService`: espansione, cache, combinazione

`ProductLoadingService.get(code, scopes)` (core-libs/core/src/product/services/product-loading.service.ts:L49-L56):

1. valida la chiave con `isKeyInvalid`, che lancia un `Error` per `__proto__`, `constructor`, `prototype` (core-libs/core/src/util/is-key-invalid.ts:L17-L27);
2. espande gli scope con `LoadingScopesService.expand('product', scopes)`;
3. inizializza (una volta) gli observable per scope e per combinazione;
4. restituisce `products[code][scopesIndex]`.

Espansione. `expand` scorre gli scope all'indietro e inserisce gli scope inclusi (`include`) *prima* dello scope che li include, senza duplicati (core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53). Il JSDoc fissa la semantica di merge: l'ultimo scope sovrascrive i precedenti, anche quelli impliciti (core-libs/core/src/occ/services/loading-scopes.service.ts:L22-L24). Il default OCC configura `details.include = [LIST, VARIANTS]` (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58), quindi `expand('product', ['details'])` produce `['list', 'variants', 'details']` (core-libs/core/src/occ/services/loading-scopes.service.ts:L33-L49).

Cache. La mappa è `products: { [code]: { [scope]: Observable<Product> } }`; per più scope la chiave è `scopes.join('ɵ')` (core-libs/core/src/product/services/product-loading.service.ts:L37-L39, core-libs/core/src/product/services/product-loading.service.ts:L88-L90).

Combinazione. Con più di uno scope l'observable combinato usa `uniteLatest` sugli stream per scope, emette `undefined` finché non sono presenti tutte le parti, poi `deepMerge({}, ...parts)` nell'ordine degli scope (core-libs/core/src/product/services/product-loading.service.ts:L74-L85). `uniteLatest` è un `combineLatest` che emette la prima volta in modo sincrono e poi audita con `asapScheduler`, riducendo emissioni simultanee a una sola (core-libs/core/src/util/rxjs/unite-latest.ts:L15-L47).

3.4 Logica di caricamento per scope

Per ogni scope `getProductForScope` costruisce tre stream (core-libs/core/src/product/services/product-loading.service.ts:L102-L143):

- `shouldLoad$`: `true` quando lo stato loader dello scope non è né `loading` né `success` né `error`;
- `productLoadLogic$`: `merge(shouldLoad$, ...reloadTriggers)` → `debounceTime(0)` → `withLatestFrom(isLoading$)` → dispatch di `LoadProduct(code, scope)` solo se non già in loading;
- `productData$`: il valore dallo store.

La logica di load è "attaccata" ai dati con `using()` e condivisa con `shareReplay({ bufferSize: 1, refCount: true })`: il load parte alla prima sottoscrizione e la logica si ferma quando non ci sono più subscriber (core-libs/core/src/product/services/product-loading.service.ts:L139-L142).

Reload trigger. `getProductReloadTriggers` aggiunge:

- un trigger `maxAge`, attivo solo se configurato **e** in browser (`isPlatformBrowser`), che osserva `LOAD_PRODUCT_SUCCESS/FAIL` e `LOAD_PRODUCT` filtrati per codice e scope (core-libs/core/src/product/services/product-loading.service.ts:L157-L185);
- i trigger `reloadOn`: classi `CxEvent` trasformate in stream con `eventService.get(e)` (core-libs/core/src/product/services/product-loading.service.ts:L187-L191).

`getMaxAge` legge `backend.loadingScopes[model][scope].maxAge` in secondi e lo converte in millisecondi; `getReloadTriggers` legge `reloadOn` o `[]` (core-libs/core/src/occ/services/loading-scopes.service.ts:L61-L77).

`getMaxAgeTrigger` calcola l'età dall'ultimo load completato: se scaduta emette subito `true`, altrimenti programma l'emissione a `maxAge - age`; dopo ogni fine-load programma un nuovo `true` a `maxAge`, ritirato (`withdrawOn`) se parte un nuovo load (core-libs/core/src/product/services/product-loading.service.ts:L204-L242). Poiché `timestamp` parte da `0`, alla prima sottoscrizione l'età risulta maggiore di `maxAge` e il trigger emette subito; l'emissione si fonde con quella di `shouldLoad$` grazie al `debounceTime(0)` (core-libs/core/src/product/services/product-loading.service.ts:L210-L227, core-libs/core/src/product/services/product-loading.service.ts:L120-L124).

3.5 Store: azioni e reducer scoped

`LoadProduct(payload: code, scope = '')`, `LoadProductFail(code, payload, scope)` e `LoadProductSuccess(payload: Product, scope)` estendono le azioni scoped-loader con entità `PRODUCT_DETAIL_ENTITY`; il success usa `payload.code ?? ''` come id (core-libs/core/src/product/store/actions/product.action.ts:L29-L64).

`scopedLoaderReducer` indicizza lo stato per `action.meta.scope ?? ''` e delega a `loaderReducer` (core-libs/core/src/state/utils/scoped-loader/scoped-loader.reducer.ts:L20-L44). Nel `loaderReducer` un'azione di load **mantiene** il `value` precedente (stale-while-loading), un errore lo azzera, il success lo sostituisce (core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L60).

Il selector `getSelectedProductStateFactory(code, scope = '')` estrae lo stato loader per entità+scope con fallback a `initialLoaderState` (core-libs/core/src/product/store/selectors/product.selectors.ts:L19-L29).

3.6 Effects: bufferizzazione e `getMany`

`ProductEffects.loadProduct$` mappa le `LoadProduct` in `{ code, scope }`, le accumula con `bufferDebounceTime(debounce)` e chiama `productConnector.getMany(products)` unendo gli stream risultanti; tutto è ritirato da `withdrawOn(contextChange$)` (cambio lingua/valuta) (core-libs/core/src/product/store/effects/product.effect.ts:L34-L57). L'effect è "a factory" (`({ scheduler, debounce = 0 } = {}) => …`) per poter iniettare lo scheduler nei test (core-libs/core/src/product/store/effects/product.effect.ts:L34-L38).

- `bufferDebounceTime` accumula i valori e li emette come array dopo `debounceTime(time)`, poi resetta il buffer (core-libs/core/src/util/rxjs/buffer-debounce-time.ts:L10-L24).
- `withdrawOn(notifier) = notifier.pipe(startWith(undefined), switchMap(() => source))`: a ogni emissione del notifier risottoscrive la sorgente, cancellando le richieste in corso (core-libs/core/src/util/rxjs/withdraw-on.ts:L19-L27).

`productLoadEffect` emette `LoadProductSuccess({ code, ...data }, scope)` o `LoadProductFail` con errore normalizzato; se `data$` manca emette `LoadProductFail` con `Error('Scoped product data does not exist')` (core-libs/core/src/product/store/effects/product.effect.ts:L59-L91).

3.7 Connector, adapter OCC e ottimizzazione delle richieste

`ProductAdapter` dichiara `load(code, scope?)` obbligatorio e `loadMany?(products)` opzionale (core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30). `ProductConnector.getMany` usa `loadMany` se esiste, altrimenti mappa ogni prodotto su `adapter.load(code, scope)` (core-libs/core/src/product/connectors/product/product.connector.ts:L19-L33).

`OccProductAdapter.loadMany` costruisce un URL per scope e delega a `OccRequestsOptimizerService.scopedDataLoad`, applicando poi `PRODUCT_NORMALIZER` a ogni `data$` (core-libs/core/src/occ/adapters/

product/occ-product.adapter.ts:L35-L52). L'URL viene da `buildUrl('product', { urlParams: { productCode }, scope })` (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L29-L59).

`scopedDataLoad` chiede a `OccFieldsService.getOptimalUrlGroups` di raggruppare gli URL che differiscono solo per `fields` (core-libs/core/src/occ/services/occ-fields.service.ts:L50-L82): con un solo scope per URL passa il `data$` diretto; con più scope fa **una sola GET** con `shareReplay(1)` ed estrae per ogni scope i propri campi con `extractFields` (core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L34-L79). Tutti gli endpoint core di prodotto condividono il path `products/${productCode}` (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28): gli scope richiesti nello stesso tick finiscono nello stesso buffer dell'effect (`bufferDebounceTime(debounce, scheduler)`, core-libs/core/src/product/store/effects/product.effect.ts:L45-L47) e producono una GET unica. Scope richiesti in momenti diversi, ad esempio `variants` caricato dalla guardia CMS `ProductVariantsGuard` (feature-libs/product/variants/components/guards/product-variants.guard.ts:L37-L37), o con path diverso, come `bulkPrices` → `orgProducts/...` (feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L14-L15), producono richieste separate.

Se uno scope non ha endpoint configurato, `getEndpointForScope` in dev mode logga un warning e ricade su `default` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L224). Gli endpoint core coprono `default`, `list`, `details`, `promotions`, `attributes`, `price`, `stock`, `unit`, `list_item` — non `variants`, `code` o `multi_dimensional*` (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28).

3.8 Normalizzazione

`ProductOccModule` registra su `PRODUCT_NORMALIZER` `ProductImageNormalizer` e `ProductNameNormalizer` (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L76).

- `ProductImageNormalizer` ristruttura la lista piatta di immagini in `images[imageType][format]`, o in array indicizzato quando c'è `galleryIndex` (core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L28-L61); prefissa gli URL relativi con `backend.media.baseUrl`, fallback `backend.occ.baseUrl`, lasciando invariati quelli che iniziano con `http`, `data:image` o `//` (core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L84-L100).
- `ProductNameNormalizer` rimuove l'HTML dal `name`, conserva l'originale in `nameHtml` e genera uno `slug` (core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts:L17-L64).
- `OccProductSearchPageNormalizer` normalizza i facet e converte ogni prodotto con `PRODUCT_NORMALIZER` (core-libs/core/src/occ/adapters/product/converters/occ-product-search-page-normalizer.ts:L31-L47); rimuove `topValues` impostando `topValueCount` (default 6) (core-libs/core/src/occ/adapters/product/converters/occ-product-search-page-normalizer.ts:L91-L102) e scarta i facet non visibili e quelli in cui nessun valore è selezionato o restringe il risultato (cioè tutti i valori hanno `count >= totalResults`); il filtro si applica solo se `pagination.totalResults` è presente (workaround CS-427) (core-libs/core/src/occ/adapters/product/converters/occ-product-search-page-normalizer.ts:L54-L80).

Se un token non ha converter registrati, `ConverterService` restituisce la sorgente invariata (core-libs/core/src/util/converter.service.ts:L80-L112).

3.9 Invalidazione: contesto e autenticazione

- `LANGUAGE_CHANGE`/`CURRENCY_CHANGE`: il meta-reducer `clearProductsState` azzera l'intero slice (core-libs/core/src/product/store/reducers/index.ts:L45-L59), mentre gli effect con `withdrawOn(contextChange$)` cancellano le richieste in volo (core-libs/core/src/product/store/effects/product.effect.ts:L26-L32). Per gli stream ancora sottoscritti lo stato torna iniziale, `shouldLoad$` riemette e parte un nuovo load (core-libs/core/src/product/services/product-loading.service.ts:L102-L112).

- LOGIN/LOGOUT: `clearProductPrice$` emette `ClearProductPrice` (core-libs/core/src/product/store/effects/product.effect.ts:L93-L99), un `EntityScopedResetAction` con id `undefined` e scope `PRICE` (core-libs/core/src/product/store/actions/product.action.ts:L66-L72); nell'`entityReducer` un id `undefined` applica l'azione a **tutte** le entità (core-libs/core/src/state/utils/entity/entity.reducer.ts:L28-L44). Lo stato by-code viene svuotato con `ClearProductSearchByCodeState` (core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L107-L113).

3.10 Ricerca: PLP e search box

`ProductSearchService.search` fa dispatch di `SearchProducts` **solo se** `query` è truthy; `getResults` seleziona `results`; `clearResults` invia `ClearProductSearchResult({ clearPageResults: true })` (core-libs/core/src/product/facade/product-search.service.ts:L21-L42).

`SearchboxService extends ProductSearchService`: `search` invia `SearchProducts(…, auxiliary = true)` anche con query vuota, `getResults` legge `auxResults` (core-libs/core/src/product/facade/searchbox.service.ts:L19-L68). `searchWithCompletion`/`searchSuggestionsWithCompletion` fanno dispatch e poi restituiscono la prima azione SUCCESS/FAIL da `ActionsSubject` con `take(1)`, pur essendo tipizzati `Observable<boolean>` (core-libs/core/src/product/facade/searchbox.service.ts:L40-L53, core-libs/core/src/product/facade/searchbox.service.ts:L86-L99); il consumer `SearchBoxComponentService` tratta infatti il valore come azione (core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L93-L108).

`ProductsSearchEffects.searchProducts$` raggruppa per `auxiliary` (`groupBy`) e usa `switchMap` per gruppo: una ricerca PLP non cancella quella del search box e viceversa (core-libs/core/src/product/store/effects/product-search.effect.ts:L26-L43). Un errore `ArrayIndexOutOfBoundsError` produce il global message `searchBox.queryError` (core-libs/core/src/product/store/effects/product-search.effect.ts:L44-L63). I suggerimenti usano `switchMap` e convertono `undefined` in `[]` (core-libs/core/src/product/store/effects/product-search.effect.ts:L72-L101). Il reducer scrive `auxResults` o `results` in base ad `auxiliary` (core-libs/core/src/product/store/reducers/product-search.reducer.ts:L25-L54).

Lato OCC, `OccProductSearchAdapter.search` usa `{ pageSize: 20 }` se `searchConfig` è `undefined`, imposta `OCC_HTTP_TOKEN.sendUserIdAsHeader: true` e mette `query` + config nei query param (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L42-L56, core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L147-L156). Se la risposta contiene `keywordRedirectUrl`, è **l'adapter stesso** a navigare con il `Router` (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L59-L69, core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L164-L194). `loadSuggestions` ha `pageSize` di default 3 (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L123-L135).

3.11 PLP guidata dall'URL

`ProductListComponentService.searchByRouting$` combina router state (distinto per `state.url`), lingua e valuta attive, applica `debounceTime(0)` e lancia la ricerca se i criteri sono cambiati (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L69-L90).

- I criteri vengono dai query param `query`, `pageSize`, `currentPage`, `sortCode`; in assenza di `query`, dai route param (`categoryCode`/`brandCode` → `:relevance:allCategories:<code>`); `pageSize` ricade su `view.defaultPageSize` (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L191-L223).
- `searchIfCriteriaHasChanged` confronta query (senza sort code), pagina, pageSize e sortCode con l'ultima risposta in store (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L92-L169); rimuove dalla query il sort `relevance` e tratta `currentPage` 0 come

undefined (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L117-L152).

- model$ lega ricerca e risultati con using() + shareReplay({ bufferSize: 1, refCount: true }): lasciata la PLP, la ricerca su route change si ferma (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L171-L183).
- sort(sortCode) naviga con queryParamsHandling: 'merge' (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L270-L286); getPageItems (infinite scroll) cerca senza navigare (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L247-L265).

ProductListComponent sceglie Grid se il template di pagina è ProductGridPageTemplate e annuncia sorting.pageViewUpdated in modo assistivo ai cambi successivi (core-libs/storefront/cms-components/product/product-list/container/product-list.component.ts:L64-L93). Il template usa cx-pagination (queryParam="currentPage") o cx-product-scroll se l'infinite scroll è attivo (core-libs/storefront/cms-components/product/product-list/container/product-list.component.html:L25-L79). ProductListModule mappa CMSProductListComponent, ProductGridComponent e SearchResultsListComponent su ProductListComponent con composition.inner: ['ProductAddToCartComponent'] (core-libs/storefront/cms-components/product/product-list/product-list.module.ts:L63-L94).

Grid/list item forniscono localmente ProductListItemContextSource (alias ProductListItemContext) e vi pubblicano il prodotto (core-libs/storefront/cms-components/product/product-list/product-grid-item/product-grid-item.component.ts:L32-L80); gli outlet ITEM_DETAILS e ITEM_ACTIONS avvolgono rating/prezzo e add-to-cart (core-libs/storefront/cms-components/product/product-list/product-grid-item/product-grid-item.component.html:L23-L47). L'add-to-cart è nascosto se manca il prezzo o purchasable === false (core-libs/storefront/cms-components/product/product-list/product-list.service.ts:L12-L16).

3.12 Facet

ProductFacetService.searchResult$ filtra il model$ della PLP verificando che il risultato appartenga alla pagina corrente (categoria: query contiene allCategories:<id>; ricerca: freeTextSearch uguale alla query di route) (core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/product-facet.service.ts:L40-L96) e toglie dagli active facet il breadcrumb implicito della categoria/brand di route (core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/product-facet.service.ts:L98-L119).

FacetService tiene lo stato UI in una Map<facet.name, BehaviorSubject<FacetCollapseState>> inizializzata con topValueCount (core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/facet.service.ts:L28-L48, core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/facet.service.ts:L113-L127). Selezionare un facet è una navigazione: routerLink="./" con [queryParams]="getLinkParams(value)" (core-libs/storefront/cms-components/product/product-list/product-facet-navigation/facet/facet.component.html:L13-L14); getLinkParams decodifica la query e sostituisce + con spazio (core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/facet.service.ts:L146-L159).

In parallelo ProductEventBuilder confronta a coppie (pairwise) i risultati PLP consecutivi ed emette FacetChangedEvent individuando l'unico breadcrumb aggiunto/rimosso (core-libs/core/src/product/event/product-event.builder.ts:L37-L67); il builder è istanziato dal costruttore di ProductEventModule (core-libs/core/src/product/event/product-event.module.ts:L10-L15).

3.13 PDP

`CurrentProductService.getProduct(scopes?)` legge `productCode` dal router state, usa di default lo scope `DETAILS`, restituisce `of(null)` senza codice e filtra gli `undefined` (core-libs/storefront/cms-components/product/current-product.service.ts:L27-L57). I componenti chiedono scope diversi: `ProductSummaryComponent` `DETAILS+PRICE+PROMOTIONS` (core-libs/storefront/cms-components/product/product-summary/product-summary.component.ts:L33-L40), `ProductAttributesComponent` `ATTRIBUTES` (mappato a `ProductSpecsTabComponent`) (core-libs/storefront/cms-components/product/product-tabs/product-attributes/product-attributes.component.ts:L19-L23).

Altri pezzi della PDP:

- il summary avvolge prezzo e testo negli outlet `PRICE` e `SUMMARY` con contesto `{ product }` (core-libs/storefront/cms-components/product/product-summary/product-summary.component.html:L6-L23);
- `ProductIntroComponent.showReviews` cerca nel DOM la tab reviews per etichetta tradotta e la clicca (core-libs/storefront/cms-components/product/product-intro/product-intro.component.ts:L65-L151);
- `ProductImagesComponent` usa `images.PRIMARY` come immagine principale e mostra le thumbnail solo se `GALLERY` ha almeno 2 elementi (core-libs/storefront/cms-components/product/product-images/product-images.component.ts:L34-L109);
- `StockNotificationComponent` calcola `outOfStock$` e, per utenti non anonimi, carica preferenze e interessi `BACK_IN_STOCK` (core-libs/storefront/cms-components/product/stock-notification/stock-notification.component.ts:L77-L100).

La route PDP è `path: null` con `CmsPageGuard`, `cxRoute: 'product'` e matcher `PRODUCT_DETAILS_URL_MATCHER` (core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L14-L38); la PLP registra `search`, `brand` e per ultima `category`, perché il suo matcher può essere greedy (core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L13-L52).

3.14 Page meta resolver

- `ProductPageMetaResolver` (`PRODUCT_PAGE`) deriva il prodotto dal parametro `productCode` con scope `DETAILS` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L57-L69); titolo `name | primaCategoria | manufacturer` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L91-L102), immagine `images.PRIMARY.zoom.url` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L145-L149), canonical che risale `baseProduct` con scope `LIST` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L167-L209).
- `CategoryPageMetaResolver` usa i risultati di ricerca solo se la pagina contiene `CMSProductListComponent` o `ProductGridComponent` (core-libs/core/src/product/services/category-page-meta.resolver.ts:L40-L54).
- `SearchPageMetaResolver` si applica a `CONTENT_PAGE` con template `SearchResultsListPageTemplate` (core-libs/core/src/product/services/search-page-meta.resolver.ts:L35-L73); `CouponSearchPageResolver` somma +1 al punteggio base se nella query string c'è `couponcode`, altrimenti −1, così sulla ricerca normale vince `SearchPageMetaResolver` (core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L89-L95).

3.15 Carousel e ricerca per codici

`ProductCarouselComponent` carica ogni codice con `ProductSearchByCodeService.get({ code, scope })`, scope `carousel` se la mappatura ha `composition.inner`, altrimenti `carouselMinimal` (core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L62-L83). Con `categoryCodes` risolve prima i prodotti di ogni categoria (scope `CODE`) e li accoda (core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L93-L120).

`ProductSearchByCodeService.get` seleziona lo stato scoped e, con `auditTime(0)`, fa dispatch del load se lo stato è iniziale; il load è legato alla sottoscrizione con `using()` (core-libs/core/src/product/facade/product-search-by-code.service.ts:L30-L61), senza `shareReplay` (core-libs/core/src/product/facade/product-search-by-code.service.ts:L57-L60).

`ProductSearchByCodeEffects.searchByCodes$` raggruppa per scope, bufferizza e fa **una** chiamata `searchByCodes(codes, scope)` per gruppo, emettendo un success per prodotto, associando prodotti e codici per indice (core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L44-L84). L'adapter spezza i codici in chunk da 100 ed esegue `search('', { filters: 'code:…', pageSize: 100 }, scope)` per chunk, unendo con `forkJoin` (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L73-L107).

3.16 Reviews e references

- `ProductReviewService.getByProductCode` fa dispatch di `LoadProductReviews` quando il selector restituisce `undefined` (prodotto diverso da quello in stato) (core-libs/core/src/product/facade/product-review.service.ts:L22-L34, core-libs/core/src/product/store/selectors/product-reviews.selectors.ts:L21-L30). Il reducer gestisce solo il success di load (core-libs/core/src/product/store/reducers/product-reviews.reducer.ts:L16-L34); il POST è form-urlencoded (core-libs/core/src/occ/adapters/product/occ-product-reviews.adapter.ts:L39-L55) e produce i global message di conferma/errore (core-libs/core/src/product/store/effects/product-reviews.effect.ts:L78-L104).
- `ProductReferencesComponent` a ogni cambio di prodotto corrente chiama `cleanReferences()`, poi ricarica con `productReferenceTypes` del CMS (core-libs/storefront/cms-components/product/carousel/product-references/product-references.component.ts:L56-L105). Il reducer accumula la lista deduplicando per `referenceType + target.code` e sovrascrive `productCode` (core-libs/core/src/product/store/reducers/product-references.reducer.ts:L21-L46); l'effect usa `mergeMap` (core-libs/core/src/product/store/effects/product-references.effect.ts:L22-L46).

3.17 Stock in tempo reale

`ProductAvailabilityService` usa `CommandService` con strategia `CancelPrevious`; il JSDoc lo distingue dallo stock eventualmente cachato di `ProductService` (core-libs/core/src/product/facade/product-availability.service.ts:L17-L59). L'adapter OCC chiama `productAvailabilities?filters=<code>:<unit>`, estrae `availabilityItems[0].unitAvailabilities[0]` e in caso di errore restituisce `{}` (core-libs/core/src/occ/adapters/product/occ-product-availability-adapter.ts:L21-L45).

3.18 Feature lib `@spartacus/product`

L'entry point principale esporta `{}`: tutto è nelle entry secondarie (feature-libs/product/public_api.ts:L14-L20). Ogni feature ha un *root module* eager che dichiara `featureModules.<nome>.cmsComponents`; l'app aggiunge `module: () => import(...)` (projects/storefrontapp/src/app/spartacus/features/product/product-variants-feature.module.ts:L21-L33); `CmsFeaturesService` mappa un tipo CMS su una feature solo se ha sia `module` sia `cmsComponents` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76).

- **variants**: il root dichiara `productVariants` per `ProductVariantSelectorComponent` e registra in modo eager l'outlet `ITEM_DETAILS` (AFTER) con le style icons (feature-libs/product/variants/root/product-variants-root.module.ts:L19-L42); il modulo lazy aggiunge l'endpoint `product.variants` (feature-libs/product/variants/occ/config/default-occ-product-variants-config.ts:L9-L20) e mappa il tipo CMS su `ProductVariantsContainerComponent` con `ProductVariantsGuard` (feature-libs/product/variants/components/product-variants-components.module.ts:L23-L32). Il guard carica lo scope `VARIANTS` e, se il

prodotto non è acquistabile, restituisce un `UrlTree` verso una variante con stock (o la prima) (feature-libs/product/variants/components/guards/product-variants.guard.ts:L32-L80); `GuardsComposer` prende solo il primo valore (core-libs/storefront/cms-structure/services/guards-composer.ts:L65-L65).

- **bulk-pricing**: aggiunge a runtime `ProductScope.BULK_PRICES = 'bulkPrices'` (augmentation + mutazione dell'enum) (feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L9-L15) con endpoint `orgProducts/${productCode}?fields=price(DEFAULT),volumePrices(FULL)` (feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L9-L20); `BulkPricingService` legge quello scope via `ProductService` e calcola lo sconto percentuale (feature-libs/product/bulk-pricing/core/services/bulk-pricing.service.ts:L16-L80).
- **future-stock**: `FutureStockFacade` è una facade lazy (`facadeFactory`, feature `productFutureStockCore`) (feature-libs/product/future-stock/root/facade/future-stock.facade.ts:L12-L29); `FutureStockService` reagisce al router state e chiama il connector solo per utenti non anonimi, senza store né cache (feature-libs/product/future-stock/core/services/future-stock.service.ts:L23-L41).
- **image-zoom**: dichiara la feature `productImageZoom` per `ProductImagesComponent`; caricata la feature, la mappatura sostituisce il componente con `ProductImageZoomProductImagesComponent` (feature-libs/product/image-zoom/root/product-image-zoom-root.module.ts:L11-L26, feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom.module.ts:L51-L60), che estende `ProductImagesComponent` con lo stesso selettore (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-product-images/product-image-zoom-product-images.component.ts:L21-L44).

3.19 `@spartacus/product-multi-dimensional`

Il root del selector fornisce `multiDimensional.imageFormat = STYLE_SWATCH` e dichiara la feature `productMultiDimensionalSelector` (feature-libs/product-multi-dimensional/selector/root/product-multi-dimensional-selector-root.module.ts:L15-L33). L'OCC aggiunge gli scope `multi_dimensional` (variantMatrix annidata) e `multi_dimensional_availability` (feature-libs/product-multi-dimensional/selector/occ/config/default-occ-product-multi-dimensional-selector-config.ts:L9-L26). Il guard, senza `productCode`, consente l'accesso solo con `cmsTicketId` (SmartEdit), altrimenti verifica la disponibilità e reindirizza (feature-libs/product-multi-dimensional/selector/components/guards/product-multi-dimensional-selector.guard.ts:L39-L61). `getVariants` scende un livello della matrice per ogni categoria (feature-libs/product-multi-dimensional/selector/core/services/product-multi-dimensional-selector.service.ts:L51-L88). La parte `list` sostituisce (`REPLACE`) l'outlet `ITEM_DETAILS` con un range di prezzo (feature-libs/product-multi-dimensional/list/root/product-multi-dimensional-list-root.module.ts:L33-L44).

3.20 Stili, schematics e test delle due librerie

`feature-libs/product/_index.scss` definisce due mixin locali, `visible-focus` (outline con `--cx-color-visual-focus`) e `button-reset`, prima di importare gli stili delle quattro sotto-feature (feature-libs/product/_index.scss:L3-L30); `feature-libs/product-multi-dimensional/_index.scss` importa solo `./selector/index` (feature-libs/product-multi-dimensional/_index.scss:L3-L3) e l'entry point primario della lib multi-dimensionale riesporta il solo modulo del selector (feature-libs/product-multi-dimensional/public_api.ts:L7-L7).

I due schematic hanno la stessa sequenza `validateSpartacusInstallation` → `analyzeCrossFeatureDependencies` → `chain([analyzeApplication, addFeatures, addPackageJsonDependenciesForLibrary, finalizeInstallation])` (feature-libs/product/schematics/add-product/index.ts:L25-L43, feature-libs/product-multi-dimensional/schematics/add-product-multi-dimensional/index.ts:L25-L45). Lo spec di product, per BulkPricing, controlla anche via snapshot il modulo di configurazione generato (feature-libs/product/schematics/add-product/index_spec.ts:L133-L140); lo

spec multi-dimensionale crea il `SchematicTestRunner` con il nome di collection `SPARTACUS_PRODUCT` (feature-libs/product-multi-dimensional/schematics/add-product-multi-dimensional/index_spec.ts:L23-L27) e copre le feature selector e list (feature-libs/product-multi-dimensional/schematics/add-product-multi-dimensional/index_spec.ts:L89-L175).

`feature-libs/product/vitest.config.ts` dichiara alias verso helper di test di core/storefront (feature-libs/product/vitest.config.ts:L16-L22), esclude dalla coverage spec, `public_api.ts`, `index.ts`, `*.module.ts`, `schematics/**` e simili (feature-libs/product/vitest.config.ts:L39-L50) e fissa le soglie 85/85/65/85 (feature-libs/product/vitest.config.ts:L51-L56).

---

4. Diagrammi

4.1 PDP: dal componente alla GET OCC (scope `details`)

■ **TAVOLA 24.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/cms-components/product/current-product.service.ts:L36-L51, core-libs/core/src/product/facade/product.service.ts:L39-L46, core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53, core-libs/core/src/product/services/product-loading.service.ts:L102-L133, core-libs/core/src/product/store/effects/product.effect.ts:L39-L54, core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L35-L52, core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L44-L76, core-libs/core/src/product/services/product-loading.service.ts:L74-L85.

4.2 Ciclo di vita di uno scope

■ **TAVOLA 24.2** diagramma a tutta pagina, in fondo al capitolo

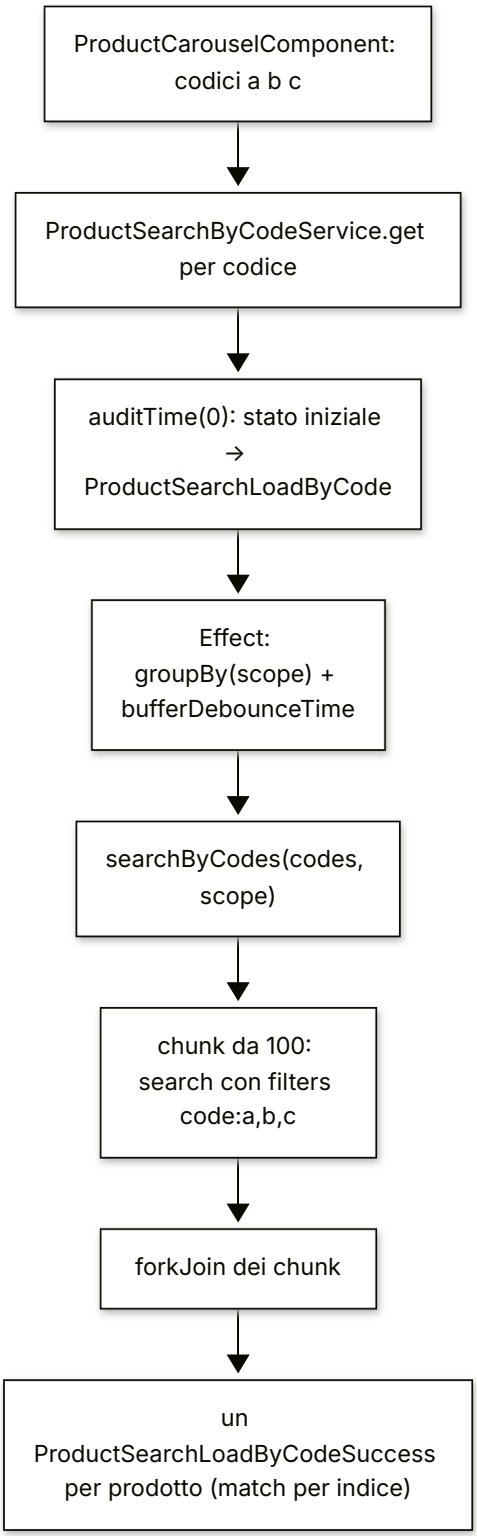
Riferimenti: core-libs/core/src/product/services/product-loading.service.ts:L102-L142, core-libs/core/src/product/services/product-loading.service.ts:L157-L191, core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L60, core-libs/core/src/product/store/reducers/index.ts:L45-L59, core-libs/core/src/product/store/effects/product.effect.ts:L93-L99, core-libs/core/src/state/utils/entity/entity.reducer.ts:L28-L44.

4.3 PLP: route, ricerca, facet

■ **TAVOLA 24.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L69-L90, core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L92-L169, core-libs/core/src/product/store/effects/product-search.effect.ts:L26-L43, core-libs/core/src/product/store/reducers/product-search.reducer.ts:L25-L32, core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/product-facet.service.ts:L40-L96, core-libs/storefront/cms-components/product/product-list/product-facet-navigation/facet/facet.component.html:L13-L14.

4.4 Carousel: N componenti, una chiamata per scope



Riferimenti: `core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L69-L83`, `core-libs/core/src/product/facade/product-search-by-code.service.ts:L46-L53`, `core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L44-L84`, `core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L73-L107`.

5. Estratti di codice reale

5.1 `ProductService.get`

core-libs/core/src/product/facade/product.service.ts:L39-L46

```ts
get(
  productCode: string,
  scopes: (ProductScope | string)[] | ProductScope | string = DEFAULT_SCOPE
): Observable<Product | undefined> {
  return productCode
    ? this.productLoading.get(productCode, ([] as string[]).concat(scopes))
    : of(undefined);
}
```

5.2 Espansione degli scope

core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53

```ts
expand(model: string, scopes: string[]): string[] {
  const scopesConfig = this.config?.backend?.loadingScopes?.[model];

  if (scopesConfig) {
    const expandedScopes = [...scopes];
    let i = expandedScopes.length;

    while (i > 0) {
      i--;
      const includedScopes = scopesConfig[expandedScopes[i]]?.include;
      if (includedScopes) {
        for (const includedScope of includedScopes) {
          if (!expandedScopes.includes(includedScope)) {
            expandedScopes.splice(i, 0, includedScope);
            i++;
          }
        }
      }
    }

    return expandedScopes;
  }

  return scopes;
}
```

5.3 Cache e combinazione degli scope

core-libs/core/src/product/services/product-loading.service.ts:L58-L90

```ts
protected initProductScopes(productCode: string, scopes: string[]): void {
  isKeyInvalid(productCode);

  if (!this.products[productCode]) {
    this.products[productCode] = {};
  }

  for (const scope of scopes) {
    if (!this.products[productCode][scope]) {
      this.products[productCode][scope] = this.getProductForScope(
        productCode,
        scope
      );
    }
  }

  if (scopes.length > 1) {
    this.products[productCode][this.getScopesIndex(scopes)] = uniteLatest(
      scopes.map((scope) => this.products[productCode][scope])
    ).pipe(
      map((productParts) =>
        productParts.every(Boolean)
          ? deepMerge({}, ...productParts)
          : undefined
      ),
      distinctUntilChanged()
    );
  }
}

protected getScopesIndex(scopes: string[]): string {
  return scopes.join('ɵ');
}
```

5.4 Logica di load per scope

core-libs/core/src/product/services/product-loading.service.ts:L98-L143

```ts
protected getProductForScope(
  productCode: string,
  scope: string
): Observable<Product> {
  const shouldLoad$ = this.store.pipe(
    select(
      ProductSelectors.getSelectedProductStateFactory(productCode, scope)
```

```
      ),
      map(
        (productState) =>
          !productState.loading && !productState.success && !productState.error
      ),
      distinctUntilChanged(),
      filter((x) => x)
    );

    const isLoading$ = this.store.pipe(
      select(
        ProductSelectors.getSelectedProductLoadingFactory(productCode, scope)
      )
    );

    const productLoadLogic$ = merge(
      shouldLoad$,
      ...this.getProductReloadTriggers(productCode, scope)
    ).pipe(
      debounceTime(0),
      withLatestFrom(isLoading$),
      tap(([, isLoading]) => {
        if (!isLoading) {
          this.store.dispatch(
            new ProductActions.LoadProduct(productCode, scope)
          );
        }
      })
    );

    const productData$ = this.store.pipe(
      select(ProductSelectors.getSelectedProductFactory(productCode, scope))
    );

    return using(
      () => productLoadLogic$.subscribe(),
      () => productData$
    ).pipe(shareReplay({ bufferSize: 1, refCount: true }));
  }
```

5.5 Trigger maxAge

core-libs/core/src/product/services/product-loading.service.ts:L210-L239

```ts
    let timestamp = 0;

    const now = () => (scheduler ? scheduler.now() : Date.now());
```

```
  const timestamp$ = loadFinish$.pipe(tap(() => (timestamp = now())));

  const shouldReload$: Observable<boolean> = defer(() => {
    const age = now() - timestamp;

    const timestampRefresh$ = timestamp$.pipe(
      delay(maxAge, scheduler),
      map(() => true),
      withdrawOn(loadStart$)
    );

    if (age > maxAge) {
      // we should emit first value immediately
      return merge(of(true), timestampRefresh$);
    } else if (age === 0) {
      // edge case, we should emit max age timeout after next load success
      // could happen with artificial schedulers
      return timestampRefresh$;
    } else {
      // we should emit first value when age will expire
      return merge(
        of(true).pipe(delay(maxAge - age, scheduler)),
        timestampRefresh$
      );
    }
  });
```

5.6 Effect di caricamento con buffer e withdraw

core-libs/core/src/product/store/effects/product.effect.ts:L26-L57

TS

```
// we want to cancel all ongoing requests when currency or language changes,
private contextChange$: Observable<Action> = this.actions$.pipe(
  ofType(
    SiteContextActions.CURRENCY_CHANGE,
    SiteContextActions.LANGUAGE_CHANGE
  )
);

loadProduct$ = createEffect(
  () =>
    ({ scheduler, debounce = 0 } = {}): Observable<
      ProductActions.LoadProductSuccess | ProductActions.LoadProductFail
    > =>
      this.actions$.pipe(
        ofType(ProductActions.LOAD_PRODUCT),
        map((action: ProductActions.LoadProduct) => ({
          code: action.payload,
```

```
            scope: action.meta.scope,
          })),
          // we are grouping all load actions that happens at the same time
          // to optimize loading and pass them all to productConnector.getMany
          bufferDebounceTime(debounce, scheduler),
          mergeMap((products) =>
            merge(
              ...this.productConnector
                .getMany(products)
                .map((productLoad) => this.productLoadEffect(productLoad))
            )
          ),
          withdrawOn(this.contextChange$)
        )
    );
```

5.7 Una GET per più scope

core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L44-L76

```ts
    const mergedUrls = this.occFields.getOptimalUrlGroups(scopedDataWithUrls);

    Object.entries(mergedUrls).forEach(
      ([url, groupedModelsSet]: [
        string,
        {
          [scope: string]: OccFieldsModel;
        },
      ]) => {
        const groupedModels = Object.values(groupedModelsSet);

        if (groupedModels.length === 1) {
          // only one scope for url, we can pass the data straightaway
          result.push({
            ...groupedModels[0].scopedData,
            data$: dataFactory?.(url),
          });
        } else {
          // multiple scopes per url
          // we have to split the model per each scope
          const data$ = dataFactory?.(url).pipe(shareReplay(1));

          groupedModels.forEach((modelData) => {
            result.push({
              ...modelData.scopedData,
              data$: data$?.pipe(
                map((data) => extractFields<T>(data, modelData.fields))
              ),
```

```
            });
          });
        }
      }
    );
```

5.8 Meta-reducer di invalidazione e SSR transfer

core-libs/core/src/product/store/reducers/index.ts:L45-L59

```ts
export function clearProductsState(
  reducer: ActionReducer<any>
): ActionReducer<any> {
  return function (state, action) {
    if (
      action.type === SiteContextActions.CURRENCY_CHANGE ||
      action.type === SiteContextActions.LANGUAGE_CHANGE
    ) {
      state = undefined;
    }
    return reducer(state, action);
  };
}

export const metaReducers: MetaReducer<any>[] = [clearProductsState];
```

core-libs/core/src/product/store/product-store.module.ts:L20-L30

```ts
export function productStoreConfigFactory(): StateConfig {
  // if we want to reuse PRODUCT_FEATURE const in config, we have to use factory instead of plain object
  const config: StateConfig = {
    state: {
      ssrTransfer: {
        keys: { [PRODUCT_FEATURE]: StateTransferType.TRANSFER_STATE },
      },
    },
  };
  return config;
}
```

5.9 Endpoint di default e loading scopes

core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28 (abbreviato)

```ts
        product: {
          default:
            'products/${productCode}?
```

```
fields=DEFAULT,averageRating,images(FULL),classifications,manufacturer,numberOfReviews,categories(FULL),
baseOptions,baseProduct,variantOptions,variantType',
          list: 'products/${productCode}?
fields=code,purchasable,name,summary,price(formattedValue),images(DEFAULT,galleryIndex),baseProduct',
          details:
            'products/${productCode}?
fields=averageRating,stock(DEFAULT),description,availableForPickup,code,url,price(DEFAULT),numberOfRevie
ws,manufacturer,categories(FULL),priceRange,multidimensional,tags,images(FULL)',
          // ...
          price: 'products/${productCode}?fields=price(formattedValue)',
          stock: 'products/${productCode}?fields=stock(DEFAULT)',
          // ...
        },
```

core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58

```ts
    loadingScopes: {
      product: {
        details: {
          include: [ProductScope.LIST, ProductScope.VARIANTS],
        },
      },
    },
```

5.10 Due canali di ricerca indipendenti

core-libs/core/src/product/store/effects/product-search.effect.ts:L29-L43

```ts
    this.actions$.pipe(
      ofType(ProductActions.SEARCH_PRODUCTS),
      groupBy((action: ProductActions.SearchProducts) => action.auxiliary),
      mergeMap((group) =>
        group.pipe(
          switchMap((action: ProductActions.SearchProducts) => {
            return this.productSearchConnector
              .search(action.payload.queryText, action.payload.searchConfig)
              .pipe(
                map((data) => {
                  return new ProductActions.SearchProductsSuccess(
                    data,
                    action.auxiliary
                  );
                }),
```

core-libs/core/src/product/store/reducers/product-search.reducer.ts:L25-L32

```ts
    case ProductActions.SEARCH_PRODUCTS_SUCCESS: {
      const results = action.payload;
      const res = action.auxiliary ? { auxResults: results } : { results };
      return {
        ...state,
        ...res,
      };
    }
```

5.11 Keyword redirect dentro l'adapter

core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L46-L71

```ts
  search(
    query: string,
    searchConfig: SearchConfig = this.DEFAULT_SEARCH_CONFIG,
    scope?: string
  ): Observable<ProductSearchPage> {
    const context = new HttpContext().set(OCC_HTTP_TOKEN, {
      sendUserIdAsHeader: true,
    });

    return this.http
      .get(this.getSearchEndpoint(query, searchConfig, scope), { context })
      .pipe(
        this.converter.pipeable(PRODUCT_SEARCH_PAGE_NORMALIZER),
        tap((productSearchPage) => {
          if (productSearchPage.keywordRedirectUrl) {
            const navigationData = this.parseUrlForNavigation(
              productSearchPage.keywordRedirectUrl
            );
            if (navigationData) {
              const [path, navigationExtras] = navigationData;
              this.router?.navigate(path, navigationExtras);
            }
          }
        })
      );
  }
```

5.12 PLP: ricerca legata alla route e al ciclo di vita del componente

core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L69-L90

```ts
protected searchByRouting$: Observable<ActivatedRouterStateSnapshot> =
  combineLatest([
    this.routing.getRouterState().pipe(
      distinctUntilChanged((x, y) => {
        // router emits new value also when the anticipated `nextState` changes
        // but we want to perform search only when current url changes
        return x.state.url === y.state.url;
      })
    ),
    ...this.siteContext,
  ]).pipe(
    debounceTime(0),
    map(([routerState, ..._context]) => (routerState as RouterState).state),
    tap((state: ActivatedRouterStateSnapshot) => {
      const criteria = this.getCriteriaFromRoute(
        state.params,
        state.queryParams
      );

      this.searchIfCriteriaHasChanged(criteria);
    })
  );
```

core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L180-L183

```ts
readonly model$: Observable<ProductSearchPage> = using(
  () => this.searchByRouting$.subscribe(),
  () => this.searchResults$
).pipe(shareReplay({ bufferSize: 1, refCount: true }));
```

5.13 Carousel: scelta dello scope

core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L69-L83

```ts
items$: Observable<Observable<Product | undefined>[]> =
  this.componentData$.pipe(
    switchMap((data) => this.handleCategoryCodes(data)),
    map((data) => {
      const componentMappingExist = !!data.composition?.inner?.length;
      const codes = data.productCodes?.trim().split(' ') ?? [];
      return { componentMappingExist, codes };
    }),
    map(({ componentMappingExist, codes }) => {
      const scope = componentMappingExist ? 'carousel' : 'carouselMinimal';
      return codes.map((code: string) =>
        this.productSearchByCodeService.get({ code, scope })
```

```
        );
      })
    );
```

5.14 Associazione per indice nella ricerca per codici

core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L63-L85

```ts
              mergeMap((payloads: { code: string; scope: string }[]) => {
                const codes = payloads.map((payload) => payload.code);

                return this.productSearchConnector
                  .searchByCodes(codes, scope)
                  .pipe(
                    switchMap(
                      (
                        searchResults
                      ): ProductActions.ProductSearchLoadByCodeSuccess[] => {
                        // We got all products in a single response from the Connector,
                        // but we need to dispatch separate success actions for each product.
                        return searchResults.products?.map(
                          (product, index) =>
                            new ProductActions.ProductSearchLoadByCodeSuccess({
                              // We assume that the order of the products in the response
                              // is the same as the order of the codes in the request! OCC
implementation guarantees that.
                              ...payloads[index],
                              product,
                            })
                        );
                      }
                    ),
```

5.15 Estendere `ProductScope` a runtime (bulk pricing)

feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L7-L15

```ts
import { ProductScope } from '@spartacus/core';

declare module '@spartacus/core' {
  enum ProductScope {
    BULK_PRICES = 'bulkPrices',
  }
}

(ProductScope as any)['BULK_PRICES'] = 'bulkPrices';
```

5.16 Guard CMS delle varianti

feature-libs/product/variants/components/guards/product-variants.guard.ts:L32-L62

```ts
canActivate(activatedRoute: ActivatedRouteSnapshot): Observable<GuardResult> {
  const productCode = activatedRoute.params?.productCode;
  if (!productCode) {
    return of(true);
  }
  return this.productService.get(productCode, ProductScope.VARIANTS).pipe(
    filter(isNotUndefined),
    switchMap((product: Product) => {
      if (!product.purchasable) {
        const purchasableCode = this.findPurchasableProductCode(product);
        if (purchasableCode) {
          return this.productService
            .get(purchasableCode, ProductScope.LIST)
            .pipe(
              filter(isNotUndefined),
              take(1),
              map((_product: Product) => {
                return this.router.createUrlTree(
                  this.semanticPathService.transform({
                    cxRoute: 'product',
                    params: _product,
                  })
                );
              })
            );
        }
      }
      return of(true);
    })
  );
}
```

5.17 Root module eager di una feature (variants)

feature-libs/product/variants/root/product-variants-root.module.ts:L18-L42

```ts
// TODO: Inline this factory when we start releasing Ivy compiled libraries
export function defaultProductVariantsComponentsConfig(): CmsConfig {
  const config: CmsConfig = {
    featureModules: {
      [PRODUCT_VARIANTS_FEATURE]: {
        cmsComponents: ['ProductVariantSelectorComponent'],
      },
    },
  };
```

```
  return config;
}

@NgModule({
  imports: [ProductVariantStyleIconsModule],
  providers: [
    provideDefaultConfigFactory(defaultProductVariantsComponentsConfig),
    provideOutlet({
      id: ProductListOutlets.ITEM_DETAILS,
      position: OutletPosition.AFTER,
      component: ProductVariantStyleIconsComponent,
    }),
  ],
})
export class ProductVariantsRootModule {}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.loadingScopes.product.<scope>.include	string[]	details: [list, variants]	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58	core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53
backend.loadingScopes.product.<scope>.maxAge	number (secondi)	non impostato (0)	core-libs/core/src/occ/config/loading-scopes-config.ts:L15-L18	core-libs/core/src/occ/services/loading-scopes.service.ts:L61-L65, core-libs/core/src/product/services/product-loading.service.ts:L158-L159
backend.loadingScopes.product.<scope>.reloadOn	Type<CxEvent>[]	[]	core-libs/core/src/occ/config/loading-scopes-config.ts:L19-L22	core-libs/core/src/occ/services/loading-scopes.service.ts:L75-L77, core-libs/core/src/product/services/product-loading.service.ts:L187-L189
backend.occ.endpoints.product.{default,list,details,promotions,attributes,price,stock,unit,list_item}	string	vedi 5.9	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28	core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L54-L59
backend.occ.endpoints.product.variants	string	name,purchasable,baseOptions(DEFAULT),...	feature-libs/product/variants/occ/config/default-occ-product-variants-config.ts:L9-L20	idem (scope variants)
backend.occ.endpoints.product.bulkPrices	string	orgProducts/${productCode}?fields=...	feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L9-L20	via BulkPricingService → ProductService

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.endpoints.product.{multi_dimensional,multi_dimensional_availability}	string	variantMatrix / stock varianti	feature-libs/product-multi-dimensional/selector/occ/config/default-occ-product-multi-dimensional-selector-config.ts:L12-L26	guard e selector MD
backend.occ.endpoints.productSearch.{default,carousel,carouselMinimal}	string	vedi file	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L37-L44	core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L147-L156
backend.occ.endpoints.productSearchByCategory.{default,code}	string	vedi file	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L45-L48	core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L137-L145
backend.occ.endpoints.productSuggestions	string	products/suggestions	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L49-L49	core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L158-L162
backend.occ.endpoints.productReviews	string	products/${productCode}/reviews	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L31-L31	core-libs/core/src/occ/adapters/product/occ-product-reviews.adapter.ts:L57-L62
backend.occ.endpoints.productReferences	string	senza referenceType nel path	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L35-L36	core-libs/core/src/occ/adapters/product/occ-product-references.adapter.ts:L34-L43
backend.occ.endpoints.productAvailabilities	string	productAvailabilities?filters=...	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L29-L30	core-libs/core/src/occ/adapters/product/occ-product-availability-adapter.ts:L25-L33
backend.occ.endpoints.{futureStock,futureStocks}	string	users/${userId}/futureStocks...	feature-libs/product/future-stock/occ/config/default-occ-future-stock.config.ts:L10-L23	feature-libs/product/future-stock/occ/adapters/occ-future-stock.adapter.ts:L67-L87
backend.media.baseUrl / backend.occ.baseUrl	string	— (vedi 21)	config OCC	core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L91-L100
view.defaultPageSize	number	12	core-libs/storefront/cms-components/product/config/default-view-config.ts:L9-L18	core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L197-L197
view.infiniteScroll.{active,productLimit,showMoreButton}	boolean/number/boolean	false/0/false	core-libs/storefront/cms-components/product/config/default-view-config.ts:L9-L18	core-libs/storefront/cms-components/product/product-list/container/product-list.component.ts:L65-L65, core-libs/storefront/cms-components/product/product-list/container/product-scroll/product-scroll.component.ts:L98-L105
multiDimensional.imageFormat	Variant Qualifier	STYLE_SWATCH	feature-libs/product-multi-dimensional/selector/root/config/default-product-multi-dimensional-config.ts:L10-L15	feature-libs/product-multi-dimensional/selector/core/services/product-multi-dimensional-selector-images.service.ts:L22-L23

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`state.ssrTransfer.keys.product`	`StateTransferType`	`TRANSFER_STATE`	`core-libs/core/src/product/store/product-store.module.ts:L20-L30`	meccanismo state transfer (07)
`featureModules.{productVariants,productBulkPricing,productFutureStock,productImageZoom,productMultiDimensionalSelector,productMultiDimensionalList}.cmsComponents`	`string[]`	nei root module	es. `feature-libs/product/variants/root/product-variants-root.module.ts:L19-L29`	`core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76`
Default adapter `pageSize` ricerca	`number`	`20` (solo se `searchConfig` è `undefined`)	`core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L42-L48`	idem
Default adapter `pageSize` suggerimenti	`number`	`3`	`core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L123-L126`	idem
Default `topValueCount` facet	`number`	`6`	`core-libs/core/src/occ/adapters/product/converters/occ-product-search-page-normalizer.ts:L25-L29`	`core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/facet.service.ts:L113-L117`

Feature toggle coinvolti: `a11yShowMoreReviewsFocusVisible` (`core-libs/storefront/cms-components/product/product-tabs/product-reviews/product-reviews.component.ts:L135-L135`), `a11yKeyboardAccessibleZoom` (`feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.ts:L234-L234`), `a11yFutureStockAccordionAriaControls` (`feature-libs/product/future-stock/components/future-stock-accordion/future-stock-accordion.component.html:L16-L16`). Vedi 42.

Esempio di configurazione (simboli esportati da `@spartacus/core`: `provideConfig`, `ProductScope`, `OccConfig`); la forma delle chiavi è quella di `LoadingScopeConfig` (`core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L23`):

```ts
// esempio illustrativo, non presente nel repo
provideConfig(<OccConfig>{
  backend: {
    loadingScopes: {
      product: {
        // ricarica il prezzo ogni 60 s (solo nel browser)
        [ProductScope.PRICE]: { maxAge: 60 },
      },
    },
  },
});
```

7. Commenti e note notevoli dal sorgente

- `// we want to cancel all ongoing requests when currency or language changes,` — core-libs/core/src/product/store/effects/product.effect.ts:L26-L26: spiega perché tutti gli effect di caricamento sono avvolti in `withdrawOn`.
- `* Could be refactored to separate service in future to use in other * max age reload implementations` — core-libs/core/src/product/services/product-loading.service.ts:L197-L198: il meccanismo maxAge esiste solo per i prodotti.
- `// edge case, we should emit max age timeout after next load success // could happen with artificial schedulers` — core-libs/core/src/product/services/product-loading.service.ts:L229-L230.
- `// We assume that the order of the products in the response // is the same as the order of the codes in the request! OCC implementation guarantees that.` — core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L78-L79.
- `const CHUNK_SIZE = 100; // Max limit of ProductSearch OCC` — core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L81-L81.
- `// Uncomment this when occ gets configured` (productReferences con `referenceType` nel path) — core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L32-L34.
- JSDoc di `normalizeUselessFacets` con riferimento al ticket CS-427 — core-libs/core/src/occ/adapters/product/converters/occ-product-search-page-normalizer.ts:L54-L62.
- JSDoc di `resolveCanonicalUrl` sui limiti delle varianti multilivello ("We're not always able to get the super baseProduct") — core-libs/core/src/product/services/product-page-meta.resolver.ts:L167-L179.
- `// TODO: Move from product to more generic place` (`SearchConfig`) — core-libs/core/src/product/model/search-config.ts:L7-L7; il JSDoc del campo `query` parla di "organizational customers", incoerente con il dominio prodotto — core-libs/core/src/product/model/search-config.ts:L20-L23.
- `* This search route configuration is not yet configurable (see https://github.com/SAP/spartacus/issues/7191).` — core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L185-L190.
- `// TODO: drop support for brands as they should be treated similarly as any category.` — core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L218-L219.
- `// TODO: consider dropping this complex passing of cleaned object` — core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L237-L237.
- `// TODO: configurable` (`initialMaxListItems = 5`) — core-libs/storefront/cms-components/product/product-tabs/product-reviews/product-reviews.component.ts:L86-L87.
- `@deprecated areReviewsAvailable$` (ora si usa `product.averageRating`) — core-libs/storefront/cms-components/product/product-intro/product-intro.component.ts:L46-L48.
- `/** @deprecated Use \scrollConfig` instead */ // CXSPA-13739–core-libs/storefront/cms-components/product/product-list/container/product-scroll/product-scroll.component.ts:L40-L41`.
- `<!-- @TODO: Temp. Comment out share link while not in use by CMS -->` — core-libs/storefront/cms-components/product/product-summary/product-summary.component.html:L25-L25.
- `// TODO: (CXSPA-7492) - Remove feature flag next major release.` — feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.ts:L233-L233.
- `// Refuse entry unless it is within a SmartEdit environment` — feature-libs/product-multi-dimensional/selector/components/guards/product-multi-dimensional-selector.guard.ts:L43-L43.

8. Extension point

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
`ProductAdapter` (`load`, `loadMany?`), `ProductSearchAdapter`, `ProductReviewsAdapter`, `ProductReferencesAdapter`, `ProductAvailabilityAdapter`	fornire una propria implementazione della classe astratta	`core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30`, `core-libs/core/src/product/connectors/search/product-search.adapter.ts:L16-L37`	52 — Ricette OCC
Token `PRODUCT_NORMALIZER`, `PRODUCT_SEARCH_PAGE_NORMALIZER`, `PRODUCT_SUGGESTION_NORMALIZER`, `PRODUCT_REVIEW_NORMALIZER`, `PRODUCT_REVIEW_SERIALIZER`, `PRODUCT_REFERENCES_NORMALIZER`	aggiungere un converter `multi: true`	`core-libs/core/src/product/connectors/product/converters.ts:L11-L13`, `core-libs/core/src/product/connectors/search/converters.ts:L14-L20`	52
Endpoint e `fields` per scope	`provideConfig` su `backend.occ.endpoints.product.<scope>`	`core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28`	52
Nuovo scope prodotto	augmentation di `ProductScope` + endpoint, come bulk pricing	`feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L9-L15`	52
`backend.loadingScopes` (`include`, `maxAge`, `reloadOn`)	config	`core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L23`, `core-libs/core/src/occ/adapters/product/product-occ-config.ts:L13-L30`	50 — Ricette config/DI
`ProductLoadingService`, `ProductCatalogService`, `CurrentProductService`, `ProductListComponentService`, `FacetService`, `ProductFacetService`	sostituzione via DI (metodi `protected`)	`core-libs/core/src/product/services/product-loading.service.ts:L58-L192`, `core-libs/core/src/product/services/product-catalog.service.ts:L13-L23`	50
`PageMetaResolver` prodotto/categoria/ricerca	estendere e registrare come multi-provider	`core-libs/core/src/product/product.module.ts:L16-L37`	52
`FacetChangedEvent`	ascolto con `EventService.get`	`core-libs/core/src/product/event/product.events.ts:L12-L37`	50
Outlet `PDP.INTRO/PRICE/SHARE/SUMMARY`, `cx-product-list-item.actions/details`	`provideOutlet` / `cxOutlet` (before/after/replace)	`core-libs/storefront/cms-components/product/product-outlets.model.ts:L7-L16`	51 — Ricette CMS/outlet

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
`ProductListItemContext`	iniettare nei componenti di outlet per leggere il prodotto della card	`core-libs/storefront/cms-components/product/product-list/model/product-list-item-context-source.model.ts:L17-L20`	51
Mappature CMS (`CMSProductListComponent`, `ProductGridComponent`, `SearchResultsListComponent`, `ProductRefinementComponent`, `ProductCarouselComponent`, `ProductReferencesComponent`, `ProductImagesComponent`, `ProductSpecsTabComponent`, `ProductReviewsTabComponent`, …)	`cmsComponents` in `provideConfig`	`core-libs/storefront/cms-components/product/product-list/product-list.module.ts:L63-L94`	51
`ViewConfig` (`view.defaultPageSize`, `view.infiniteScroll`)	config	`core-libs/storefront/shared/config/view-config.ts:L10-L30`	50
Feature lazy (root module + `featureModules`)	pattern di variants/bulk-pricing/future-stock	`feature-libs/product/variants/root/product-variants-root.module.ts:L19-L42`, `feature-libs/product/future-stock/root/facade/future-stock.facade.ts:L12-L29`	53 — Ricette feature lib
Guard CMS (`ProductVariantsGuard`, `ProductMultiDimensionalSelectorGuard`)	`guards` nella mappatura CMS	`feature-libs/product/variants/components/product-variants-components.module.ts:L23-L32`	51

Per i limiti di upgrade (classi non esportate come `CouponSearchPageResolver`) vedi 54 — Upgrade safe: il barrel dei servizi prodotto non esporta `CouponSearchPageResolver` (`core-libs/core/src/product/services/index.ts:L7-L11`). `ProductAvailabilityAdapter` invece è pubblica nonostante il file si chiami `prduct-availability.adapter.ts` (`core-libs/core/src/product/connectors/product/index.ts:L11-L11`, `core-libs/core/src/product/connectors/index.ts:L7-L7`, `core-libs/core/src/product/index.ts:L7-L7`, `core-libs/core/public_api.ts:L29-L29`).

9. Trappole, casi limite, comportamenti controintuitivi

1. **Scope di default diversi.** `get()` usa `'default'`, `isLoading/isSuccess/hasError` usano `''`: chiamati senza scope non osservano lo stesso stato (`core-libs/core/src/product/facade/product.service.ts:L41-L41`, `core-libs/core/src/product/facade/product.service.ts:L53-L53`).
2. **Un errore non si ricarica da solo.** `shouldLoad$` richiede `!error`; servono `maxAge`, `reloadOn` o un reset (`core-libs/core/src/product/services/product-loading.service.ts:L106-L111`).
3. `maxAge` **non funziona in SSR** (`isPlatformBrowser`) (`core-libs/core/src/product/services/product-loading.service.ts:L159-L159`).
4. **La mappa di observable cresce.** `products[code][scope]` non viene mai svuotata (`core-libs/core/src/product/services/product-loading.service.ts:L37-L39`, `core-libs/core/src/product/services/product-loading.service.ts:L61-L72`).

5. **Tutto o niente con più scope.** Il prodotto combinato è `undefined` finché tutte le parti non sono arrivate; un errore su uno scope lo lascia `undefined` perché il value fallito è azzerato (core-libs/core/src/product/services/product-loading.service.ts:L78-L82, core-libs/core/src/state/utils/loader/loader.reducer.ts:L45-L52).
6. `variants` **senza la feature variants.** Il core include `variants` in `details`, ma l'endpoint `product.variants` arriva solo dal modulo lazy: senza di esso lo scope ricade su `default` con warning in dev mode (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58, feature-libs/product/variants/occ/config/default-occ-product-variants-config.ts:L13-L16, core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L224). DA VERIFICARE: se la feature viene caricata dopo che lo scope è già in `success`, nel codice non c'è alcun reload alla registrazione dell'endpoint; l'effetto pratico va osservato a runtime.
7. **Associazione per indice nel carousel.** Se OCC restituisce meno prodotti (codice inesistente), l'associazione slitta e i codici in coda restano in `loading` (core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L73-L83). DA VERIFICARE: dipende dal comportamento reale del backend, non osservabile dal codice frontend.
8. `ProductSearchByCodeService.get` **non condivide.** Manca `shareReplay` (core-libs/core/src/product/facade/product-search-by-code.service.ts:L57-L60): ogni subscriber attiva il proprio `loading$`. DA VERIFICARE se due subscriber simultanei sullo stesso codice producano due `ProductSearchLoadByCode` (entrambi vedrebbero lo stato iniziale dopo `auditTime(0)`) e quindi un codice duplicato nella richiesta bufferizzata.
9. `TypeError` **possibile.** `normalizedError?.details?.[0].type` non protegge il caso di array vuoto (core-libs/core/src/product/store/effects/product-search.effect.ts:L49-L52).
10. **Tipo bugiardo.** `searchWithCompletion` è `Observable<boolean>` ma emette un'azione NgRx (core-libs/core/src/product/facade/searchbox.service.ts:L40-L53).
11. **Query vuote.** `ProductSearchService.search` le ignora, `SearchboxService.search` le invia (core-libs/core/src/product/facade/product-search.service.ts:L22-L29, core-libs/core/src/product/facade/searchbox.service.ts:L25-L35).
12. **Navigazione dallo strato dati.** Il keyword redirect avviene nell'adapter, anche per le ricerche ausiliarie del search box (endpoint `default`, che richiede `keywordRedirectUrl`) (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L59-L69, core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L38-L39). `searchByCodes` riusa `search` (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L95-L95), ma con gli endpoint di default `carousel`/`carouselMinimal` il campo non viene richiesto (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L40-L43), quindi il redirect scatta solo se li si personalizza includendolo.
13. **References accumulate.** Il reducer concatena references di prodotti diversi; senza `cleanReferences()` restano voci vecchie (core-libs/core/src/product/store/reducers/product-references.reducer.ts:L25-L46, core-libs/storefront/cms-components/product/carousel/product-references/product-references.component.ts:L60-L60). Inoltre `productReferenceTypes` è usato come filtro di uguaglianza esatta: DA VERIFICARE il formato reale del campo CMS (singolo valore o lista) (core-libs/core/src/product/store/selectors/product-references.selectors.ts:L33-L37).
14. **Review postata invisibile.** `POST_PRODUCT_REVIEW_SUCCESS` non aggiorna lo stato reviews (core-libs/core/src/product/store/reducers/product-reviews.reducer.ts:L20-L33).
15. **Converter assenti = pass-through.** Nessun converter core per review, suggerimenti e future stock (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L32-L74, core-libs/core/src/util/converter.service.ts:L80-L112).
16. `"undefined"` **come codice prodotto.** `data.productCodes + ' ' + codes` con `productCodes` assente produce la stringa `undefined` (core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L111-L114).

17. **Il carousel di default usa** `carouselMinimal`**.** La mappatura `ProductCarouselComponent` non fornisce `composition` (core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.module.ts:L42-L50); i dati statici di mappatura si fondono con quelli backend (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L35-L51). Le costanti `PRODUCT_SCOPE`/`PRODUCT_SCOPE_ITEM` del componente non sono usate (core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L47-L49).
18. **Stato facet persistente.** `FacetService` (root) indicizza per `facet.name` e non azzera mai: lo stato espanso si conserva tra categorie con facet omonimi (core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/facet.service.ts:L31-L31).
19. `showReviews` **dipende dal DOM** e dal testo tradotto del bottone (core-libs/storefront/cms-components/product/product-intro/product-intro.component.ts:L115-L151).
20. `[innerHTML]` per `nameHtml` e `summary`: ci si affida alla sanitizzazione Angular (core-libs/storefront/cms-components/product/product-list/product-grid-item/product-grid-item.component.html:L17-L21, core-libs/storefront/cms-components/product/product-summary/product-summary.component.html:L22-L22).
21. `galleryIndex` **0 = assente** nello zoom (`if (this.galleryIndex)`) (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-trigger/product-image-zoom-trigger.component.ts:L73-L77).
22. **Thumbnail varianti senza** `media.baseUrl`**.** Usano solo `occ.baseUrl` (feature-libs/product/variants/components/variant-style-selector/product-variant-style-selector.component.ts:L46-L54), a differenza di `ProductImageNormalizer`.
23. **Feature** `productFutureStockCore` **mai configurata.** Funziona perché il componente vive nel modulo lazy che fornisce già la facade (feature-libs/product/future-stock/core/facade/facade-providers.ts:L11-L17). DA VERIFICARE a runtime un uso della facade fuori dal modulo lazy.
24. `productMultiDimensionalList` **senza** `module`**.** `CmsFeaturesService` la ignora; la "feature" agisce solo con l'outlet eager (feature-libs/product-multi-dimensional/list/root/product-multi-dimensional-list-root.module.ts:L19-L31, core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L69). Con variants attiva, sullo stesso `ITEM_DETAILS` convivono un AFTER e un REPLACE: DA VERIFICARE il risultato combinato (dipende da `OutletService`, capitolo 17).
25. **Guard MD vs variants.** Per prodotti non acquistabili senza varianti il guard MD restituisce `false`, quello variants `true` (feature-libs/product-multi-dimensional/selector/components/guards/product-multi-dimensional-selector.guard.ts:L56-L58, feature-libs/product/variants/components/guards/product-variants.guard.ts:L59-L59).
26. **Augmentation con side effect.** `ProductScope.BULK_PRICES` e `LAUNCH_CALLER.PRODUCT_IMAGE_ZOOM` esistono a runtime solo dopo l'import dei rispettivi file (feature-libs/product/bulk-pricing/core/model/index.ts:L8-L9, feature-libs/product/image-zoom/root/models/index.ts:L7-L8).
27. **Stock real-time: errore =** `{}` (core-libs/core/src/occ/adapters/product/occ-product-availability-adapter.ts:L41-L43). Il file dell'adapter astratto si chiama `prduct-availability.adapter.ts` (refuso) (core-libs/core/src/product/connectors/product/index.ts:L11-L11).

10. Come fare debug

- **Redux DevTools**: lo slice `product.details.entities[<code>]` mostra uno stato loader per ogni scope (`loading/success/error/value`), coerente con `scopedLoaderReducer` (core-libs/core/src/state/utils/scoped-loader/scoped-loader.reducer.ts:L20-L44). Cercate le azioni con `[Product] Detail Entity` come entità (core-libs/core/src/product/store/product-state.ts:L15-L15).

- **Quante GET?** Nel tab Network una PDP produce una richiesta `products/<code>?fields=…` con i `fields` fusi per ogni gruppo di scope richiesti nello stesso tick (core-libs/core/src/product/store/effects/product.effect.ts:L45-L47); se ne vedete più, confrontate gli endpoint degli scope coinvolti: vengono fusi solo URL che differiscono per `fields` (core-libs/core/src/occ/services/occ-fields.service.ts:L50-L82).
- **Warning in console** `product endpoint configuration missing for scope "..."`: uno scope senza endpoint sta ricadendo su `default` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L224).
- **Prodotto** `undefined` **per sempre**: verificate se uno degli scope espansi è in `error` (vedi trappole 2 e 5).
- **Ricerca PLP che non riparte**: `searchIfCriteriaHasChanged` salta la ricerca se query/pagina/pageSize/sort coincidono con l'ultima risposta; confrontate `search.results.currentQuery` e `pagination` nello store (core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L96-L115).
- **Search box e PLP che si sovrascrivono**: controllate il flag `auxiliary` sulle azioni `SearchProducts` (core-libs/core/src/product/store/actions/product-search.action.ts:L27-L53).
- **SSR**: lo slice `product` viene trasferito (core-libs/core/src/product/store/product-store.module.ts:L20-L30); se in hydration vedete richieste duplicate, verificate che lo stato trasferito contenga gli scope richiesti dal client.
- **Test**: l'effect `loadProduct$` accetta `{ scheduler, debounce }` per controllare il buffer nei test marble (core-libs/core/src/product/store/effects/product.effect.ts:L34-L38); la lib `product-multi-dimensional` usa Vitest (feature-libs/product-multi-dimensional/vitest.config.ts:L11-L59).

---

11. Domande di autoverifica

1. **Quali scope vengono effettivamente richiesti quando un componente chiama** `ProductService.get(code, ProductScope.DETAILS)` **con la config di default, e in che ordine vengono fusi?** `['list', 'variants', 'details']`; `deepMerge` nell'ordine, quindi `details` sovrascrive i precedenti. — core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L53, core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58, core-libs/core/src/product/services/product-loading.service.ts:L74-L85.
2. **Perché tre scope non producono tre richieste HTTP?** L'effect bufferizza le `LoadProduct` simultanee e chiama `getMany`; l'optimizer raggruppa gli URL che differiscono solo per `fields` in un'unica GET condivisa. — core-libs/core/src/product/store/effects/product.effect.ts:L45-L54, core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L44-L76.
3. **Cosa succede ai prodotti nello store quando l'utente cambia valuta?** Il meta-reducer azzera lo slice, gli effect cancellano le richieste in volo e gli stream ancora osservati ricaricano. — core-libs/core/src/product/store/reducers/index.ts:L45-L59, core-libs/core/src/product/store/effects/product.effect.ts:L26-L32, core-libs/core/src/product/services/product-loading.service.ts:L102-L112.
4. **E al login?** Solo lo scope `price` viene resettato per tutte le entità (id `undefined`), e lo stato by-code viene svuotato. — core-libs/core/src/product/store/actions/product.action.ts:L66-L72, core-libs/core/src/state/utils/entity/entity.reducer.ts:L28-L44, core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L107-L113.
5. **Perché configurare** `maxAge` **non aggiorna i prezzi durante il rendering SSR?** Il trigger è aggiunto solo se `isPlatformBrowser`. — core-libs/core/src/product/services/product-loading.service.ts:L157-L159.
6. **Come fanno PLP e search box a non cancellarsi a vicenda?** L'effect raggruppa per `auxiliary` e applica `switchMap` per gruppo; il reducer scrive `results` o `auxResults`. — core-libs/core/src/product/store/

effects/product-search.effect.ts:L29-L43, core-libs/core/src/product/store/reducers/product-search.reducer.ts:L25-L32.

7. **Selezionare un facet chiama direttamente** `ProductSearchService.search`**?** No: è una navigazione (`routerLink="./" + queryParams`); la ricerca riparte da `searchByRouting$`. — core-libs/storefront/cms-components/product/product-list/product-facet-navigation/facet/facet.component.html:L13-L14, core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L69-L90.
8. **Perché un carousel CMS standard usa lo scope** `carouselMinimal`**?** Lo scope `carousel` è scelto solo se la mappatura ha `composition.inner`, che `ProductCarouselModule` non fornisce. — core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L73-L78, core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.module.ts:L42-L50.
9. **Come aggiunge bulk pricing un nuovo scope senza modificare il core?** Con module augmentation dell'enum `ProductScope` più la mutazione runtime dell'oggetto enum, e un endpoint `product.bulkPrices`. — feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L9-L15, feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L9-L20.
10. **Quando viene caricato il modulo lazy** `@spartacus/product/variants`**?** Quando una pagina contiene `ProductVariantSelectorComponent`: il root module lo dichiara in `featureModules.productVariants.cmsComponents`, l'app fornisce `module`, e `CmsFeaturesService` mappa il tipo sulla feature. — feature-libs/product/variants/root/product-variants-root.module.ts:L19-L29, projects/storefrontapp/src/app/spartacus/features/product/product-variants-feature.module.ts:L21-L33, core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76.

TAVOLA 24.1 **4.1 PDP: dal componente alla GET OCC (scope details)**

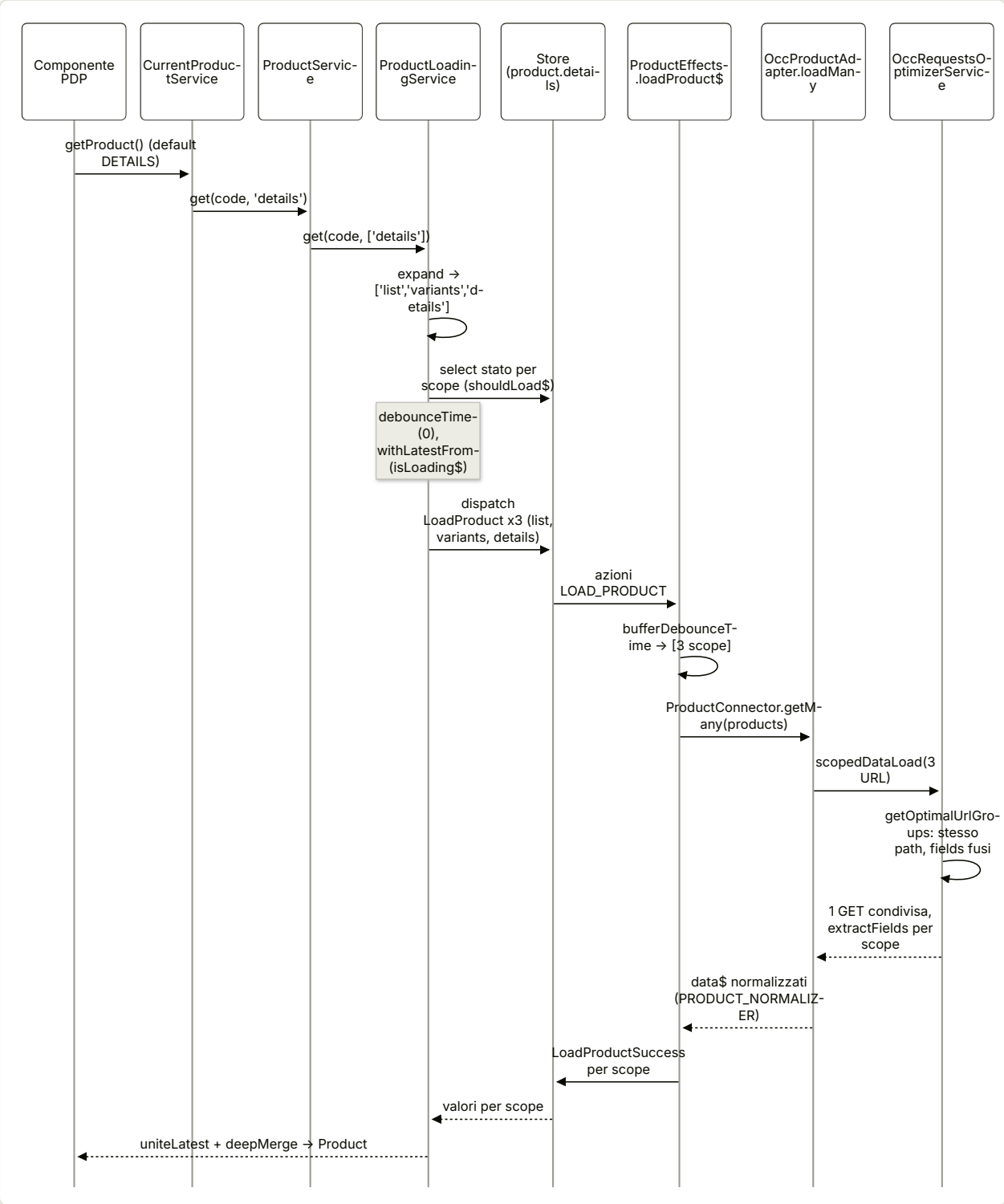


TAVOLA 24.2 **4.2 Ciclo di vita di uno scope**

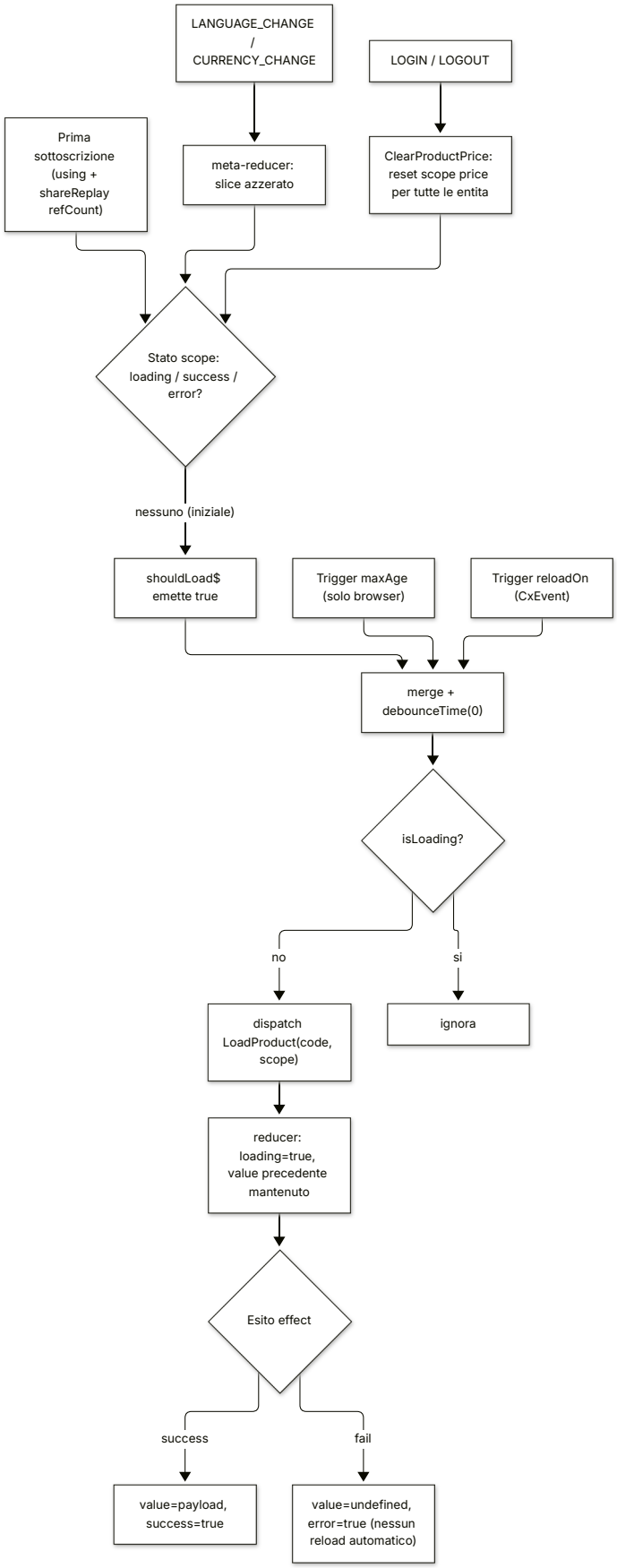
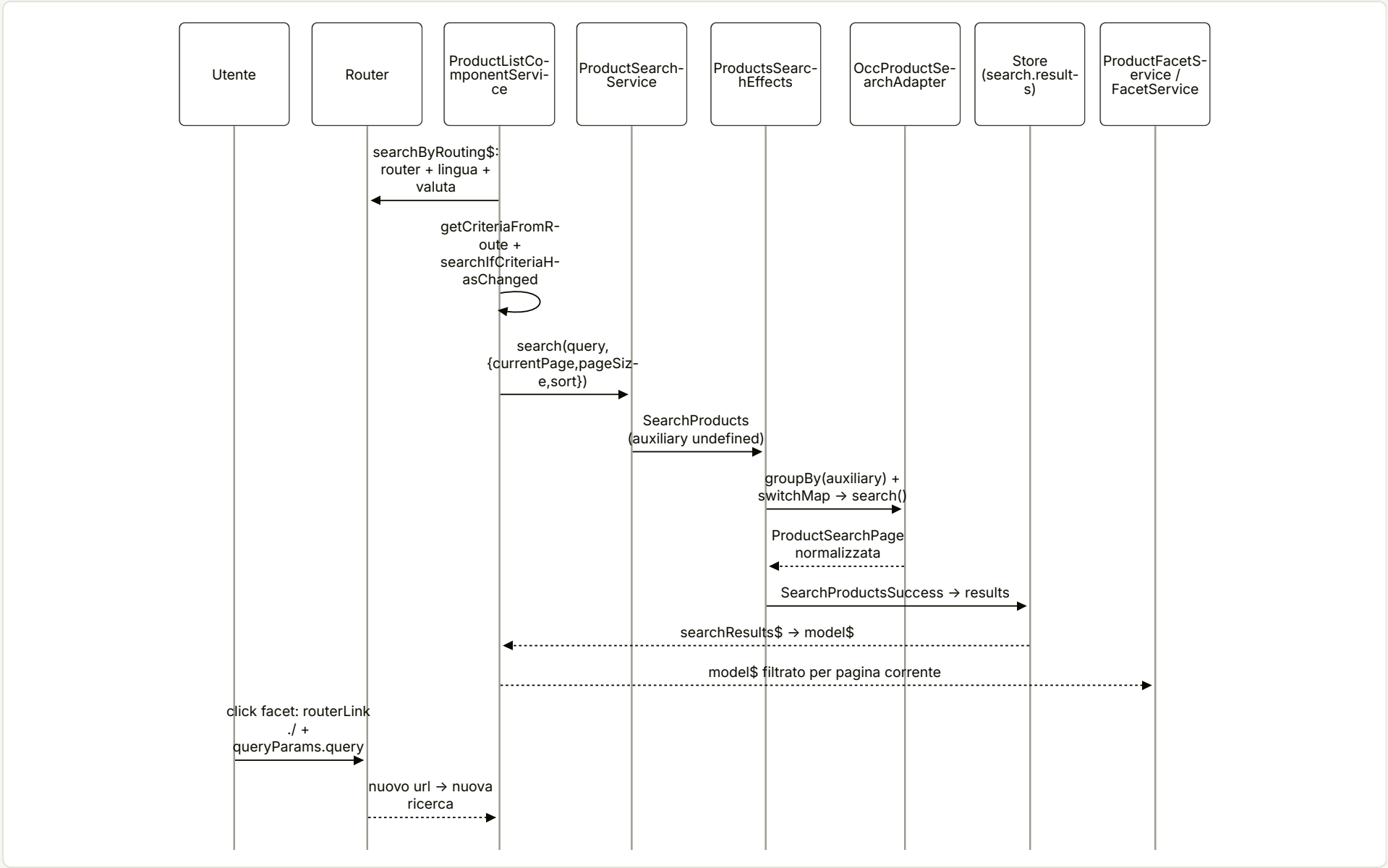


TAVOLA 24.3 **4.3 PLP: route, ricerca, facet**



25

Carrello

`@spartacus/cart`, esposta solo tramite secondary entry point: active cart, saved cart, wish list, quick order, import/export.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Il carrello è la feature più "trafficata" dello storefront: lo tocca la pagina prodotto (add to cart), l'header (mini-cart), la pagina carrello, il checkout, l'account (saved cart, wish list, quick order) e l'import/export CSV. In Spartacus vive nella libreria `@spartacus/cart`, che però **non esporta nulla dal proprio entry point principale**: tutto è esposto da *secondary entry point* — `feature-libs/cart/public_api.ts:L18-L20`:

- `@spartacus/cart/base` — il carrello vero e proprio (attivo, multi-cart, entry, voucher, validazione, guest user, access code, selective cart "save for later");
- `@spartacus/cart/saved-cart` — carrelli salvati;
- `@spartacus/cart/wish-list` — lista desideri (V1 basata su saved cart, V2 opzionale sull'API `/wishlists`);
- `@spartacus/cart/quick-order` — ordine rapido per codice prodotto;
- `@spartacus/cart/import-export` — import/export CSV di entry.

Ognuno ripete lo schema a quattro strati che il libro incontra in tutte le feature lib (capitolo 12):

STRATO	CONTENUTO	CARICAMENTO
`root`	facade astratte, modelli, eventi, config di default, rotte, token	**eager** (nel bundle iniziale)
`core`	implementazioni delle facade, store NgRx, effects, connector, guard, servizi	lazy
`occ`	adapter OCC, endpoint, normalizer	lazy
`components`	componenti CMS e loro servizi	lazy

Il problema di fondo che il codice del carrello risolve è **"quale carrello è quello attivo, e come lo identifico verso il backend"**, in presenza di:

1. utenti anonimi (il carrello si identifica con il `guid`) e loggati (con il `code`) — `feature-libs/cart/base/core/utils/utils.ts:L14-L22`;
2. creazione del carrello pigra (solo al primo add) con un id **temporaneo** lato frontend finché il backend non risponde — `feature-libs/cart/base/core/utils/utils.ts:L68-L88`;
3. persistenza dell'id del carrello anonimo tra un refresh e l'altro in `localStorage` — `feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L33-L62`;
4. merge del carrello anonimo con quello dell'utente al login — `feature-libs/cart/base/core/facade/active-cart.service.ts:L281-L320`;
5. più carrelli contemporaneamente nello store (attivo, wish list, selective, saved cart, carrelli appena creati) — `feature-libs/cart/base/root/models/cart.model.ts:L197-L202`;
6. caricamento lazy della libreria, che deve partire il più tardi possibile ma in tempo — `feature-libs/cart/base/root/cart-base-root.module.ts:L45-L52`.

La risposta architetturale è uno **store NgRx "multi-cart"** (entità indicizzate per id, più un indice "tipo → id") sopra cui stanno due facade: `MultiCartService` (operazioni su qualunque carrello, dato il suo id) e `ActiveCartService` (che sa qual è il carrello attivo e orchestra load/create/merge). Tutto il resto — saved cart, wish list, quick order, import — riusa queste due facade e lo stesso slice di stato.

2. Mappa dei file

PATH	RUOLO
feature-libs/cart/public_api.ts	entry principale vuoto (`export {}`)
feature-libs/cart/base/cart-base.module.ts	`CartBaseModule`: aggrega core + occ + components
feature-libs/cart/base/root/cart-base-root.module.ts	`CartBaseRootModule` (eager): rotta `cart`, `featureModules`, default config
feature-libs/cart/base/root/feature-name.ts	nomi feature `cartBase`, `cartBaseCore`, `miniCart`, `addToCart`
feature-libs/cart/base/root/config/cart-config.ts	`CartConfig` (augment di `Config`)
feature-libs/cart/base/root/config/default-cart-config.ts	default `cart.*`
feature-libs/cart/base/root/config/default-cart-routing-config.ts	rotta `cart` → `['cart']`
feature-libs/cart/base/root/facade/active-cart.facade.ts	`ActiveCartFacade` (facadeFactory, `async: true`)
feature-libs/cart/base/root/facade/multi-cart.facade.ts	`MultiCartFacade`
feature-libs/cart/base/root/facade/cart-voucher.facade.ts	`CartVoucherFacade`
feature-libs/cart/base/root/facade/selective-cart.facade.ts	`SelectiveCartFacade`
feature-libs/cart/base/root/facade/cart-validation.facade.ts	`CartValidationFacade`
feature-libs/cart/base/root/facade/cart-access-code.facade.ts	`CartAccessCodeFacade`
feature-libs/cart/base/root/facade/cart-guest-user.facade.ts	`CartGuestUserFacade`
feature-libs/cart/base/root/models/cart.model.ts	`Cart`, `OrderEntry`, `CartType`, `AbstractOrderKey`, `CartValidationStatusCode`, `PromotionLocation`
feature-libs/cart/base/root/models/augmented.model.ts	augment di `LAUNCH_CALLER` (`CLEAR_CART`, `ADDED_TO_CART`)
feature-libs/cart/base/root/models/cart-outlets.model.ts	enum `CartOutlets`
feature-libs/cart/base/root/models/cart-item-context.model.ts	`CartItemContext` astratto
feature-libs/cart/base/root/tokens/converters.ts	token `CART_NORMALIZER` & co.
feature-libs/cart/base/root/events/cart.events.ts	classi evento del carrello
feature-libs/cart/base/root/events/multi-cart-event.listener.ts	listener degli eventi "comando" `LoadCartEvent`/`RemoveCartEvent`
feature-libs/cart/base/root/events/cart-base-event.module.ts	istanzia il listener (eager)
feature-libs/cart/base/root/context/order-entires.context.ts	`ORDER_ENTRIES_CONTEXT`
feature-libs/cart/base/root/utils/cart-validation.utils.ts	parsing di `statusMessage` (min/max)
feature-libs/cart/base/core/cart-base-core.module.ts	`CartBaseCoreModule`: store, persistenza, eventi, connector, facade, error handler

PATH	RUOLO
feature-libs/cart/base/core/cart-persistence.module.ts	MODULE_INITIALIZER + meta-reducer per il ripristino da storage
feature-libs/cart/base/core/facade/facade-providers.ts	binding facade astratta → implementazione
feature-libs/cart/base/core/facade/active-cart.service.ts	ActiveCartService
feature-libs/cart/base/core/facade/multi-cart.service.ts	MultiCartService
feature-libs/cart/base/core/facade/cart-validation.service.ts	CartValidationService (comando)
feature-libs/cart/base/core/facade/cart-voucher.service.ts	CartVoucherService
feature-libs/cart/base/core/facade/selective-cart.service.ts	SelectiveCartService (save for later)
feature-libs/cart/base/core/facade/cart-guest-user.service.ts	CartGuestUserService
feature-libs/cart/base/core/facade/cart-access-code.service.ts	CartAccessCodeService
feature-libs/cart/base/core/store/multi-cart-state.ts	forma dello stato, MULTI_CART_FEATURE, MULTI_CART_DATA
feature-libs/cart/base/core/store/multi-cart-store.module.ts	StoreModule.forFeature / EffectsModule.forFeature
feature-libs/cart/base/core/store/reducers/index.ts	factory reducer, meta-reducer di logout
feature-libs/cart/base/core/store/reducers/multi-cart.reducer.ts	reducer indice e valori entità
feature-libs/cart/base/core/store/actions/cart.action.ts	azioni load/create/merge/delete
feature-libs/cart/base/core/store/actions/cart-entry.action.ts	azioni add/remove/update entry
feature-libs/cart/base/core/store/actions/cart-voucher.action.ts	azioni voucher
feature-libs/cart/base/core/store/actions/multi-cart.action.ts	SetActiveCartId, ClearCartState, SetCartTypeIndex, …
feature-libs/cart/base/core/store/effects/cart.effect.ts	CartEffects
feature-libs/cart/base/core/store/effects/cart-entry.effect.ts	CartEntryEffects
feature-libs/cart/base/core/store/effects/cart-voucher.effect.ts	CartVoucherEffects
feature-libs/cart/base/core/store/effects/multi-cart.effect.ts	MultiCartEffects (indice)
feature-libs/cart/base/core/store/effects/multi-cart-effect.service.ts	MultiCartEffectsService (pubblico, personalizzabile)
feature-libs/cart/base/core/store/selectors/multi-cart.selector.ts	selettori factory per cartId
feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts	sync {active} ↔ localStorage

PATH	RUOLO
feature-libs/cart/base/core/services/active-cart-state-persistence.service.ts	persistenza entry guest per il merge in code flow
feature-libs/cart/base/core/services/cart-validation-state.service.ts	risultati validazione tra navigazioni
feature-libs/cart/base/core/services/cart-item-validation.service.ts	messaggi min/max per riga
feature-libs/cart/base/core/services/product-import-info.service.ts	esiti add per l'import
feature-libs/cart/base/core/guards/cart-validation.guard.ts	`CartValidationGuard`
feature-libs/cart/base/core/event/cart-event.builder.ts	azioni NgRx → eventi
feature-libs/cart/base/core/event/cart-page-event.builder.ts	`CartPageEvent`
feature-libs/cart/base/core/http-interceptors/handlers/bad-cart-request.handler.ts	`BadCartRequestHandler`
feature-libs/cart/base/core/http-interceptors/handlers/bad-voucher-request.handler.ts	`BadVoucherRequestHandler`
feature-libs/cart/base/core/utils/utils.ts	`getCartIdByUserId`, `isTempCartId`, `isJustLoggedIn`, predicati di errore
feature-libs/cart/base/core/connectors/cart/cart.connector.ts	`CartConnector`
feature-libs/cart/base/core/connectors/cart/cart.adapter.ts	`CartAdapter` astratto
feature-libs/cart/base/occ/cart-base-occ.module.ts	binding adapter OCC e normalizer
feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts	endpoint OCC di default
feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts	`CartOccEndpoints` (augment `OccEndpoints`)
feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts	`OccCartAdapter`
feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts	`OccCartEntryAdapter`
feature-libs/cart/base/occ/adapters/occ-cart-voucher.adapter.ts	`OccCartVoucherAdapter`
feature-libs/cart/base/occ/adapters/occ-cart-validation.adapter.ts	`OccCartValidationAdapter`
feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts	`OccCartNormalizer`
feature-libs/cart/base/occ/adapters/converters/order-entry-promotions-normalizer.ts	`OrderEntryPromotionsNormalizer`
feature-libs/cart/base/components/cart-base-components.module.ts	aggregatore componenti, layout handler, contesto order entries
feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts	`AddToCartComponent`

PATH	RUOLO
`feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog-event.listener.ts`	apertura/chiusura dialog "aggiunto al carrello"
`feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts`	mini-cart senza forzare il lazy load
`feature-libs/cart/base/components/cart-page-layout-handler.ts`	esclusione slot nella `CartPageTemplate`
`feature-libs/cart/base/components/page-context/active-cart-order-entries.context.ts`	`ActiveCartOrderEntriesContext`
`feature-libs/cart/saved-cart/root/saved-cart-root.module.ts`	rotte e feature saved cart
`feature-libs/cart/saved-cart/core/facade/saved-cart.service.ts`	`SavedCartService`
`feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts`	effects saved cart
`feature-libs/cart/saved-cart/occ/adapters/occ-saved-cart.adapter.ts`	`OccSavedCartAdapter`
`feature-libs/cart/wish-list/core/store/effects/wish-list.effect.ts`	effects wish list V1
`feature-libs/cart/wish-list/core/facade/wish-list.service.ts`	`WishListService` (V1)
`feature-libs/cart/wish-list/wish-list-v2.module.ts`	`WishListV2Module` (switch a V2)
`feature-libs/cart/wish-list/core/facade/wish-list-v2.service.ts`	`WishListV2Service`
`feature-libs/cart/quick-order/root/quick-order-root.module.ts`	rotta e feature quick order
`feature-libs/cart/quick-order/core/services/quick-order.service.ts`	`QuickOrderService` (lista in memoria)
`feature-libs/cart/import-export/core/config/default-import-export-config.ts`	default import/export
`feature-libs/cart/import-export/components/import-to-cart/import-entries-dialog/import-entries-dialog.component.ts`	dialog di import
`feature-libs/cart/schematics/add-cart/index.ts`	schematic `ng add`
`projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts`	configurazione lazy nell'app demo

3. Come funziona passo per passo

3.1 Moduli e lazy loading

`CartBaseModule` è il modulo che l'app carica in lazy: importa `CartBaseCoreModule`, `CartBaseOccModule` e `CartBaseComponentsModule` — feature-libs/cart/base/cart-base.module.ts:L7-L15. Nell'app demo `CartBaseFeatureModule` importa il root module e configura tre feature lazy: `cartBase` → `CartBaseWrapperModule` (import dinamico), `miniCart` → `@spartacus/cart/base/components/mini-cart`,

addToCart → @spartacus/cart/base/components/add-to-cart — projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L55.

I nomi delle feature sono quattro: cartBase, cartBaseCore, miniCart, addToCart — feature-libs/cart/base/root/feature-name.ts:L7-L11. La config defaultCartComponentsConfig assegna sei CMS component a cartBase, MiniCartComponent a miniCart, ProductAddToCartComponent a addToCart, e — dettaglio importante — dichiara cartBaseCore come **alias** stringa di cartBase — feature-libs/cart/base/root/cart-base-root.module.ts:L26-L57. Il commento spiega il perché: il core del carrello deve restare nello stesso chunk dei componenti, per ritardare il caricamento della libreria finché davvero serve — feature-libs/cart/base/root/cart-base-root.module.ts:L45-L52. Mini-cart e add-to-cart sono invece entry secondari separati, così l'header e la PDP possono renderizzare senza scaricare subito il core (vedi §3.14).

Il root module, eager, fa tre cose — feature-libs/cart/base/root/cart-base-root.module.ts:L59-L83:

1. importa CartBaseEventModule, che istanzia MultiCartEventListener (§3.10) — feature-libs/cart/base/root/events/cart-base-event.module.ts:L10-L15;
2. registra la rotta CMS con cxRoute: 'cart', CmsPageGuard, PageLayoutComponent e un cxContext che associa ORDER_ENTRIES_CONTEXT al token ActiveCartOrderEntriesContextToken;
3. fornisce i default config (componenti, cart.*, routing). Il path di default della rotta cart è cart — feature-libs/cart/base/root/config/default-cart-routing-config.ts:L9-L15.

CartBaseCoreModule (lazy) importa CartEventModule, MultiCartStoreModule, CartPersistenceModule, CartPageEventModule, fornisce i sei connector, i facadeProviders e due HttpErrorHandler multi (BadCartRequestHandler, BadVoucherRequestHandler) — feature-libs/cart/base/core/cart-base-core.module.ts:L22-L49. CartBaseOccModule registra gli endpoint di default, lega i sei adapter astratti alle implementazioni Occ… con useClass e registra due normalizer multi (CART_NORMALIZER → OccCartNormalizer, ORDER_ENTRY_PROMOTIONS_NORMALIZER → OrderEntryPromotionsNormalizer) — feature-libs/cart/base/occ/cart-base-occ.module.ts:L32-L72. CartBaseComponentsModule registra CartPageLayoutHandler come PAGE_LAYOUT_HANDLER e lega ActiveCartOrderEntriesContextToken a ActiveCartOrderEntriesContext — feature-libs/cart/base/components/cart-base-components.module.ts:L41-L51.

3.2 Facade nel root, implementazioni nel core

Ogni facade del root è una classe astratta providedIn: 'root' con useFactory: facadeFactory(...). ActiveCartFacade dichiara la feature cartBaseCore, 22 metodi proxy e async: true — feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46. Quando un componente eager chiama un metodo della facade, facadeFactory risolve la feature (scaricando il chunk) e inoltra la chiamata all'istanza nel modulo lazy; con async: true aggiunge un delay(0) dopo aver ottenuto l'istanza, "required to make lazy NgRx store feature ready" — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L56, core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L36. Il meccanismo generale è descritto nel capitolo 20.

Lo stesso schema vale per MultiCartFacade (19 metodi, async: true) — feature-libs/cart/base/root/facade/multi-cart.facade.ts:L13-L43; CartVoucherFacade e SelectiveCartFacade usano async: true, mentre CartAccessCodeFacade, CartGuestUserFacade e CartValidationFacade no — feature-libs/cart/base/root/facade/cart-voucher.facade.ts:L12-L28, feature-libs/cart/base/root/facade/selective-cart.facade.ts:L13-L30, feature-libs/cart/base/root/facade/cart-access-code.facade.ts:L12-L20, feature-libs/cart/base/root/facade/cart-guest-user.facade.ts:L13-L21, feature-libs/cart/base/root/facade/cart-validation.facade.ts:L13-L21. La differenza è sensata: le ultime tre sono basate su CommandService/connector e non su store NgRx (§3.12).

Nel core, facadeProviders lega ogni facade astratta all'implementazione con useExisting (es. ActiveCartFacade → ActiveCartService) — feature-libs/cart/base/core/facade/facade-providers.ts:L25-

L61. Nota che `MultiCartFacade.removeCart` rimuove il carrello solo dallo stato; per cancellarlo anche sul backend serve `deleteCart` — feature-libs/cart/base/root/facade/multi-cart.facade.ts:L223-L239.

3.3 Modello dati

`Cart` è un'interfaccia con soli campi opzionali (`code`, `guid`, `entries`, `user`, `saveTime`, `totalItems`, `quoteCode`, ...) — feature-libs/cart/base/root/models/cart.model.ts:L62-L105. Due campi governano gran parte della logica:

- `code` vs `guid`: per l'utente anonimo l'id da usare nelle API è il `guid`, altrimenti il `code` — feature-libs/cart/base/core/utils/utils.ts:L14-L22;
- `saveTime`: se presente, il carrello è un saved cart e **non** può essere il carrello attivo — feature-libs/cart/base/core/store/effects/multi-cart-effect.service.ts:L18-L34, feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L50.

`CartType` elenca i tipi indicizzati nello store: `Active`, `WishList`, `Selective`, `NewCreated` — feature-libs/cart/base/root/models/cart.model.ts:L197-L202. `AbstractOrderKey` modella cart/order/quote/saved cart; per il tipo cart non c'è `id` perché esiste un solo carrello attivo ("current") — feature-libs/cart/base/root/models/cart.model.ts:L124-L150. Altri enum: `CartValidationStatusCode` (`noStock`, `lowStock`, ..., `below_min_quantity`, `exceed_max_quantity`) — feature-libs/cart/base/root/models/cart.model.ts:L208-L216; `PromotionLocation` — feature-libs/cart/base/root/models/cart.model.ts:L25-L31.

Il modello estende anche tipi di altre librerie per *module augmentation*: `LAUNCH_CALLER` di storefront riceve `CLEAR_CART` e `ADDED_TO_CART` (con assegnazione esplicita del valore a runtime, perché un enum aumentato non genera codice) — feature-libs/cart/base/root/models/augmented.model.ts:L7-L16; `Config` di core riceve `CartConfig` — feature-libs/cart/base/root/config/cart-config.ts:L12-L32.

3.4 Lo store multi-cart

La feature NgRx si chiama `cart`. Lo stato ha due rami — feature-libs/cart/base/core/store/multi-cart-state.ts:L10-L27:

- `carts`: un `EntityProcessesLoaderState<Cart | undefined>`, cioè una mappa id → `{ value, loading, success, error, processesCount }` (le utility `StateUtils` sono descritte nel capitolo 19); entity type `'[Multi Cart] Multi Cart Data'`;
- `index`: una mappa `CartType → cartId`.

`MultiCartStoreModule` registra `StoreModule.forFeature('cart', multiCartReducerToken, { metaReducers })` e `EffectsModule.forFeature(effects)` — feature-libs/cart/base/core/store/multi-cart-store.module.ts:L21-L32. La factory del reducer legge il toggle `enableCartSlowNetworkResilience` con `inject(FeatureToggles)` dentro `try/catch` (fallback `false`) e compone `carts` con `entityProcessesLoaderReducer(MULTI_CART_DATA, createCartEntitiesReducer(enabled))` e `index` con `cartTypeIndexReducer` — feature-libs/cart/base/core/store/reducers/index.ts:L34-L53. Il meta-reducer `clearMultiCartState` azzera l'intero slice su `AuthActions.LOGOUT` — feature-libs/cart/base/core/store/reducers/index.ts:L17-L28.

Reducer dell'indice — feature-libs/cart/base/core/store/reducers/multi-cart.reducer.ts:L11-L40: stato iniziale `{ Active: '' }`; `SET_CART_TYPE_INDEX` scrive l'id per tipo; `REMOVE_CART` e `DELETE_CART_SUCCESS` riportano `Active` a `''` se riguardano il carrello attivo; `CLEAR_CART_STATE` ripristina lo stato iniziale. Il valore `''` ha un significato preciso: "nessun id particolare, usa `current`" (§3.7).

Reducer dei valori — feature-libs/cart/base/core/store/reducers/multi-cart.reducer.ts:L79-L87: sostituisce il valore con `payload.cart` su `LOAD_CART_SUCCESS`, `CREATE_CART_SUCCESS`, `SET_CART_DATA`, e con `payload`

su LOAD_CARTS_SUCCESS. Solo con il toggle attivo, CART_ADD_ENTRY_SUCCESS fonde l'entry restituita dal POST nel carrello (sostituendo per entryNumber o appendendo), così la UI la mostra prima del LoadCart di riconciliazione (CXSPA-10582) — feature-libs/cart/base/core/store/reducers/multi-cart.reducer.ts:L42-L65, feature-libs/cart/base/core/store/reducers/multi-cart.reducer.ts:L88-L91.

Processi. Il concetto chiave per capire il carrello è processesCount. Le azioni di entry (CartAddEntry, CartRemoveEntry, CartUpdateEntry) estendono EntityProcessesIncrementAction, i relativi Success/Fail estendono EntityProcessesDecrementAction sullo stesso cartId — feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L22-L176. Un carrello è "stabile" quando !processesCount && !loading, ha "pending processes" quando processesCount > 0 — core-libs/core/src/state/utils/processes-loader/processes-loader.selectors.ts:L9-L17. Altre azioni rilevanti:

- CreateCart è un EntityLoadAction sulla chiave tempCartId (il vero id non è ancora noto), CreateCartSuccess un EntitySuccessAction sul cartId reale — feature-libs/cart/base/core/store/actions/cart.action.ts:L39-L83;
- MergeCart è deprecato a favore di MergeCartAndIncrementProcessesCount, che usa lo stesso type ma incrementa i processi per marcare il carrello come occupato — feature-libs/cart/base/core/store/actions/cart.action.ts:L181-L200; MergeCartAbort li decrementa se il merge non avviene — feature-libs/cart/base/core/store/actions/cart.action.ts:L222-L236;
- ResetCartDetails azzera valori e flag di tutte le entità mantenendone l'elenco; ResetCartDetailsByIds solo per alcuni id — feature-libs/cart/base/core/store/actions/cart.action.ts:L238-L257;
- DeleteCart/DeleteCartFail sono Action semplici, DeleteCartSuccess è EntityRemoveAction — feature-libs/cart/base/core/store/actions/cart.action.ts:L270-L288;
- SetActiveCartId imposta solo l'id attivo ("then services take care of loading that cart"), ClearCartState è un EntityRemoveAllAction, SetCartData un EntitySuccessAction — feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L44-L74;
- i voucher sono un'eccezione: CartAddVoucher/Success/Fail agiscono sull'entità addVoucher del **process store** (PROCESS_FEATURE), non sul carrello; CartRemoveVoucher invece incrementa i processi del carrello — feature-libs/cart/base/core/store/actions/cart-voucher.action.ts:L22-L81.

Le azioni sono esposte nel namespace CartActions — feature-libs/cart/base/core/store/actions/index.ts:L7-L8 — e i selettori nel namespace MultiCartSelectors (factory per cartId: entity, value, isStable, hasPendingProcesses, entries, entry; più getCartIdByTypeFactory) — feature-libs/cart/base/core/store/selectors/index.ts:L7-L8, feature-libs/cart/base/core/store/selectors/multi-cart.selector.ts:L30-L118. getCartEntriesSelectorFactory restituisce [] se il carrello o le entry mancano; getCartEntrySelectorFactory restituisce la prima entry con quel product code — feature-libs/cart/base/core/store/selectors/multi-cart.selector.ts:L73-L91.

3.5 Gli effects

Gli effects registrati sono CartEntryEffects, CartVoucherEffects, CartEffects, MultiCartEffects — feature-libs/cart/base/core/store/effects/index.ts:L12-L17.

loadCart$ è il più sofisticato — feature-libs/cart/base/core/store/effects/cart.effect.ts:L63-L84: raggruppa i LoadCart per cartId (groupBy), così carrelli diversi si caricano in parallelo; dentro ogni gruppo **scarta** i LoadCart mentre il carrello ha processi pendenti (un GET concorrente a POST ancora in volo restituirebbe uno stato vecchio) e usa switchMap per annullare un caricamento precedente dello stesso carrello. In successo emette LoadCartSuccess con cartId ricalcolato da getCartIdByUserId e, se si era chiesto current, anche RemoveCart({ cartId: 'current' }), perché il carrello vive ormai sotto il suo id reale; se il connector restituisce undefined emette LoadCartFail con error: {} — feature-libs/cart/base/core/store/effects/cart.effect.ts:L85-L113. In errore handleLoadCartError distingue tre casi: errori con reason === 'invalid'

(coupon scaduto) → nuovo `LoadCart`; "cart not found" o `UnknownResourceError` → `RemoveCart`; altrimenti `LoadCartFail` con errore normalizzato — feature-libs/cart/base/core/store/effects/cart.effect.ts:L124-L149.

`createCart$` chiama `cartConnector.create(userId, oldCartId, toMergeCartGuid)` ed emette `CreateCartSuccess`, `RemoveCart(tempCartId)` e, se c'era `oldCartId`, `MergeCartSuccess` — feature-libs/cart/base/core/store/effects/cart.effect.ts:L151-L199.

`mergeCart$` carica il carrello `current` dell'utente appena loggato: se il suo `code` è diverso dal carrello anonimo dispatcha `CreateCart` con `oldCartId` = carrello anonimo e `toMergeCartGuid` = guid del carrello utente (il merge vero lo fa il backend); se coincidono emette `CreateCartSuccess` + `MergeCartAbort`; in errore `MergeCartAbort` + `RemoveCart` — feature-libs/cart/base/core/store/effects/cart.effect.ts:L201-L263.

Riconciliazione dopo le mutazioni. `refresh$` su `CART_ADD_VOUCHER_SUCCESS` emette `CartProcessesDecrement` e `LoadCart` — feature-libs/cart/base/core/store/effects/cart.effect.ts:L265-L282. `refreshWithoutProcesses$` reagisce ai Success di add/remove/update entry e remove voucher: con `enableCartSlowNetworkResilience` attende che i processi del carrello scendano a 0 e poi emette **un solo** `LoadCart`; senza toggle emette `LoadCart` subito per ogni success (che però `loadCart$` scarterà finché ci sono processi pendenti) — feature-libs/cart/base/core/store/effects/cart.effect.ts:L304-L349.

Cambio contesto. Su `CURRENCY_CHANGE`/`LANGUAGE_CHANGE` (prezzi e nomi cambiano): senza `enableCartReloadOnContextChange` emette `ResetCartDetails`; con il toggle emette `ResetCartDetailsByIds` per i carrelli non attivi e `LoadCart` del carrello attivo con `userId: 'current'` — feature-libs/cart/base/core/store/effects/cart.effect.ts:L351-L393. Inoltre `loadCart$`, `createCart$`, `mergeCart$`, `addEmail$` e gli effects di entry sono avvolti in `withdrawOn(contextChange$)`, che risottoscrive la sorgente abbandonando le trasformazioni in corso a ogni cambio — feature-libs/cart/base/core/store/effects/cart.effect.ts:L120-L120, core-libs/core/src/util/rxjs/withdraw-on.ts:L10-L27. Il site context è trattato nel capitolo 13.

Altri effects di `CartEffects`: `addEmail$` emette sempre anche `LoadCart` (in successo e in errore) — feature-libs/cart/base/core/store/effects/cart.effect.ts:L395-L434; `deleteCart$` in errore emette `DeleteCartFail` + `LoadCart`, perché il carrello potrebbe essere stato rimosso comunque e, se il load fallisce con NotFound, interverrà `RemoveCart` — feature-libs/cart/base/core/store/effects/cart.effect.ts:L449-L460.

`CartEntryEffects` usa `concatMap` per add/remove/update, cioè **serializza** le chiamate HTTP, e in errore emette `*Fail` + `LoadCart` — feature-libs/cart/base/core/store/effects/cart-entry.effect.ts:L37-L78. `CartAddEntrySuccess` spalma payload e `CartModification` del backend (`deliveryModeChanged`, `entry`, `quantityAdded`, `statusCode`, `statusMessage`) — feature-libs/cart/base/core/store/effects/cart-entry.effect.ts:L55-L61, feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L38-L57. `removeEntry$` mostra il messaggio `cartItems.itemRemoved` tramite un `GlobalMessageService` iniettato `optional` — feature-libs/cart/base/core/store/effects/cart-entry.effect.ts:L33-L35, feature-libs/cart/base/core/store/effects/cart-entry.effect.ts:L92-L99.

`CartVoucherEffects` mostra `voucher.applyVoucherSuccess` / `voucher.removeVoucherSuccess`; in errore di add emette `CartAddVoucherFail`, `CartProcessesDecrement` e `LoadCart` — feature-libs/cart/base/core/store/effects/cart-voucher.effect.ts:L30-L109. Il corrispondente incremento lo fa `MultiCartEffects.processesIncrement$` su `CART_ADD_VOUCHER`, perché `CartAddVoucher` agisce sul process store — feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L20-L29.

`MultiCartEffects` mantiene l'indice. `setSelectiveId$` indicizza come `Selective` un carrello il cui id inizia con `selectivecart` — feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L31-L50. `setActiveCartId$` aggiorna `Active` — feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L52-L138:

AZIONE	NUOVO VALORE DI INDEX.ACTIVE
LOAD_CART con cartId === 'current'	''
LOAD_CART_SUCCESS con extraData.active	'' se il carrello ha saveTime, altrimenti meta.entityId (via MultiCartEffectsService)
CREATE_CART con extraData.active	id temporaneo (meta.entityId)
CREATE_CART_SUCCESS	meta.entityId sotto Active se extraData.active, altrimenti sotto NewCreated
SET_ACTIVE_CART_ID	il payload

La logica su LOAD_CART_SUCCESS è stata estratta in MultiCartEffectsService, pubblico, proprio per permettere la personalizzazione (CXSPA-3551) — feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L115-L121, feature-libs/cart/base/core/store/effects/multi-cart-effect.service.ts:L18-L34.

3.6 MultiCartService

MultiCartService è un @Injectable() senza providedIn (lo fornisce facadeProviders) che opera direttamente sullo Store<StateWithMultiCart> — feature-libs/cart/base/core/facade/multi-cart.service.ts:L33-L40. Punti salienti:

- isStable(cartId) applica un debounce **asimmetrico**: il passaggio a true è ritardato di un tick (timer(0)), quello a false è immediato, per evitare flicker quando molte azioni seguono la fine di un processo — feature-libs/cart/base/core/facade/multi-cart.service.ts:L79-L89;
- l'id temporaneo è temp-<uuid>, generato con crypto.getRandomValues su nativeWindow; il commento avverte che in SSR non funziona — feature-libs/cart/base/core/facade/multi-cart.service.ts:L94-L100;
- createCart dispatcha CreateCart con un nuovo tempCartId e restituisce il carrello indicizzato come Active o NewCreated secondo extraData.active — feature-libs/cart/base/core/facade/multi-cart.service.ts:L107-L139;
- mergeToCurrentCart dispatcha MergeCartAndIncrementProcessesCount con un nuovo tempCartId — feature-libs/cart/base/core/facade/multi-cart.service.ts:L146-L166;
- addEntries dispatcha un CartAddEntry per prodotto, nessuna chiamata bulk — feature-libs/cart/base/core/facade/multi-cart.service.ts:L260-L275;
- updateEntry con quantity <= 0 delega a removeEntry; l'entryNumber viene convertito in stringa nelle azioni — feature-libs/cart/base/core/facade/multi-cart.service.ts:L304-L326;
- reloadCart usa userIdService.takeUserId() e dispatcha LoadCart — feature-libs/cart/base/core/facade/multi-cart.service.ts:L387-L397.

Il JSDoc di isTempCartId riassume il ciclo di vita del carrello temporaneo: è un'entità solo frontend, rappresentata dai flag di creazione, rimossa al tick successivo alla risposta — feature-libs/cart/base/core/utils/utils.ts:L68-L88.

3.7 ActiveCartService: il carrello attivo

ActiveCartService è il cervello della feature. Il suo stato derivato parte da activeCartId$ — feature-libs/cart/base/core/facade/active-cart.service.ts:L58-L68:

1. attende il primo userId con take(1) (non vuole reagire ai cambi utente, gestiti altrove);
2. legge l'id indicizzato Active tramite getCartIdByType;
3. filtra undefined, che significa "non ho ancora ripristinato l'id dallo storage" (§3.8);
4. mappa '' su 'current'.

Il costruttore chiama `initActiveCart()`, `detectUserChange()` e, se sono attivi sia `authorizationCodeFlowByDefault` sia `mergeGuestCartOnCodeFlowLogin`, `persistGuestCartForCodeFlowMerge()` — feature-libs/cart/base/core/facade/active-cart.service.ts:L79-L92.

Lettura e caricamento automatico (`initActiveCart`). Lo stream `cartValue$` emette il carrello solo se stabile **o vuoto**: il commento descrive la sequenza `old_cart -> {} -> new_cart` durante il merge, in cui nessuno deve usare il vecchio id ormai obsoleto — feature-libs/cart/base/core/facade/active-cart.service.ts:L96-L110. Uno stream di side-effect `loading` chiama `load()` quando il carrello è stabile, vuoto, non ancora caricato, l'id non è temporaneo e `shouldLoadCartOnCodeFlow` è vero — feature-libs/cart/base/core/facade/active-cart.service.ts:L112-L126. `activeCart$` lega i due con `using()` (il side-effect vive esattamente quanto la sottoscrizione al valore), normalizza a `{}`, applica `distinctUntilChanged` e `shareReplay({ bufferSize: 1, refCount: true })` — feature-libs/cart/base/core/facade/active-cart.service.ts:L128-L136. Conseguenza pratica: **è sottoscrivere** `getActive()` **a far caricare il carrello**. `load()` non carica mai nel caso anonimo + `current` (non esiste un carrello "corrente" lato server per un anonimo) e marca sempre `extraData.active: true` — feature-libs/cart/base/core/facade/active-cart.service.ts:L262-L275.

Id per le API e attese. `getActiveCartId()` deriva l'id dal carrello attivo (`guid` per l'anonimo, `code` altrimenti) — feature-libs/cart/base/core/facade/active-cart.service.ts:L197-L203. `takeActive()`/`takeActiveCartId()` attendono `isStable()` e restituiscono un solo valore — feature-libs/cart/base/core/facade/active-cart.service.ts:L185-L215. `getPickupEntries`/`getDeliveryEntries` distinguono le entry per presenza di `deliveryPointOfService` — feature-libs/cart/base/core/facade/active-cart.service.ts:L718-L732.

Mutazioni. `addEntry` passa per `requireLoadedCart()` (che crea il carrello se serve), mentre `removeEntry`, `updateEntry`, `addEmail` usano direttamente `activeCartId$` con `take(1)` — feature-libs/cart/base/core/facade/active-cart.service.ts:L548-L630.

`requireLoadedCart()`. Con il toggle `enableCartSlowNetworkResilience` spento ricostruisce la pipeline a ogni chiamata (comportamento "pre-CXSPA-10582"); con il toggle acceso (e non in guest merge) memorizza in `loadedCart$` un observable condiviso con `shareReplay`, così più `addEntry` rapidi non creano più carrelli in parallelo; la cache si azzera dopo l'emissione o su `finalize` — feature-libs/cart/base/core/facade/active-cart.service.ts:L413-L461. La pipeline vera — feature-libs/cart/base/core/facade/active-cart.service.ts:L482-L538:

1. evita load/create se c'è già una creazione in corso (`isCartCreating`);
2. per un utente loggato con carrello vuoto prova a caricare `current` ("might have been created on another device");
3. attende un esito (successo o errore per i loggati; subito per l'anonimo);
4. se il carrello è ancora vuoto chiama `createCart({ active: true })`;
5. attende che l'id attivo punti al codice/guid reale e non più al temporaneo, poi emette il carrello.

`isCartCreating` è vero per un id `temp-` con almeno un flag di loading/success/error; tutti i flag falsi indicano un id temporaneo "sbagliato", ripristinato per esempio da un reload a metà creazione — feature-libs/cart/base/core/facade/active-cart.service.ts:L390-L401.

Cambio utente e merge. `detectUserChange` usa `pairwise()` sullo userId, quindi non scatta mai al bootstrap, e se `isJustLoggedIn` chiama `loadOrMerge` — feature-libs/cart/base/core/facade/active-cart.service.ts:L139-L156. `isJustLoggedIn` è vero se l'utente non è `anonymous` ed è diverso dal precedente (login o inizio emulazione ASM) — feature-libs/cart/base/core/utils/utils.ts:L108-L119. `loadOrMerge` — feature-libs/cart/base/core/facade/active-cart.service.ts:L281-L320:

- se l'id è `current` o l'utente precedente non era anonimo (es. passaggio a ASM) → `loadCart` del carrello dell'utente;
- se c'è uno stato guest persistito (toggle code flow) o il carrello è di un guest → `guestCartMerge`;

- altrimenti → `mergeToCurrentCart`.

`guestCartMerge`, marcato "TODO: Remove once backend is updated", cancella il carrello guest con userId anonimo e ri-aggiunge le sue entry al carrello utente; nel code flow usa le entry persistite e poi le cancella dallo storage — feature-libs/cart/base/core/facade/active-cart.service.ts:L322-L348. Un carrello è "guest" se `user.name === 'guest'` o se la parte di `uid` dopo il primo `|` è un'email — feature-libs/cart/base/core/facade/active-cart.service.ts:L652-L659.

Code flow OAuth. Con l'Authorization Code flow l'app viene ri-bootstrappata dopo il login, quindi `pairwise()` non vede alcuna transizione. Se in `localStorage` è presente la chiave `oAuthRedirectCodeFlow`, il servizio mette `shouldLoadCartOnCodeFlow = false` (non viene mai ripristinato: il caricamento automatico di `initActiveCart` resta disattivato per tutta la vita del servizio) e si sottoscrive a `getUserId()` senza `take(1)` né filtro `isJustLoggedIn`: a **ogni** emissione esegue `loadOrMerge(cartId, userId, 'anonymous')` e rimuove la chiave — feature-libs/cart/base/core/facade/active-cart.service.ts:L158-L172, feature-libs/cart/base/core/facade/active-cart.service.ts:L77-L77, feature-libs/cart/base/core/facade/active-cart.service.ts:L121-L121, feature-libs/cart/base/core/facade/active-cart.service.ts:L403-L408, core-libs/core/src/auth/user-auth/utils/oauth-constants.ts:L7-L7. Il flusso di autenticazione è nel capitolo 22.

3.8 Persistenza del carrello attivo

`CartPersistenceModule` registra un `MODULE_INITIALIZER` che, quando il contesto è stabile (`getStable('context')`), avvia `MultiCartStatePersistenceService.initSync()`, e un `META_REDUCERS` multi — feature-libs/cart/base/core/cart-persistence.module.ts:L19-L74. `MODULE_INITIALIZER` è "un `APP_INITIALIZER` per moduli lazy" (se il modulo è eager si comporta come `APP_INITIALIZER`); i lazy initializer li esegue `LazyModulesService.runModuleInitializersForModule` — core-libs/core/src/lazy-loading/tokens.ts:L12-L20, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L142-L157.

Il meta-reducer `uninitializeActiveCartMetaReducerFactory` su `@ngrx/store/init` imposta `index.Active = undefined`, così `ActiveCartService` non usa `current` prima che l'id sia ripristinato (ricorda il filtro su `undefined` di `activeCartId$`) — feature-libs/cart/base/core/cart-persistence.module.ts:L33-L52. Vedi però la trappola in §9 sul caso lazy.

`initSync` sincronizza in `localStorage`, sotto chiave `cart` e con contesto il base site, solo `{ active: index.Active ?? '' }` — feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L33-L62. La chiave effettiva è `spartacus⚿<baseSite>⚿cart` — core-libs/core/src/state/services/state-persistence.service.ts:L24-L30. `getCartState` scarta le prime emissioni dello store in cui lo slice `cart` non esiste ancora (modulo lazy) — feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L48-L51. In lettura (anche a ogni cambio di base site) `onRead` dispatcha `ClearCartState` e poi `SetActiveCartId(state.active)` oppure `SetActiveCartId('')` — feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L64-L71.

Si persiste **solo l'id**, mai il contenuto: dopo un refresh il carrello si ricarica dal backend tramite il meccanismo di §3.7.

Per il merge guest in code flow, `ActiveCartStatePersistenceService` persiste in `localStorage` (chiave `pendingGuestCartMerge`) le entry del carrello guest ridotte a `{ product: { code }, quantity }`; `clearGuestCartState` rimuove la chiave ricostruendola a mano come `spartacus⚿<ctx>⚿<key>` — feature-libs/cart/base/core/services/active-cart-state-persistence.service.ts:L29-L109. Il meccanismo generico `StatePersistenceService` è descritto nel capitolo 19.

3.9 Connector, adapter, OCC

Lo schema connector/adapter (capitolo 21): `CartConnector` (`providedIn: 'root'`) delega ogni metodo al `CartAdapter` astratto (`loadAll`, `load`, `create`, `delete`, `save`, `addEmail`) — feature-libs/cart/base/core/connectors/cart/cart.connector.ts:L12-L54, feature-libs/cart/base/core/connectors/cart/cart.adapter.ts:L10-L69. `CartAccessCodeConnector` e `CartGuestUserConnector` sono invece `@Injectable()` senza `providedIn` e usano `inject()` — feature-libs/cart/base/core/connectors/access-code/cart-access-code.connector.ts:L11-L13, feature-libs/cart/base/core/connectors/guest-user/cart-guest-user.connector.ts:L12-L14.

Gli endpoint OCC di default sono `carts`, `cart`, `createCart`, `addEntries`, `updateEntries`, `removeEntries`, `addEmail`, `deleteCart`, `cartVoucher`, `cartApplyVoucher`, `cartRemoveVoucher`, `saveCart`, `validate`, `cartAccessCode`, `cartGuestUser` — feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L9-L39; `CartOccEndpoints` augmenta `OccEndpoints` di core con chiavi `string | OccEndpoint` — feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L9-L99.

`OccCartAdapter`:

- `load` per `cartId === 'current'` scarica **tutti** i carrelli (`loadAll`) e prende il primo senza `saveTime`; altrimenti usa l'endpoint `cart`; entrambi passano per `CART_NORMALIZER` — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L35-L60;
- `create` fa POST di `{}` su `createCart` con query param opzionali `oldCartId` e `toMergeCartGuid` — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L62-L87;
- `delete` aggiunge l'header `USE_CLIENT_TOKEN` solo per l'anonimo; `addEmail` lo aggiunge sempre e invia `email` form-urlencoded con PUT — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L89-L145;
- `save` fa PATCH su `saveCart` con `saveCartName`/`saveCartDescription` ed estrae `savedCartData` dalla risposta prima della normalizzazione — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L102-L127.

`OccCartEntryAdapter`: `add` passa `quantity` come url param e, se l'URL generato contiene `quantity=<n>` (caso B2B), usa form-urlencoded con `code` in query; altrimenti POST JSON `{ quantity, product: { code }, deliveryPointOfService? }` — feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L25-L64. `update` usa PUT `{ quantity }` per passare da ritiro in negozio a consegna, altrimenti PATCH; `remove` è un DELETE — feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L86-L125.

`OccCartVoucherAdapter`: `add` usa `POST applyVoucher` con body `{ voucherId }` se `enableApplyVoucherEndpoint`, altrimenti `POST vouchers?voucherId=`; `remove` usa `POST removeVoucher` se `enableRemoveVoucherEndpoint`, altrimenti `DELETE vouchers/<id>` — feature-libs/cart/base/occ/adapters/occ-cart-voucher.adapter.ts:L71-L124. Usa `USE_CLIENT_TOKEN` per l'anonimo e rilancia l'errore normalizzato con `tryNormalizeHttpError` — feature-libs/cart/base/occ/adapters/occ-cart-voucher.adapter.ts:L59-L69, feature-libs/cart/base/occ/adapters/occ-cart-voucher.adapter.ts:L77-L82.

Altri adapter: `OccCartValidationAdapter.validate` fa POST con body `null` e normalizza con `CART_VALIDATION_NORMALIZER` — feature-libs/cart/base/occ/adapters/occ-cart-validation.adapter.ts:L33-L44; `OccCartAccessCodeAdapter` e `OccCartGuestUserAdapter` applicano `backOff({ shouldRetry: isServerError, maxTries: 2 })`, e il guest user adapter usa sempre `USE_CLIENT_TOKEN` — feature-libs/cart/base/occ/adapters/occ-cart-access-code.adapter.ts:L34-L47, feature-libs/cart/base/occ/adapters/occ-cart-guest-user.adapter.ts:L35-L89.

Normalizzazione. I token del root sono `CART_NORMALIZER`, `ORDER_ENTRY_PROMOTIONS_NORMALIZER`, `CART_MODIFICATION_NORMALIZER`, `SAVE_CART_NORMALIZER`, `CART_VOUCHER_NORMALIZER` — feature-libs/cart/base/root/tokens/converters.ts:L17-L35; il core aggiunge `CART_VALIDATION_NORMALIZER`, `CART_ACCESS_CODE_NORMALIZER`, `CART_GUEST_USER_NORMALIZER` — feature-libs/cart/base/core/connectors/

validation/converters.ts:L11-L13, feature-libs/cart/base/core/connectors/access-code/converters.ts:L10-L12, feature-libs/cart/base/core/connectors/guest-user/converters.ts:L11-L13. `ConverterService.pipeable` restituisce l'observable invariato se per un token non è registrato alcun converter — core-libs/core/src/util/converter.service.ts:L80-L89. `OccCartNormalizer` deduplica le quattro liste di promozioni (confronto `JSON.stringify`), mappa `sapQuote` su `quoteCode`/`quotePurchaseOrderNumber` e per ogni entry normalizza `product` (`PRODUCT_NORMALIZER`) e calcola `promotions` (`ORDER_ENTRY_PROMOTIONS_NORMALIZER`) — feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts:L23-L86. `OrderEntryPromotionsNormalizer` assegna a un'entry le promozioni il cui `consumedEntries[].orderEntryNumber` corrisponde all'entry — feature-libs/cart/base/occ/adapters/converters/order-entry-promotions-normalizer.ts:L37-L70.

3.10 Eventi

`CartEvent` (astratto) definisce `cartId` (guid per anonimi, code per loggati), `cartCode` (sempre il code) e `userId` — feature-libs/cart/base/root/events/cart.events.ts:L11-L33. `CartUiEventAddToCart` estende `CxEvent` (non `CartEvent`) e porta `productCode`, `quantity`, `numberOfEntriesBeforeAdd`, `pickupStoreName` e `triggerElementRef` per rifocalizzare il bottone dopo la chiusura del dialog — feature-libs/cart/base/root/events/cart.events.ts:L122-L136.

Gli eventi viaggiano in due direzioni:

- **azioni → eventi**: `CartEventBuilder` (istanziato da `CartEventModule`) registra create/delete/voucher via `StateEventService.register` e add/remove/update entry e merge success via `registerMapped` — feature-libs/cart/base/core/event/cart-event.builder.ts:L60-L69, feature-libs/cart/base/core/event/cart-event.module.ts:L10-L15. `registerMapped` emette l'evento solo se `payload.cartId` coincide con l'id del carrello attivo e usa `switchMap` invece di `withLatestFrom` sullo stream esterno, per non sottoscrivere il carrello (e quindi farlo caricare) finché non arriva l'azione — feature-libs/cart/base/core/event/cart-event.builder.ts:L208-L236. Gli eventi `DeleteCart*` impostano `cartCode = payload.cartId` — feature-libs/cart/base/core/event/cart-event.builder.ts:L136-L166. `CartPageEventBuilder` deriva `CartPageEvent` dai `NavigationEvent` con `semanticRoute === 'cart'` — feature-libs/cart/base/core/event/cart-page-event.builder.ts:L26-L35;
- **eventi → comandi**: `LoadCartEvent` e `RemoveCartEvent` sono ascoltati da `MultiCartEventListener`, che chiama `multiCartFacade.loadCart` (solo se userId e cartId presenti) e `removeCart` — feature-libs/cart/base/root/events/multi-cart-event.listener.ts:L29-L43. Il listener vive nel root eager, così altre feature (checkout, ordini) possono chiedere un reload del carrello senza dipendere dal core del carrello.

Il catalogo eventi è nell'appendice A07.

3.11 Validazione del carrello

`CartValidationService` crea un comando (`CommandStrategy.CancelPrevious`) che attende il carrello stabile e chiama `cartValidationConnector.validate(cartId, userId)` — feature-libs/cart/base/core/facade/cart-validation.service.ts:L27-L44. `CartValidationGuard` — feature-libs/cart/base/core/guards/cart-validation.guard.ts:L44-L102:

1. se `cart.validation.enabled` è falso restituisce `true`;
2. altrimenti valida e salva i risultati in `CartValidationStateService`;
3. se ci sono modifiche sceglie il messaggio (`validation.cartEntryRemoved` se l'unica entry è finita fuori stock, `validation.cartQuantityLimitsViolated` se toggle `cartValidationDisplayBackendMessages` e violazioni min/max, altrimenti `validation.cartEntriesChangeDuringCheckout`) — feature-libs/cart/base/core/guards/cart-validation.guard.ts:L60-L86;

4. lo mostra per `GLOBAL_MESSAGE_TIMEOUT = 10000` ms, ricarica il carrello e restituisce la `UrlTree` della rotta `cart` — feature-libs/cart/base/core/guards/cart-validation.guard.ts:L42-L42, feature-libs/cart/base/core/guards/cart-validation.guard.ts:L88-L99.

Il guard è inserito nei `guards` CMS dei componenti checkout, ad esempio `guards: [CheckoutAuthGuard, CartNotEmptyGuard, CartValidationGuard]` — feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.module.ts:L39-L39 (vedi capitolo 26). `CartValidationStateService` mantiene i risultati in un `ReplaySubject(1)` e li azzera dopo `NAVIGATION_SKIPS = 2` navigazioni, tranne (toggle backend messages) quando si è sulla pagina `cart` — feature-libs/cart/base/core/services/cart-validation-state.service.ts:L16-L54. `CartItemValidationService` è attivo solo con validazione abilitata **e** toggle `cartValidationDisplayBackendMessages`, e memorizza per codice prodotto gli observable min/max — feature-libs/cart/base/core/services/cart-item-validation.service.ts:L26-L42. I limiti si estraggono dal testo: `parseCartModificationQuantityInfo` cerca `Min=<n>/Max=<n>` nello `statusMessage` — feature-libs/cart/base/root/utils/cart-validation.utils.ts:L29-L38; `cartModificationMatchesCode` abbina per `entry.product.code` o, senza entry, cerca `product code <code>` con lookahead `(?!\w)` — feature-libs/cart/base/root/utils/cart-validation.utils.ts:L51-L85.

3.12 Voucher, guest user, access code, selective cart

- `CartVoucherService` senza `cartId` esplicito usa il carrello attivo e legge loading/success/error dal process `addVoucher` — feature-libs/cart/base/core/facade/cart-voucher.service.ts:L55-L98.
- `CartGuestUserService` e `CartAccessCodeService` sono basati su `CommandService`, senza store NgRx — feature-libs/cart/base/core/facade/cart-guest-user.service.ts:L19-L47, feature-libs/cart/base/core/facade/cart-access-code.service.ts:L19-L27.
- `SelectiveCartService` ("save for later") carica, per utente loggato con `customerId`, il carrello `selectivecart<baseSite><customerId>` e lo legge tramite l'indice `Selective` — feature-libs/cart/base/core/facade/selective-cart.service.ts:L46-L73; dipende da `UserProfileFacade` di `@spartacus/user/profile/root` — feature-libs/cart/base/core/facade/selective-cart.service.ts:L20-L41. `CartDetailsComponent.saveForLater` per utente loggato rimuove l'entry dal carrello attivo e la aggiunge al selective cart, altrimenti naviga a `login` — feature-libs/cart/base/components/cart-details/cart-details.component.ts:L110-L120.
- `CartConfigService` espone `isSelectiveCartEnabled()` e `isCartValidationEnabled()` — feature-libs/cart/base/core/services/cart-config.service.ts:L13-L23.

3.13 Gestione degli errori HTTP

Il carrello contribuisce due `HttpErrorHandler` al meccanismo globale (capitolo 41):

- `BadCartRequestHandler` (400, priorità NORMAL) fa match solo se ci sono errori di tipo `CartError`/`CartAddressError`/`CartEntryError`/`CartEntryGroupError` (ignorando `JaloObjectNoLongerValidError`); mostra `httpHandlers.cartNotFound` per i not-found e il `message` del backend (o `httpHandlers.otherCartErrors`) per gli altri — feature-libs/cart/base/core/http-interceptors/handlers/bad-cart-request.handler.ts:L21-L74, feature-libs/cart/base/core/utils/utils.ts:L59-L66. `isCartNotFoundError` esclude i selective cart, per tacere il caso di estensione disabilitata — feature-libs/cart/base/core/utils/utils.ts:L31-L45;
- `BadVoucherRequestHandler` fa match su `VoucherOperationError` e mostra `httpHandlers.voucherExceeded` per `coupon.already.redeemed` e `httpHandlers.invalidCodeProvided` per `coupon.invalid.code.provided` (due filtri indipendenti); per gli altri messaggi non mostra nulla — feature-libs/cart/base/core/http-interceptors/handlers/bad-voucher-request.handler.ts:L32-L70, feature-libs/cart/base/core/http-

interceptors/handlers/bad-voucher-request.handler.ts:L39-L42, feature-libs/cart/base/core/utils/utils.ts:L47-L53.

3.14 Componenti: add-to-cart, dialog, mini-cart, pagina carrello

Add to cart. `ProductAddToCartComponent` è mappato su `AddToCartComponent` con `data.inventoryDisplay: false` — feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts:L34-L45. Il prodotto arriva da `@Input product`, da `productCode`, da `ProductListItemContext` o da `CurrentProductService` (con scope `UNIT + DETAILS` se `showRealTimeStockInPDP` è abilitato) — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L152-L188. Lo stock viene da `ProductAvailabilityService.getRealTimeStock` (real-time stock con `sapUnit.sapCode`) o da `product.stock` — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L203-L228. La quantità scelta viene pubblicata su `CartItemQuantityService` (BehaviorSubject, default 1) perché altre feature (es. OPF Quick Buy) possano riusarla — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L190-L201, feature-libs/cart/base/root/services/cart-item-quantity.service.ts:L10-L21.

`addToCart()` legge prima le entry correnti (`getEntries().pipe(take(1))`), poi chiama `activeCartService.addEntry` e dispatcha `CartUiEventAddToCart` con `numberOfEntriesBeforeAdd`. Il commento spiega il trucco: chiamare `getEntries()` prima garantisce che la lib carrello sia già caricata e quindi che il listener del dialog esista — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L278-L302.

Dialog "aggiunto al carrello". `AddedToCartDialogEventListener` apre il dialog `LAUNCH_CALLER.ADDED_TO_CART` su `CartUiEventAddToCart` e lo chiude su `CartAddEntryFailEvent`; il risultato dell'add arriva al dialog tramite un `ReplaySubject(1)` alimentato da `race(success, fail)` — feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog-event.listener.ts:L32-L82. Il listener è istanziato nel costruttore di `AddedToCartDialogModule` — feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog.module.ts:L50-L54.

Mini-cart. `MiniCartComponentService` evita di caricare la lib carrello finché non serve: interroga `ActiveCartFacade` solo se c'è un carrello attivo in storage, l'utente è loggato o è stato emesso `CreateCartEvent`; lo stream `activeCartRequired` si completa al primo `true` (`takeWhile(..., true)`) — feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L126-L149. Per sapere se c'è un carrello in storage legge direttamente `localStorage` con `StatePersistenceService.readStateFromStorage({ key: 'cart', ... })`, senza passare dallo store — feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L164-L176. `getUpdating()` emette sempre `false` con il toggle `enableCartSlowNetworkResilience` spento, altrimenti `!isStable()` con `debounceTime(250)` — feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L28-L35, feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L107-L124.

Pagina carrello. `CartPageLayoutHandler`, per la `CartPageTemplate`, esclude slot secondo lo stato (carrello vuoto in caricamento, con item, solo selective cart con item, vuoto) — feature-libs/cart/base/components/cart-page-layout-handler.ts:L28-L61. `CartProceedToCheckoutComponent` calcola `cartUpdating$` da `!isStable()` con debounce 250 ms e una "safety valve" di 10 s — feature-libs/cart/base/components/cart-proceed-to-checkout/cart-proceed-to-checkout.component.ts:L34-L40, feature-libs/cart/base/components/cart-proceed-to-checkout/cart-proceed-to-checkout.component.ts:L95-L112. `ClearCartDialogComponentService.deleteActiveCart` chiama `multiCartFacade.deleteCart` e attende `DeleteCartSuccessEvent`/`DeleteCartFailEvent` per chiudere il dialog — feature-libs/cart/base/components/clear-cart/clear-cart-dialog/clear-cart-dialog-component.service.ts:L40-L85. Le righe del carrello espongono 20 outlet `CartOutlets` (es. `cx-cart-item`, `cx-order-summary`, `cx-add-to-cart-container`) — feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L31 — e un `CartItemContext` astratto con sei stream readonly per i componenti in outlet — feature-libs/cart/base/root/models/cart-item-context.model.ts:L17-L29.

3.15 Contesti "order entries"

Import, export e quick order lavorano su "una sorgente di entry" senza sapere se è il carrello attivo, un saved cart o la lista quick order. L'astrazione è `ORDER_ENTRIES_CONTEXT` (un `Symbol`) con tipo `Partial<AddOrderEntriesContext & GetOrderEntriesContext>` — feature-libs/cart/base/root/context/order-entires.context.ts:L10-L14. `AddOrderEntriesContext.type` (`OrderEntriesSource`) permette di riconoscere la sorgente "meglio di instanceof", anche con servizi sovrascritti — feature-libs/cart/base/root/context/add-order-entries.context.ts:L14-L30. Ogni rotta dichiara il proprio token in `data.cxContext` (la rotta `cart` usa `ActiveCartOrderEntriesContextToken`, §3.1). `ActiveCartOrderEntriesContext.addEntries` chiama `activeCartFacade.addEntries` e restituisce `ProductImportInfoService.getResults(cartId)` limitato a `products.length` emissioni — feature-libs/cart/base/components/page-context/active-cart-order-entries.context.ts:L34-L48. `ProductImportInfoService` ascolta `ActionsSubject` per `CART_ADD_ENTRY_SUCCESS/FAIL` del carrello e mappa su `ProductImportInfo` (`lowStock`, `success`, `noStock`, `unknownIdentifier`, `unknownError`) — feature-libs/cart/base/core/services/product-import-info.service.ts:L32-L90.

3.16 Saved cart

Feature `cartSavedCart`/`cartSavedCartCore`; il root module dichiara 5 CMS component con dipendenza da `cartBase` e due rotte protette da `AuthGuard` + `CmsPageGuard` (`savedCartsDetails` con `SavedCartOrderEntriesContextToken`, `savedCarts` con `NewSavedCartOrderEntriesContextToken`) — feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L31-L79; path `my-account/saved-carts` e `my-account/saved-cart/:savedCartId` — feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L83-L95.

Il saved cart **non ha un proprio reducer**: `SavedCartStoreModule` registra solo effects, e i carrelli salvati vivono nello slice multi-cart (`MULTI_CART_DATA`) più process state dedicati — feature-libs/cart/saved-cart/core/store/saved-cart-store.module.ts:L11-L14, feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L45-L54. I process id sono `saved-cart-list`, `saved-cart-restore-cart`, `saved-cart-save-cart`, `saved-cart-clone-cart`, ed `EditSavedCart` riusa `saved-cart-save-cart` — feature-libs/cart/saved-cart/core/store/saved-cart-constants.ts:L7-L10, feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L238-L249.

- `getSavedCartList` filtra dai carrelli dello store quelli con `saveTime`, escludendo wish list (`wishlist<customerId>`) e selective — feature-libs/cart/saved-cart/core/facade/saved-cart.service.ts:L158-L175;
- `get` carica il saved cart se l'entità non ha flag. Dopo un `DeleteSavedCartEvent` salta solo il caricamento: il `return EMPTY` dentro `tap` non ha effetto sullo stream, che continua a emettere quando lo stato ha `success` o `error` (il JSDoc "it won't emit" non corrisponde al codice) — feature-libs/cart/saved-cart/core/facade/saved-cart.service.ts:L69-L95, feature-libs/cart/saved-cart/core/facade/saved-cart.service.ts:L76-L94;
- `deleteSavedCart` riusa `multiCartService.deleteCart` — feature-libs/cart/saved-cart/core/facade/saved-cart.service.ts:L287-L296;
- `loadSavedCarts$` emette `CartActions.LoadCartsSuccess(savedCarts)` + `LoadSavedCartsSuccess` — feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts:L61-L88;
- `restoreSavedCart$`: se il carrello attivo ha entry, non è legato a una quote e ha `code`, prepara un `EditSavedCart` (nome/descrizione vuoti, per non pulire lo stato), chiama il restore e, in successo, emette in sequenza `EditSavedCart`, `LoadCartSuccess` con `extraData.active: true` e `RestoreSavedCartSuccess`: il salvataggio del vecchio carrello attivo avviene **dopo** il restore — feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts:L102-L151, feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts:L116-L126, feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts:L142-L151. `MultiCartEffectsService` indicizza il carrello restituito come `Active` se la risposta

non contiene `saveTime`, altrimenti imposta `Active = ''` — feature-libs/cart/base/core/store/effects/multi-cart-effect.service.ts:L21-L28 (DA VERIFICARE: il contenuto della risposta backend di `restoresavedcart` non è verificabile dal codice);

- `saveCart$` emette `ClearCartState` prima di `LoadCartSuccess`; `editSavedCart$` no — feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts:L167-L255;
- `cloneSavedCart$` in successo emette `CloneSavedCartSuccess`, `RestoreSavedCart({ cartId })` sul carrello d'origine e `LoadSavedCarts` — feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts:L271-L283.

Endpoint: `savedCarts` (`?savedCartsOnly=true`), `savedCart`, `restoreSavedCart`, `cloneSavedCart`, tutti con `/` iniziale — feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L13-L19; `OccSavedCartAdapter` usa GET + `savedCartData`, PATCH per il restore, POST con param `name` per il clone — feature-libs/cart/saved-cart/occ/adapters/occ-saved-cart.adapter.ts:L27-L73.
`NewSavedCartOrderEntriesContext` (import in un nuovo saved cart) crea un carrello con `extraData.active: false`, lo salva, attende la fine del process di salvataggio e poi aggiunge le entry; se tutte le righe falliscono cancella il saved cart — feature-libs/cart/saved-cart/components/page-context/saved-carts-page/new-saved-cart-order-entries.context.ts:L45-L106.

3.17 Wish list

La wish list V1 è un carrello salvato di nome `wishlist<customerId>` — feature-libs/cart/wish-list/core/utils/utils.ts:L10-L12. `loadWishList$` carica tutti i carrelli e cerca quello con quel nome: se c'è emette `LoadWishListSuccess`, altrimenti `CreateWishList`; poi rimuove l'entità temporanea con chiave = nome — feature-libs/cart/wish-list/core/store/effects/wish-list.effect.ts:L79-L121. `createWishList$` crea un carrello e lo salva subito con quel nome — feature-libs/cart/wish-list/core/store/effects/wish-list.effect.ts:L41-L77. Su successo la wish list viene indicizzata come `WishList` e scritta con `SetCartData`; su cambio lingua/valuta viene ricaricata — feature-libs/cart/wish-list/core/store/effects/wish-list.effect.ts:L123-L203. `WishListService.addEntry` aggiunge sempre quantità 1 via `MultiCartFacade` — feature-libs/cart/wish-list/core/facade/wish-list.service.ts:L77-L81.

`WishListV2Module`, opzionale, sovrascrive `WishListFacade` con `WishListV2Service`, che usa l'API `/wishlists` di `@spartacus/user/wishlist` — feature-libs/cart/wish-list/wish-list-v2.module.ts:L20-L50.
`WishListV2Service` mappa la `Wishlist` in un `Cart` fittizio (`entryNumber` = indice, `quantity: 1`); `createWishList` e `loadWishList` sono no-op, `getWishListLoading` emette sempre `false` — feature-libs/cart/wish-list/core/facade/wish-list-v2.service.ts:L82-L137.

3.18 Quick order

`QuickOrderRootModule` registra la rotta `quickOrder` (`my-account/quick-order`) con contesto `QuickOrderOrderEntriesContextToken` e i CMS component `QuickOrderComponent`, `CartQuickOrderFormComponent` — feature-libs/cart/quick-order/root/quick-order-root.module.ts:L23-L69. La config ha default `searchForm { displayProductImages: true, maxProducts: 5, minCharactersBeforeRequest: 3 }` e `list.hardDeleteTimeout: 240_000` — feature-libs/cart/quick-order/root/config/default-quick-order.config.ts:L9-L20.

`QuickOrderService` tiene la lista in un `BehaviorSubject<OrderEntry[]>`, non nello store — feature-libs/cart/quick-order/core/services/quick-order.service.ts:L39-L52. `addToCart` raccoglie gli eventi di add parziale/fallito, chiama `activeCartService.addEntries`, svuota la lista e completa quando il carrello è stabile — feature-libs/cart/quick-order/core/services/quick-order.service.ts:L177-L213. Le righe eliminate sono soft-deleted e rimosse dopo `hardDeleteTimeout` — feature-libs/cart/quick-order/core/services/quick-order.service.ts:L283-L305. La lista è persistita in `sessionStorage` (chiave `quick-order`), con

`initSync()` chiamato da `QuickOrderComponent` — feature-libs/cart/quick-order/core/services/quick-order-state-persistance.service.ts:L33-L50, feature-libs/cart/quick-order/components/quick-order/quick-order.component.ts:L100-L100. `QuickOrderOrderEntriesContext.addEntries` verifica il limite con `canAdd`, scarica il prodotto via `ProductConnector` e restituisce `LIMIT_EXCEEDED` oltre il limite — feature-libs/cart/quick-order/components/page-context/quick-order-order-entries.context.ts:L49-L82.

3.19 Import/export

`ImportExportRootModule` dichiara la feature `cartImportExport` con 3 CMS component; `ImportExportCoreModule.forRoot()` fornisce il default config — feature-libs/cart/import-export/root/import-export-root.module.ts:L11-L26, feature-libs/cart/import-export/core/import-export-core.module.ts:L11-L19. Default: separatore `,`, file max 1 MB, max entry per sorgente (100 per saved cart nuovi/esistenti, 10 per carrello attivo e quick order), export con colonne extra e file `cart.csv` — feature-libs/cart/import-export/core/config/default-import-export-config.ts:L11-L61.

`ImportOrderEntriesComponent` legge il contesto della pagina con `ContextService.get(ORDER_ENTRIES_CONTEXT)` e apre il dialog `IMPORT_TO_CART` passando `{ orderEntriesContext }` — feature-libs/cart/import-export/components/import-to-cart/import-entries/import-order-entries.component.ts:L41-L50; il dialog è registrato come launch config `inlineRoot` di tipo `DIALOG_TYPE.DIALOG` — feature-libs/cart/import-export/components/import-to-cart/default-import-entries-layout.config.ts:L10-L18. Il dialog invoca `context.addEntries(products, savedCartInfo)` e aggrega i `ProductImportInfo` in un riepilogo — feature-libs/cart/import-export/components/import-to-cart/import-entries-dialog/import-entries-dialog.component.ts:L81-L114. `ImportProductsFromCsvService` interpreta ogni riga come `[productCode, quantity]` — feature-libs/cart/import-export/components/import-to-cart/import-products-from-csv.service.ts:L18-L28.

3.20 Schematic di installazione

La schematic `add-cart` valida l'installazione Spartacus, analizza le dipendenze tra feature e aggiunge feature e peerDependencies — feature-libs/cart/schematics/add-cart/index.ts:L25-L43. Lo schema generale delle schematics di Spartacus è nel capitolo 43.

---

4. Diagrammi

4.1 Architettura a strati del carrello base

■ **TAVOLA 25.1** diagramma a tutta pagina, in fondo al capitolo

Ogni freccia corrisponde a un fatto di §3.1–§3.9: binding `useExisting` (feature-libs/cart/base/core/facade/facade-providers.ts:L25-L61), dispatch dallo store (feature-libs/cart/base/core/facade/multi-cart.service.ts:L33-L40), effects → connector (feature-libs/cart/base/core/store/effects/cart.effect.ts:L83-L84), connector → adapter (feature-libs/cart/base/core/connectors/cart/cart.connector.ts:L12-L54), persistenza (feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L33-L71).

4.2 Bootstrap e ripristino del carrello attivo (anonimo, refresh)

■ TAVOLA 25.2 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/cart/base/core/cart-persistence.module.ts:L19-L52, feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L64-L71, feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L81-L85, feature-libs/cart/base/core/facade/active-cart.service.ts:L58-L68, feature-libs/cart/base/core/facade/active-cart.service.ts:L112-L126, feature-libs/cart/base/core/store/effects/cart.effect.ts:L63-L113. La risoluzione della feature attende il completamento dei `MODULE_INITIALIZER` del modulo lazy (e quindi `initSync()`, chiamato nel `tap` della Promise `lastValueFrom(getStable('context')…)`) prima di restituire il `moduleRef` da cui `facadeFactory` ottiene l'istanza: `FeatureModulesService.resolveFeature` → `LazyModulesService.resolveModuleInstance` → `runModuleInitializersForModule` (`Promise.all` degli initializer, poi `of(moduleRef)`) → `injector.get(facadeClass)` + `delay(0)` — core-libs/core/src/lazy-loading/feature-modules.service.ts:L58-L65, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L98, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L158-L171, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L55, feature-libs/cart/base/core/cart-persistence.module.ts:L19-L31. Il filtro su `undefined` in `activeCartId$` rende comunque il flusso robusto se il meta-reducer ha effetto (vedi §9).

4.3 Add to cart con carrello inesistente

■ TAVOLA 25.3 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L278-L302, feature-libs/cart/base/core/facade/active-cart.service.ts:L482-L560, feature-libs/cart/base/core/facade/multi-cart.service.ts:L107-L139, feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L52-L138, feature-libs/cart/base/core/store/effects/cart.effect.ts:L151-L199, feature-libs/cart/base/core/store/effects/cart-entry.effect.ts:L37-L78, feature-libs/cart/base/core/event/cart-event.builder.ts:L208-L236, feature-libs/cart/base/core/store/effects/cart.effect.ts:L304-L349, feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog-event.listener.ts:L32-L82.

4.4 Login: decisione tra load, guest merge e merge

■ TAVOLA 25.4 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/cart/base/core/facade/active-cart.service.ts:L139-L156, feature-libs/cart/base/core/utils/utils.ts:L108-L119, feature-libs/cart/base/core/facade/active-cart.service.ts:L281-L348, feature-libs/cart/base/core/store/effects/cart.effect.ts:L151-L263.

---

5. Estratti di codice reale

5.1 Alias `cartBaseCore` e rotta `cart`

feature-libs/cart/base/root/cart-base-root.module.ts:L39-L75

```ts
      [MINI_CART_FEATURE]: {
        cmsComponents: ['MiniCartComponent'],
      },
      [ADD_TO_CART_FEATURE]: {
        cmsComponents: ['ProductAddToCartComponent'],
      },
      // By default core is bundled together with components.
      // The cart lib should keep using this default.
      //
      // While the lazy loading configurations make it possible to
      // split the core part and the components part, it is required that
      // they stay together for the cart lib.  This compromise is required to
      // optimize performances by delaying the moment the cart lib is loaded and
      // making sure cart lib is loaded when needed.
      [CART_BASE_CORE_FEATURE]: CART_BASE_FEATURE,
    },
  };
  return config;
}

@NgModule({
  imports: [
    CartBaseEventModule,
    RouterModule.forChild([
      {
        // @ts-ignore
        path: null,
        canActivate: [CmsPageGuard],
        component: PageLayoutComponent,
        data: {
          cxRoute: 'cart',
          cxContext: {
            [ORDER_ENTRIES_CONTEXT]: ActiveCartOrderEntriesContextToken,
          },
        },
      },
    ]),
```

5.2 Forma dello stato

feature-libs/cart/base/core/store/multi-cart-state.ts:L10-L27

```ts
export const MULTI_CART_FEATURE = 'cart';
export const MULTI_CART_DATA = '[Multi Cart] Multi Cart Data';

/**
 * Add voucher process const
 */
```

```
export const ADD_VOUCHER_PROCESS_ID = 'addVoucher';

export interface StateWithMultiCart {
  [MULTI_CART_FEATURE]: MultiCartState;
}

export interface MultiCartState {
  carts: StateUtils.EntityProcessesLoaderState<Cart | undefined>;
  index: {
    [cartType: string]: string;
  };
}
```

5.3 Reducer dell'indice

feature-libs/cart/base/core/store/reducers/multi-cart.reducer.ts:L11-L40

TS

```
export const cartTypeIndexInitialState = { [CartType.ACTIVE]: '' };
export function cartTypeIndexReducer(
  state: {
    [cartType: string]: string;
  } = cartTypeIndexInitialState,
  action: CartActions.MultiCartActions | CartActions.CartAction
): {
  [cartType: string]: string;
} {
  switch (action.type) {
    case CartActions.SET_CART_TYPE_INDEX:
      return {
        ...state,
        [action.payload.cartType]: action.payload.cartId as string,
      };
    case CartActions.REMOVE_CART:
    case CartActions.DELETE_CART_SUCCESS: {
      if (action.payload?.cartId === state[CartType.ACTIVE]) {
        return {
          ...state,
          [CartType.ACTIVE]: '',
        };
      }
      return state;
    }
    case CartActions.CLEAR_CART_STATE:
      return cartTypeIndexInitialState;
  }
  return state;
}
```

5.4 `loadCart$`: groupBy, filtro sui processi, switchMap

feature-libs/cart/base/core/store/effects/cart.effect.ts:L63-L84

```ts
> = createEffect(() =>
  this.actions$.pipe(
    ofType(CartActions.LOAD_CART),
    map((action: CartActions.LoadCart) => action.payload),
    groupBy((payload) => payload.cartId),
    mergeMap((group$) =>
      group$.pipe(
        switchMap((payload) => {
          return of(payload).pipe(
            withLatestFrom(
              this.store.pipe(
                select(
                  getCartHasPendingProcessesSelectorFactory(payload.cartId)
                )
              )
            )
          );
        }),
        filter(([_, hasPendingProcesses]) => !hasPendingProcesses),
        map(([payload]) => payload),
        switchMap((payload) => {
          return this.cartConnector.load(payload.userId, payload.cartId).pipe(
```

5.5 `activeCartId$`: l'id del carrello attivo

feature-libs/cart/base/core/facade/active-cart.service.ts:L58-L68

```ts
// This stream is used for referencing carts in API calls.
protected activeCartId$ = this.userIdService.getUserId().pipe(
  // We want to wait the initialization of cartId until the userId is initialized
  // We have take(1) to not trigger this stream, when userId changes.
  take(1),
  switchMap(() => this.multiCartFacade.getCartIdByType(CartType.ACTIVE)),
  // We also wait until we initialize cart from localStorage
  filter((cartId) => cartId !== undefined),
  // fallback to current when we don't have particular cart id
  map((cartId) => (cartId === '' ? OCC_CART_ID_CURRENT : cartId))
);
```

5.6 Caricamento automatico legato alla sottoscrizione

feature-libs/cart/base/core/facade/active-cart.service.ts:L112-L136

```ts
    // Responsible for loading cart when it does not exist (eg. app initialization when we have only cartId)
    const loading = cartValue$.pipe(
      withLatestFrom(this.activeCartId$, this.userIdService.getUserId()),
      tap(([{ cart, loaded, isStable }, cartId, userId]) => {
        if (
          isStable &&
          isEmpty(cart) &&
          !loaded &&
          !isTempCartId(cartId) &&
          this.shouldLoadCartOnCodeFlow
        ) {
          this.load(cartId, userId);
        }
      })
    );

    this.activeCart$ = using(
      () => loading.subscribe(),
      () => cartValue$
    ).pipe(
      // Normalization for empty cart value returned as empty object.
      map(({ cart }) => (cart ? cart : {})),
      distinctUntilChanged(),
      shareReplay({ bufferSize: 1, refCount: true })
    );
```

5.7 loadOrMerge

feature-libs/cart/base/core/facade/active-cart.service.ts:L281-L320

```ts
  protected loadOrMerge(
    cartId: string,
    userId: string,
    previousUserId: string
  ): void {
    if (
      cartId === OCC_CART_ID_CURRENT ||
      // It covers the case when you are logged in and then asm user login, you don't want to merge, but only load emulated user cart
      // Similarly when you are logged in as asm user and you logout and want to resume previous user session
      previousUserId !== OCC_USER_ID_ANONYMOUS
    ) {
      this.multiCartFacade.loadCart({
        userId,
        cartId,
        extraData: {
```

```
          active: true,
        },
      });
    } else if (
      this.featureToggles.authorizationCodeFlowByDefault &&
      this.featureToggles.mergeGuestCartOnCodeFlowLogin &&
      this.activeCartStatePersistenceService.readGuestCartState()
    ) {
      // Authorization-code flow: the SPA was re-bootstrapped on login, so the
      // in-memory guest cart (and thus `isGuestCart()`) is gone. Fall back to
      // the entries we persisted before the redirect.
      this.guestCartMerge(cartId);
    } else if (getLastValueSync(this.isGuestCart())) {
      this.guestCartMerge(cartId);
    } else {
      // We have particular cart locally, but we logged in, so we need to combine this with current cart
or make it ours.
      this.multiCartFacade.mergeToCurrentCart({
        userId,
        cartId,
        extraData: {
          active: true,
        },
      });
    }
  }
```

5.8 Id temporaneo e `isStable` antiflicker

feature-libs/cart/base/core/facade/multi-cart.service.ts:L79-L100

TS

```
  isStable(cartId: string): Observable<boolean> {
    return this.store.pipe(
      select(MultiCartSelectors.getCartIsStableSelectorFactory(cartId)),
      // We dispatch a lot of actions just after finishing some process or loading, so we want this flag
not to flicker.
      // This flickering should only be avoided when switching from false to true
      // Start of loading should be showed instantly (no debounce)
      // Extra actions are only dispatched after some loading
      debounce((isStable) => (isStable ? timer(0) : of(undefined))),
      distinctUntilChanged()
    );
  }

  /**
   * Simple random temp cart id generator
   */
  protected generateTempCartId(): string {
```

```
    const array = new Uint32Array(1);
    // This will not work in SSR because 'window' is only available in the browser (client-side).
    this.windowRef.nativeWindow?.crypto.getRandomValues(array);
    const pseudoUuid = array[0].toString(36).substring(2, 11);
    return `temp-${pseudoUuid}`;
  }
```

5.9 Persistenza: initializer e meta-reducer

feature-libs/cart/base/core/cart-persistence.module.ts:L19-L52

TS

```ts
export function cartStatePersistenceFactory(
  cartStatePersistenceService: MultiCartStatePersistenceService,
  configInit: ConfigInitializerService
): () => Promise<Config> {
  return () =>
    lastValueFrom(
      configInit.getStable('context').pipe(
        tap(() => {
          cartStatePersistenceService.initSync();
        })
      )
    );
}

/**
 * Before `MultiCartStatePersistenceService` restores the active cart id `ActiveCartService`
 * will use `current` cart instead of the one saved in browser. This meta reducer
 * sets the value on store initialization to undefined cart which holds active cart loading
 * until the data from storage is restored.
 */
export function uninitializeActiveCartMetaReducerFactory(): MetaReducer<any> {
  const metaReducer =
    (reducer: ActionReducer<any>) => (state: any, action: Action) => {
      const newState = { ...state };
      if (action.type === '@ngrx/store/init') {
        newState.cart = {
          ...newState.cart,
          ...{ index: { [CartType.ACTIVE]: undefined } },
        };
      }
      return reducer(newState, action);
    };
  return metaReducer;
}
```

5.10 `OccCartAdapter.load` e `create`

feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L87

```ts
  public load(userId: string, cartId: string): Observable<Cart | undefined> {
    if (cartId === OCC_CART_ID_CURRENT) {
      return this.loadAll(userId).pipe(
        map((carts) => carts.find((cart) => cart['saveTime'] === undefined))
      );
    } else {
      return this.http
        .get<Occ.Cart>(
          this.occEndpointsService.buildUrl('cart', {
            urlParams: { userId, cartId },
          })
        )
        .pipe(this.converterService.pipeable(CART_NORMALIZER));
    }
  }

  create(
    userId: string,
    oldCartId?: string,
    toMergeCartGuid?: string
  ): Observable<Cart> {
    const toAdd = JSON.stringify({});

    const params = <any>{};

    if (oldCartId) {
      params['oldCartId'] = oldCartId;
    }
    if (toMergeCartGuid) {
      params['toMergeCartGuid'] = toMergeCartGuid;
    }
    // ...
```

5.11 `registerMapped`: evento solo per il carrello attivo

feature-libs/cart/base/core/event/cart-event.builder.ts:L208-L236

```ts
  protected registerMapped<T>(mapping: ActionToEventMapping<T>): () => void {
    const eventStream$ = this.getAction(mapping.action).pipe(
      switchMap((action) => {
        // SwitchMap was used instead of withLatestFrom, because we only want to subscribe to cart stream when action is dispatched.
        // Using withLatestFrom would trigger subscription to cart observables on event subscription and that causes side effects,
        // such as loading cart when we don't yet need it.
```

```
      return of(action).pipe(
        withLatestFrom(
          this.activeCartService.getActive(),
          this.activeCartService.getActiveCartId()
        )
      );
    }),
    filter(
      ([action, _activeCart, activeCartId]) =>
        action.payload['cartId'] === activeCartId
    ),
    map(([action, activeCart]) =>
      createFrom(mapping.event as Type<T>, {
        ...action.payload,
        cartCode: activeCart.code,
        entry: action.payload.entry
          ? action.payload.entry
          : activeCart.entries?.[Number(action.payload.entryNumber)],
      })
    )
  );
  return this.event.register(mapping.event as Type<T>, eventStream$);
}
```

5.12 Mini-cart: quando serve davvero la facade

feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L136-L149

```ts
protected activeCartRequired(): Observable<boolean> {
  return combineLatest([
    this.hasActiveCartInStorage(),
    this.authService.isUserLoggedIn(),
    this.isCartCreated(),
  ]).pipe(
    map(
      ([hasCartInStorage, isUserLoggedIn, isCartCreated]) =>
        hasCartInStorage || isUserLoggedIn || isCartCreated
    ),
    distinctUntilChanged(),
    takeWhile((hasCart) => !hasCart, true)
  );
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cart.validation.enabled	boolean	false	feature-libs/cart/base/root/config/default-cart-config.ts:L9-L21	feature-libs/cart/base/core/services/cart-config.service.ts:L20-L22 → feature-libs/cart/base/core/guards/cart-validation.guard.ts:L45-L45
cart.selectiveCart.enabled	boolean	false	feature-libs/cart/base/root/config/default-cart-config.ts:L9-L21	feature-libs/cart/base/core/services/cart-config.service.ts:L16-L18; feature-libs/cart/base/components/cart-page-layout-handler.ts:L63-L67; feature-libs/cart/base/components/cart-details/cart-details.component.ts:L81-L81
cart.showRealTimeStockInPDP.enabled	boolean	false	feature-libs/cart/base/root/config/default-cart-config.ts:L9-L21	feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L123-L124
routing.routes.cart.paths	string[]	['cart']	feature-libs/cart/base/root/config/default-cart-routing-config.ts:L9-L15	rotta cxRoute: 'cart' feature-libs/cart/base/root/cart-base-root.module.ts:L69-L69
featureModules.cartBase / miniCart / addToCart / cartBaseCore	CmsConfig	vedi §3.1	feature-libs/cart/base/root/cart-base-root.module.ts:L26-L57	facadeFactory (feature cartBaseCore)
backend.occ.endpoints.{carts, cart,createCart,…}	string | OccEndpoint	vedi §3.9	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L9-L39	adapter OCC (buildUrl)
backend.occ.endpoints.{savedCarts,savedCart, restoreSavedCart, cloneSavedCart}	string	vedi §3.16	feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L9-L23	feature-libs/cart/saved-cart/occ/adapters/occ-saved-cart.adapter.ts:L75-L102
routing.routes.savedCarts / savedCartsDetails	RoutingConfig	my-account/saved-carts, my-account/saved-cart/:savedCartId	feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L83-L95	rotte del root module
routing.routes.quickOrder	RoutingConfig	my-account/quick-order	feature-libs/cart/quick-order/root/quick-order-root.module.ts:L36-L44	rotta cxRoute: 'quickOrder'
quickOrder.searchForm.{displayProductImages,maxProducts,minCharactersBeforeRequest}	bool / number / number	true / 5 / 3	feature-libs/cart/quick-order/root/config/default-quick-order.config.ts:L9-L20	maxProducts: passato dalla Config iniettata in feature-libs/cart/quick-order/components/quick-order/form/quick-order-form.component.ts:L325-L328 (il default statico è solo fallback in feature-libs/cart/quick-order/core/services/quick-order.service.ts:L78-L82); minCharactersBeforeRequest in feature-libs/cart/quick-order/components/quick-order/form/quick-order-form.component.ts:L305-L307; displayProductImages in feature-libs/cart/quick-order/components/quick-order/form/quick-order-form.component.html:L85-L85

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`quickOrder.list.hardDeleteTimeout`	number	`240_000` (fallback nel codice `7000`)	`feature-libs/cart/quick-order/root/config/default-quick-order.config.ts:L16-L18`	`feature-libs/cart/quick-order/core/services/quick-order.service.ts:L49-L50`
CMS `QuickOrderComponent.data.quickOrderListLimit`	number	`10`	`feature-libs/cart/quick-order/components/quick-order/quick-order-list.module.ts:L55-L58`	`feature-libs/cart/quick-order/components/quick-order/quick-order.component.ts:L60-L62` (tipo in `feature-libs/cart/quick-order/core/models/cms.model.ts:L9-L11`)
`cartImportExport.*`	ImportExportConfig	vedi §3.19	`feature-libs/cart/import-export/core/config/default-import-export-config.ts:L11-L61`	componenti import/export
CMS `ProductAddToCartComponent.data.inventoryDisplay`	boolean	`false`	`feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts:L35-L44`	`feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L102-L103`
toggle `enableCartSlowNetworkResilience`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L583-L583` (default `:L1084-L1084`)	`feature-libs/cart/base/core/store/reducers/index.ts:L37-L37`; `feature-libs/cart/base/core/store/effects/cart.effect.ts:L323-L323`; `feature-libs/cart/base/core/facade/active-cart.service.ts:L426-L426`; `feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L108-L108`
toggle `enableCartReloadOnContextChange`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L475-L475` (default `:L1073-L1073`)	`feature-libs/cart/base/core/store/effects/cart.effect.ts:L46-L47`
toggle `authorizationCodeFlowByDefault`	boolean	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L161-L161` (default `:L1029-L1029`)	`feature-libs/cart/base/core/facade/active-cart.service.ts:L87-L87`, `:L300-L300`, `:L332-L332`
toggle `mergeGuestCartOnCodeFlowLogin`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L698-L698` (default `:L1099-L1099`)	`feature-libs/cart/base/core/facade/active-cart.service.ts:L88-L88`, `:L301-L301`, `:L333-L333`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `cartValidationDisplayBackendMessages`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L853-L853` (default `:L1106-L1106`)	`feature-libs/cart/base/core/services/cart-item-validation.service.ts:L40-L40`; `feature-libs/cart/base/core/services/cart-validation-state.service.ts:L44-L44`; `feature-libs/cart/base/core/guards/cart-validation.guard.ts:L61-L61`
toggle `enableApplyVoucherEndpoint`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L406-L406` (default `:L1068-L1068`)	`feature-libs/cart/base/occ/adapters/occ-cart-voucher.adapter.ts:L74-L74`
toggle `enableRemoveVoucherEndpoint`	boolean	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L413-L413` (default `:L1069-L1069`)	`feature-libs/cart/base/occ/adapters/occ-cart-voucher.adapter.ts:L102-L102`

Chiavi di storage usate dalla feature:

CHIAVE	STORAGE	CONTENUTO	RIFERIMENTO
`spartacus⚿<baseSite>⚿cart`	localStorage	`{ active: string }`	`feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L33-L62`, `core-libs/core/src/state/services/state-persistence.service.ts:L24-L30`
`spartacus⚿<baseSite>⚿pendingGuestCartMerge`	localStorage	entry guest `{ product: { code }, quantity }`	`feature-libs/cart/base/core/services/active-cart-state-persistence.service.ts:L29-L109`
`oAuthRedirectCodeFlow`	localStorage	marcatore di login in code flow	`core-libs/core/src/auth/user-auth/utils/oauth-constants.ts:L7-L7`
`quick-order` (con contesto base site)	sessionStorage	lista quick order	`feature-libs/cart/quick-order/core/services/quick-order-state-persistance.service.ts:L33-L50`

Per i feature toggle in generale vedi il capitolo 42 e l'appendice A09; per gli endpoint l'appendice A04.

7. Commenti e note notevoli dal sorgente

- `@deprecated since 221121.1` su `MergeCart`: usare `MergeCartAndIncrementProcessesCount` — `feature-libs/cart/base/core/store/actions/cart.action.ts:L181-L185`.
- `// TODO: Remove once backend is updated` su `guestCartMerge`, "Temporary method … because of backend limitation" — `feature-libs/cart/base/core/facade/active-cart.service.ts:L322-L326`. Il JSDoc del toggle `mergeGuestCartOnCodeFlowLogin` precisa che il backend rifiuta il merge nativo di un carrello guest ("Cart is not anonymous") — `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L690-L698`.

- `// TODO: Make cart required param in 4.0` su `isGuestCart`: un TODO di molte major fa, mai chiuso — feature-libs/cart/base/core/facade/active-cart.service.ts:L639-L639.
- `// TODO(#7241)` su `refresh$` e `processesIncrement$`: il voucher come process è un debito tecnico riconosciuto — feature-libs/cart/base/core/store/effects/cart.effect.ts:L265-L265, feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L20-L20.
- Il lungo commento CXSPA-10582 su `refreshWithoutProcesses$` spiega perché il filtro di `loadCart$` da solo non basta: su 3G lento "rapid multi-product adds finished POSTing server-side but the UI showed a stale cart" — feature-libs/cart/base/core/store/effects/cart.effect.ts:L284-L303.
- Commento sul "pre-CXSPA-10582 codepath" in `requireLoadedCart`, mantenuto per i clienti che ne hanno fatto override — feature-libs/cart/base/core/facade/active-cart.service.ts:L422-L425. Il JSDoc del toggle elenca le classi impattate (ActiveCartService, CartEffects, MultiCartReducer, MiniCart, CartDetails, ProceedToCheckout, Place Order) — core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L578-L583.
- `// This will not work in SSR because 'window' is only available in the browser (client-side).` — feature-libs/cart/base/core/facade/multi-cart.service.ts:L96-L96.
- "Extracted small portion of private effect's logic to a public service to allow FSA for customizing it (for more, see CXSPA-3551)" — feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L118-L119.
- JSDoc di `AddToCartComponent.product`: "As long as we do not support #5026, we require product input" — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L84-L88.
- `// TODO(#14059): Remove condition` in `QuickOrderService.searchProducts` — feature-libs/cart/quick-order/core/services/quick-order.service.ts:L76-L76.
- `// TODO: for future releases, refactor this part to thrown errors` in `SavedCartService` — feature-libs/cart/saved-cart/core/facade/saved-cart.service.ts:L63-L65.
- `// TODO: Inline this factory when we start releasing Ivy compiled libraries` — feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L30-L30.
- `// CHECK SONAR` a fine file — feature-libs/cart/base/occ/adapters/converters/order-entry-promotions-normalizer.ts:L73-L73.

8. Extension point

PUNTO	COME SI ESTENDE	DOVE	RICETTA
Facade astratte (`ActiveCartFacade`, `MultiCartFacade`, `SelectiveCartFacade`, `CartVoucherFacade`, `CartValidationFacade`, `CartAccessCodeFacade`, `CartGuestUserFacade`)	provider custom per la facade	feature-libs/cart/base/root/public_api.ts:L11-L11	Ricette: configurazione, DI, interceptor, eventi
Implementazioni core (`ActiveCartService`, `MultiCartService`, …) con metodi `protected`	sottoclasse + `useExisting`/`useClass` sulla facade	feature-libs/cart/base/core/public_api.ts:L11-L11	Ricette: feature lib, carrello/checkout

PUNTO	COME SI ESTENDE	DOVE	RICETTA
Adapter astratti (`CartAdapter`, `CartEntryAdapter`, `CartVoucherAdapter`, `CartValidationAdapter`, `CartAccessCodeAdapter`, `CartGuestUserAdapter`, `SavedCartAdapter`)	adapter custom per un backend non OCC	feature-libs/cart/base/core/public_api.ts:L9-L9, feature-libs/cart/saved-cart/core/connectors/saved-cart.adapter.ts:L10-L35	Ricette: OCC, adapter, converter
Adapter OCC (`OccCartAdapter`, `OccCartEntryAdapter`, `OccCartVoucherAdapter`, …)	sottoclasse con override dei metodi	feature-libs/cart/base/occ/public_api.ts:L7-L7	Ricette: OCC, adapter, converter
Converter multi (`CART_NORMALIZER`, `ORDER_ENTRY_PROMOTIONS_NORMALIZER`, `CART_MODIFICATION_NORMALIZER`, `SAVE_CART_NORMALIZER`, `CART_VOUCHER_NORMALIZER`, `CART_VALIDATION_NORMALIZER`, …)	`{ provide: CART_NORMALIZER, useExisting: MyNormalizer, multi: true }`	feature-libs/cart/base/root/tokens/converters.ts:L17-L35, feature-libs/cart/base/root/public_api.ts:L15-L15	Ricette: OCC, adapter, converter
Endpoint OCC (campi aggiuntivi, URL)	`provideConfig({ backend: { occ: { endpoints: { cart: … } } } })`	feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L97-L99	Ricette: OCC, adapter, converter
`CartConfig`, `QuickOrderConfig`, `ImportExportConfig`	`provideConfig`	feature-libs/cart/base/root/config/cart-config.ts:L12-L32, feature-libs/cart/quick-order/root/config/quick-order-config.ts:L10-L29, feature-libs/cart/import-export/core/config/import-export-config.ts:L12-L26	Ricette: configurazione

PUNTO	COME SI ESTENDE	DOVE	RICETTA
Outlet `CartOutlets.*` e `CartItemContext`	`provideOutlet` / template in outlet, iniettando `CartItemContext`	feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L31, feature-libs/cart/base/root/models/cart-item-context.model.ts:L17-L29	Ricette: CMS, outlet, layout
Componenti CMS del carrello (`CartComponent`, `ProductAddToCartComponent`, `MiniCartComponent`, …)	mapping `cmsComponents`	feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts:L34-L45	Ricette: CMS, outlet, layout
`ORDER_ENTRIES_CONTEXT` e token per pagina	nuovo contesto che implementa `AddOrderEntriesContext`/`GetOrderEntriesContext`, dichiarato in `data.cxContext`	feature-libs/cart/base/root/context/order-entires.context.ts:L10-L14, feature-libs/cart/base/root/tokens/context.ts:L9-L11	Ricette: feature lib, carrello/checkout
`MultiCartEffectsService.getActiveCartTypeOnLoadSuccess`	override del servizio pubblico	feature-libs/cart/base/core/store/effects/multi-cart-effect.service.ts:L18-L34, feature-libs/cart/base/core/public_api.ts:L15-L15	Ricette: feature lib, carrello/checkout
`CartActions`, `MultiCartSelectors`	effects/selettori custom sullo slice `cart`	feature-libs/cart/base/core/public_api.ts:L14-L17	API pubblica vs interna

PUNTO	COME SI ESTENDE	DOVE	RICETTA
Eventi (`CartAddEntrySuccessEvent`, `CartUiEventAddToCart`, `CartPageEvent`, comandi `LoadCartEvent`/`RemoveCartEvent`)	`EventService.get(...)` / `dispatch(...)`	`feature-libs/cart/base/root/events/cart.events.ts:L11-L136`, `feature-libs/cart/base/root/events/multi-cart-event.listener.ts:L29-L43`	Ricette: eventi
`HttpErrorHandler` (`BadCartRequestHandler`, `BadVoucherRequestHandler`)	handler custom multi o sottoclasse	`feature-libs/cart/base/core/cart-base-core.module.ts:L37-L46`	Ricette: interceptor
`PAGE_LAYOUT_HANDLER` (`CartPageLayoutHandler`)	handler multi aggiuntivo	`feature-libs/cart/base/components/cart-base-components.module.ts:L42-L46`	Ricette: CMS, outlet, layout
`LAUNCH_CALLER` augmentato (`CLEAR_CART`, `ADDED_TO_CART`, `SAVED_CART`, `IMPORT_TO_CART`)	override del `launch` layout config dei dialog	`feature-libs/cart/base/root/models/augmented.model.ts:L7-L16`, `feature-libs/cart/saved-cart/root/model/augmented-core.model.ts:L7-L15`, `feature-libs/cart/import-export/core/model/augmented-core.model.ts:L9-L15`	Ricette: CMS, outlet, layout
`WishListV2Module`	importarlo per passare all'API `/wishlists`	`feature-libs/cart/wish-list/wish-list-v2.module.ts:L35-L50`	Ricette: feature lib, carrello/checkout
Tipi aumentabili `Cart`, `DeliveryMode`, `OrderEntry`	`declare module '@spartacus/cart/base/root'`	`feature-libs/cart/base/root/public_api.ts:L18-L20`	Ricette: feature lib, carrello/checkout

Due consigli che derivano direttamente dal codice: se si fa override di `ActiveCartService.requireLoadedCart`, ricordare che con il toggle slow-network il metodo usa una cache (`loadedCart$`) e delega a `buildRequireLoadedCartPipeline` — feature-libs/cart/base/core/facade/active-cart.service.ts:L413-L469; e per personalizzare l'indicizzazione del carrello attivo conviene `MultiCartEffectsService` anziché gli effects, i cui metodi rilevanti sono `private` — feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L115-L121. La distinzione tra API pubblica e interna è trattata in 54 — upgrade-safe.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Sottoscrivere `getActive()` causa un GET.** Il side-effect di caricamento è legato alla sottoscrizione via `using()` — feature-libs/cart/base/core/facade/active-cart.service.ts:L128-L136. Un componente che "dà solo un'occhiata" al carrello forza il lazy load e la richiesta HTTP. Per questo `CartEventBuilder` usa `switchMap` e il mini-cart verifica prima lo storage — feature-libs/cart/base/core/event/cart-event.builder.ts:L211-L213, feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L126-L149.
2. `activeCartId$` **non reagisce ai cambi utente.** Il `take(1)` sullo userId è voluto; i cambi passano solo da `detectUserChange` — feature-libs/cart/base/core/facade/active-cart.service.ts:L60-L62.
3. `LoadCart` **scartati in silenzio.** `loadCart$` ignora i `LoadCart` mentre il carrello ha processi pendenti; senza il toggle slow-network un burst di add può lasciare il carrello non riconciliato — feature-libs/cart/base/core/store/effects/cart.effect.ts:L81-L81, commento feature-libs/cart/base/core/store/effects/cart.effect.ts:L292-L296. Un `reloadActiveCart()` lanciato durante un add non avrà effetto.
4. **Il commento cita una riga sbagliata**: `refreshWithoutProcesses$` rimanda a `cart.effect.ts:80` per il filtro, che in realtà è a riga 81 — feature-libs/cart/base/core/store/effects/cart.effect.ts:L286-L287 vs feature-libs/cart/base/core/store/effects/cart.effect.ts:L81-L81.
5. `current` **non è un id.** Per un anonimo `load()` con `current` non fa nulla — feature-libs/cart/base/core/facade/active-cart.service.ts:L266-L266; per un loggato `OccCartAdapter.load('current')` scarica *tutti* i carrelli e sceglie il primo senza `saveTime` — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L47-L50 — e poi `loadCart$` rimuove l'entità `current` — feature-libs/cart/base/core/store/effects/cart.effect.ts:L96-L104.
6. **Id temporaneo in SSR.** `generateTempCartId` dipende da `window.crypto`; in SSR `nativeWindow` è `undefined`, l'array resta a `0` e dal codice si deduce che l'id diventi `temp-` seguito da stringa vuota (`'0'.toString(36).substring(2, 11)` è `''`), quindi uguale per ogni creazione — feature-libs/cart/base/core/facade/multi-cart.service.ts:L94-L100. `DA VERIFICARE` a runtime: dedotto dal codice, non eseguito (nessun `node_modules`).
7. **Il meta-reducer di "disinizializzazione" e il lazy loading.** `uninitializeActiveCartMetaReducerFactory` scatta solo su `@ngrx/store/init` — feature-libs/cart/base/core/cart-persistence.module.ts:L39-L52. In NgRx `META_REDUCERS` viene combinato in `_RESOLVED_META_REDUCERS` dentro `_provideStore`, cioè nei provider dello store **root** — deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L1337-L1353. Con `CartBaseCoreModule` caricato lazy il provider `META_REDUCERS` sta nell'injector del modulo lazy. `DA VERIFICARE` se il meta-reducer abbia effetto in configurazione lazy (non verificato a runtime; dossier T17/Q1). Il reducer dell'indice parte comunque da `{ Active: '' }` — feature-libs/cart/base/core/store/reducers/multi-cart.reducer.ts:L11-L11.
8. **Voucher e processi.** `CartAddVoucher` agisce sul process store, non sul carrello: l'incremento dei processi del carrello lo fa un effect separato e il decremento avviene in `refresh$` o in caso di errore — feature-libs/cart/base/core/store/actions/cart-voucher.action.ts:L22-L81, feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L20-L29, feature-libs/cart/base/core/store/effects/cart.

effect.ts:L265-L282. Un effect custom che dispatcha `CartAddVoucher` senza passare dagli effects standard può lasciare `processesCount` sbilanciato.

9. **Rilevamento B2B dall'URL.** `OccCartEntryAdapter.add` decide il formato (form-urlencoded vs JSON) controllando se l'URL contiene `quantity=<n>`: funziona solo se l'endpoint `addEntries` contiene `${quantity}` in query — feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L32-L37. Personalizzando l'endpoint si può cambiare formato senza volerlo.
10. `USE_CLIENT_TOKEN` **non uniforme.** `delete` cart e voucher lo usano per l'anonimo, `OccCartEntryAdapter.remove` no — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L89-L101, feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L107-L125.
11. **Endpoint con `/` iniziale (falso allarme).** `saveCart` e tutti gli endpoint saved cart iniziano con `/`, a differenza degli altri — feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L30-L30, feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L14-L18 — ma l'effetto è nullo: `buildUrl` unisce base URL ed endpoint con `urlPathJoin`, che normalizza gli slash di ogni parte (`cleanSlashes`) — core-libs/core/src/occ/services/occ-endpoints.service.ts:L233-L238, core-libs/core/src/occ/utils/occ-url-util.ts:L12-L32. Vedi capitolo 21.
12. `cloneSavedCart$` **ripristina l'originale**, non il clone: dopo il clone emette `RestoreSavedCart` sul `cartId` di partenza — feature-libs/cart/saved-cart/core/store/effects/saved-cart.effect.ts:L267-L283. DA VERIFICARE la semantica backend di `clonesavedcart` (non verificabile senza backend).
13. **Process id condiviso.** `EditSavedCart` riusa il process `saved-cart-save-cart`: loading/success/error di save ed edit si sovrappongono — feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L238-L249.
14. `SaveCartFailEvent.type` **è** `'SaveCartEvent'`, identico a quello di `SaveCartEvent` — feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L35-L39 (cfr. feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L20-L20).
15. **Optional chaining mancante.** `ProductImportInfoService.mapMessages` accede a `error?.details[0]` (se `details` è `undefined` lancia `TypeError`) — feature-libs/cart/base/core/services/product-import-info.service.ts:L76-L76; analogamente `QuickOrderOrderEntriesContext.handleErrors` con `errors[0]` — feature-libs/cart/quick-order/components/page-context/quick-order-order-entries.context.ts:L117-L117.
16. **Fallback divergente.** `hardDeleteTimeout` ha default `240_000` ma fallback nel codice `7000` — feature-libs/cart/quick-order/core/services/quick-order.service.ts:L49-L50.
17. `distinctUntilChanged` **inefficace.** `SelectiveCartService.getCart` (e `WishListService.getWishList`) applicano `distinctUntilChanged()` all'array prodotto da `combineLatest`, che è sempre un nuovo riferimento — feature-libs/cart/base/core/facade/selective-cart.service.ts:L46-L54, feature-libs/cart/wish-list/core/facade/wish-list.service.ts:L48-L53.
18. **Chiave di storage ricostruita a mano.** `clearGuestCartState` replica il formato `spartacus⁓…` di `StatePersistenceService`: se il formato cambiasse, la rimozione fallirebbe — feature-libs/cart/base/core/services/active-cart-state-persistence.service.ts:L98-L109.
19. `WishListV2Service.wishlistV2$` usa `shareReplay({ refCount: false })`: la sottoscrizione HTTP resta viva per tutta la vita del servizio — feature-libs/cart/wish-list/core/facade/wish-list-v2.service.ts:L38-L53.
20. `getLastEntry` **con ramo morto.** `filteredEntries` è sempre un array (truthy), quindi il ramo `: undefined` non è mai raggiunto; con array vuoto si ottiene comunque `undefined` — feature-libs/cart/base/core/facade/multi-cart.service.ts:L215-L222.
21. **Logout = slice azzerato.** Su `AuthActions.LOGOUT` sparisce anche la wish list e qualunque saved cart caricato, non solo il carrello attivo — feature-libs/cart/base/core/store/reducers/index.ts:L17-L28.

10. Come fare debug

1. **Ispezionare lo slice** `cart` dello store (ad esempio con Redux DevTools, se l'app lo integra): `cart.index.Active` dice quale id usa `ActiveCartService` (`undefined` = non ancora ripristinato, `''` = current, `temp-…` = creazione in corso); `cart.carts.entities[<id>]` mostra `loading`, `success`, `error`, `processesCount` — feature-libs/cart/base/core/store/multi-cart-state.ts:L10-L27, feature-libs/cart/base/core/facade/active-cart.service.ts:L58-L68. Un `processesCount` che non torna a 0 spiega un carrello "mai stabile" (e quindi `takeActive()` che non emette mai) — feature-libs/cart/base/core/facade/active-cart.service.ts:L185-L215.
2. **Seguire le azioni** per prefisso: `[Cart] …` e `[Cart-entry] …` — feature-libs/cart/base/core/store/actions/cart.action.ts:L12-L37, feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L11-L20, feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L12-L20. Sequenze attese: add su carrello nuovo = `Create Cart` → `Create Cart Success` + `Remove Cart` → `Add Entry` → `Add Entry Success` → `Load Cart` (§4.3); merge = `Merge Cart` → `Create Cart` → `Create Cart Success` + `Remove Cart` + `Merge Cart Success` (§4.4).
3. **Controllare il** `localStorage`: `spartacus⚿<baseSite>⚿cart` deve contenere `{"active":"…"}` con il guid per l'anonimo o il code per il loggato — core-libs/core/src/state/services/state-persistence.service.ts:L24-L30. Se il carrello anonimo "si perde" al refresh, verificare che la chiave esista per il base site giusto.
4. **Network**: un `GET users/<userId>/carts` (non `carts/<id>`) senza `savedCartsOnly=true` indica un `load('current')` oppure il caricamento della wish list V1 (`loadWishList$` usa `loadAll`); con `savedCartsOnly=true` è la lista dei saved cart — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L35-L50, feature-libs/cart/wish-list/core/store/effects/wish-list.effect.ts:L88-L90, feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L14-L15; un `POST …/carts?oldCartId=…&toMergeCartGuid=…` indica un merge — feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L62-L87.
5. **Feature toggle**: prima di indagare su race condition o carrello "stale", verificare `enableCartSlowNetworkResilience` ed `enableCartReloadOnContextChange` (§6), perché cambiano reducer, effects e `requireLoadedCart`.
6. **Eventi**: sottoscrivere `EventService.get(CartAddEntrySuccessEvent)` / `CartAddEntryFailEvent`. Se un evento non arriva, ricordare che `registerMapped` filtra per `cartId === activeCartId` — feature-libs/cart/base/core/event/cart-event.builder.ts:L221-L224: un add su wish list o saved cart non emette eventi "di carrello attivo".
7. **Validazione al checkout**: se il checkout rimanda alla pagina carrello, il colpevole tipico è `CartValidationGuard`; i risultati restano in `CartValidationStateService` per due navigazioni — feature-libs/cart/base/core/guards/cart-validation.guard.ts:L88-L99, feature-libs/cart/base/core/services/cart-validation-state.service.ts:L16-L54.
8. **Test unitari**: la libreria ha un `vitest.config.ts` (`feature-libs/cart/vitest.config.ts`); il comando `nx test-vitest <library-name>` è documentato in `.claude/CLAUDE.md` del repo. Il tooling di test è trattato nel capitolo 44.

11. Domande di autoverifica

1. **Perché** `cartBaseCore` **è configurato come alias di** `cartBase`**?** Perché il core del carrello deve restare nello stesso chunk dei componenti: così la lib si carica il più tardi possibile ma sicuramente quando serve —

feature-libs/cart/base/root/cart-base-root.module.ts:L45-L53.

2. **Che cosa significano i valori** `undefined`, `''` **e** `temp-xxx` **di** `index.Active`**?** `undefined`: id non ancora ripristinato dallo storage (`activeCartId$` attende); `''`: nessun id particolare, si usa `current`; `temp-xxx`: creazione in corso, id solo frontend — feature-libs/cart/base/core/facade/active-cart.service.ts:L58-L68, feature-libs/cart/base/core/cart-persistence.module.ts:L33-L52, feature-libs/cart/base/core/utils/utils.ts:L68-L88.
3. **Perché per l'utente anonimo l'id del carrello nelle API è il** `guid`**?** Perché `getCartIdByUserId` restituisce `guid` per `anonymous` e `code` altrimenti: il code di un carrello anonimo non basta a identificarlo in modo sicuro lato API — feature-libs/cart/base/core/utils/utils.ts:L14-L22. (Il motivo di sicurezza è un'interpretazione: DA VERIFICARE, non è scritto nel codice.)
4. **Due click rapidi su "aggiungi" con carrello inesistente: perché non si creano due carrelli?** `requireLoadedCart` scarta load/create se `isCartCreating` è vero; con il toggle slow-network, inoltre, le chiamate concorrenti condividono lo stesso observable in cache (`loadedCart$`) — feature-libs/cart/base/core/facade/active-cart.service.ts:L482-L491, feature-libs/cart/base/core/facade/active-cart.service.ts:L413-L461.
5. **Perché le chiamate di add/update/remove entry sono serializzate?** Perché `CartEntryEffects` usa `concatMap` — feature-libs/cart/base/core/store/effects/cart-entry.effect.ts:L45-L45.
6. **Al login, quando Spartacus carica invece di fare merge?** Quando l'id attivo è `current` (nessun carrello anonimo) o l'utente precedente non era anonimo (es. passaggio ad ASM) — feature-libs/cart/base/core/facade/active-cart.service.ts:L286-L298.
7. **Chi esegue davvero il merge dei carrelli?** Il backend: `mergeCart$` dispatcha `CreateCart` con `oldCartId` e `toMergeCartGuid`, che `OccCartAdapter.create` passa come query param del POST — feature-libs/cart/base/core/store/effects/cart.effect.ts:L218-L227, feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L71-L76. Fa eccezione il carrello guest, unito lato frontend da `guestCartMerge` — feature-libs/cart/base/core/facade/active-cart.service.ts:L322-L348.
8. **Cosa viene salvato nel** `localStorage` **per il carrello attivo?** Solo `{ active: <id> }` sotto `spartacus⚿<baseSite>⚿cart`, mai il contenuto — feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L33-L62, core-libs/core/src/state/services/state-persistence.service.ts:L24-L30.
9. **Come fa il mini-cart a non scaricare la libreria carrello per un visitatore senza carrello?** Interroga `ActiveCartFacade` solo se c'è un id in storage, l'utente è loggato o è stato emesso `CreateCartEvent`, e legge lo storage direttamente con `StatePersistenceService` — feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L136-L176.
10. **Perché un saved cart caricato con** `extraData.active` **non diventa il carrello attivo?** Perché `MultiCartEffectsService.getActiveCartTypeOnLoadSuccess` imposta `Active = ''` se il carrello ha `saveTime` — feature-libs/cart/base/core/store/effects/multi-cart-effect.service.ts:L21-L28.

TAVOLA 25.1 **4.1 Architettura a strati del carrello base**

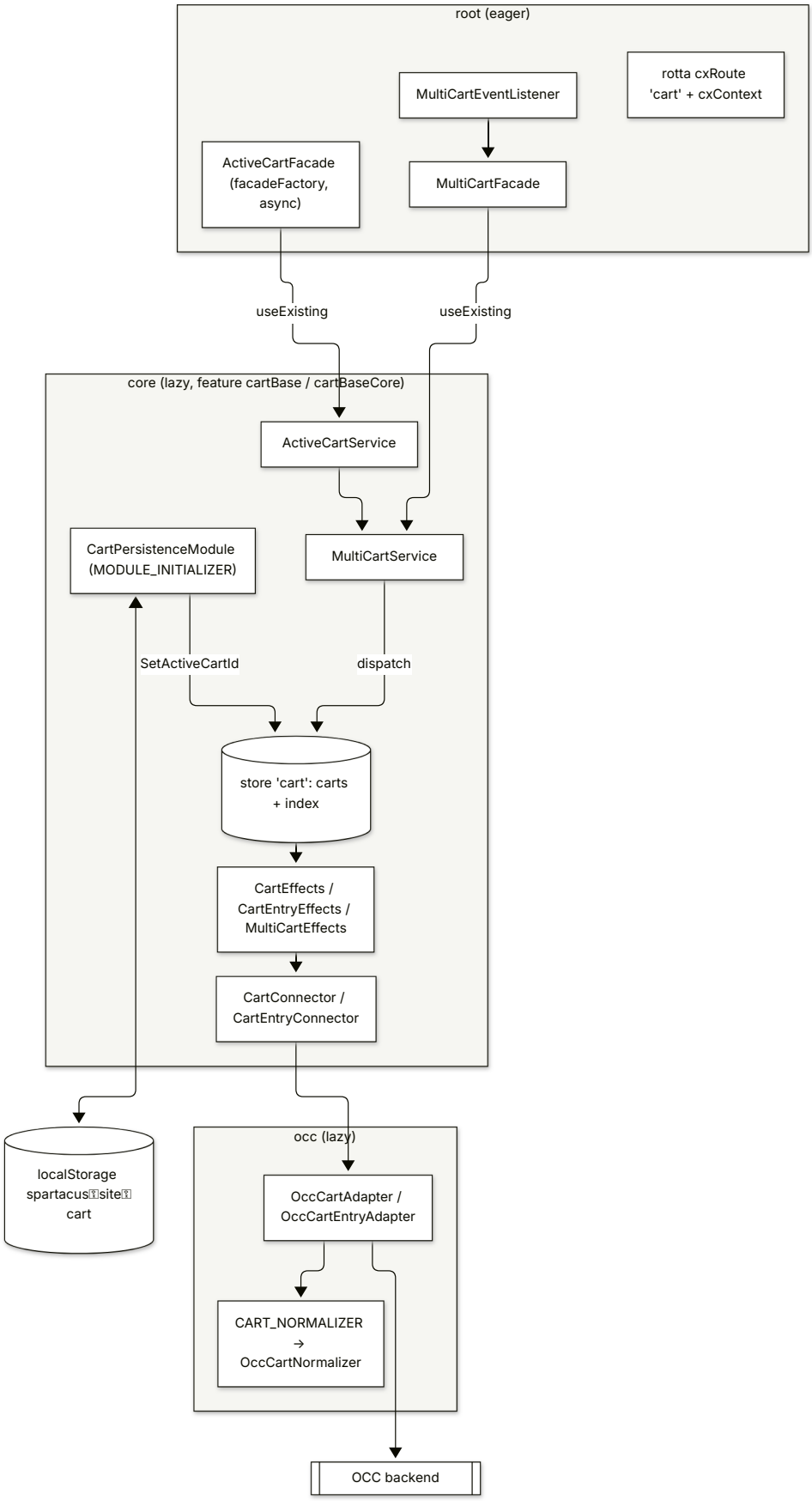


TAVOLA 25.2 **4.2 Bootstrap e ripristino del carrello attivo (anonimo, refresh)**

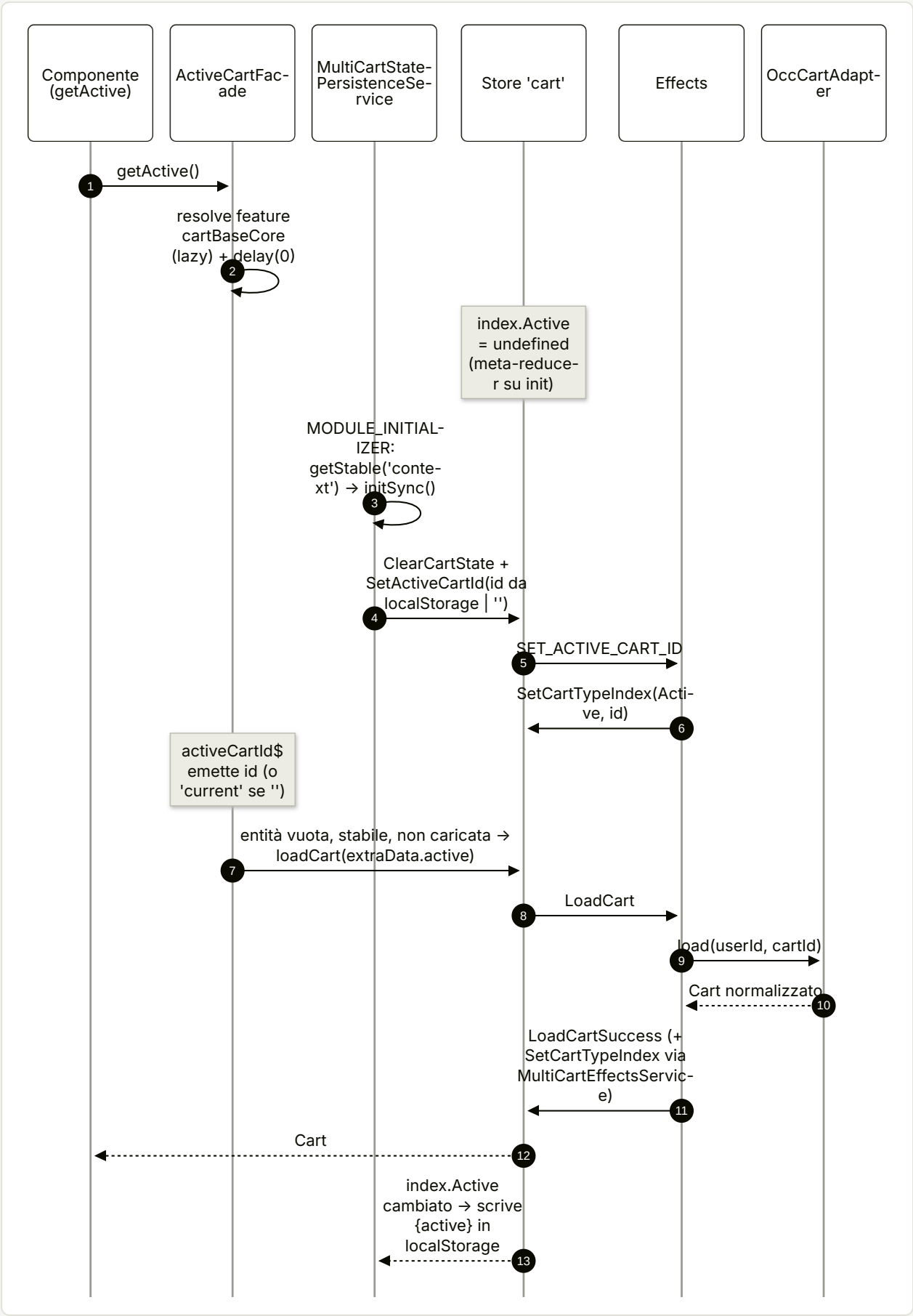


TAVOLA 25.3 **4.3 Add to cart con carrello inesistente**

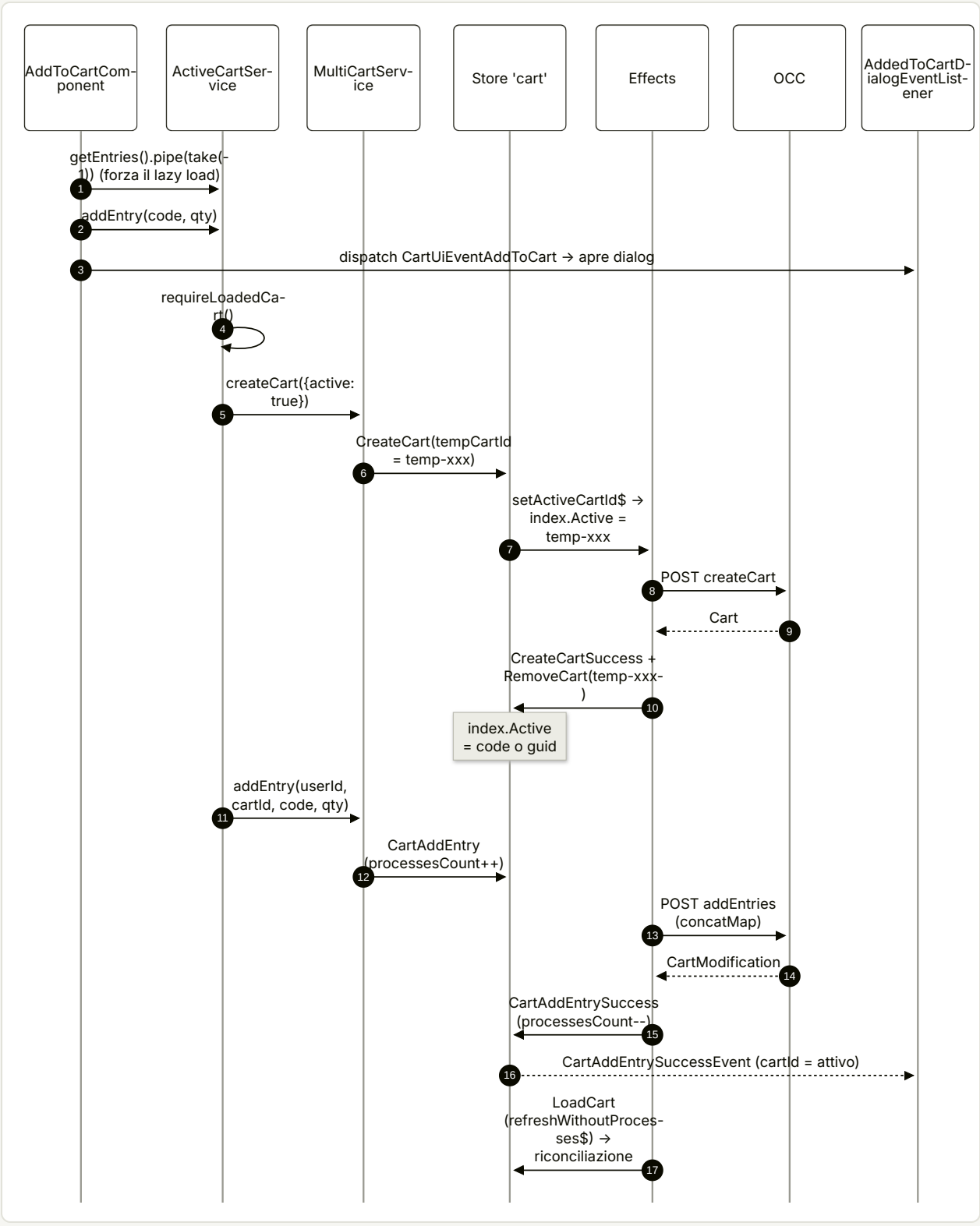
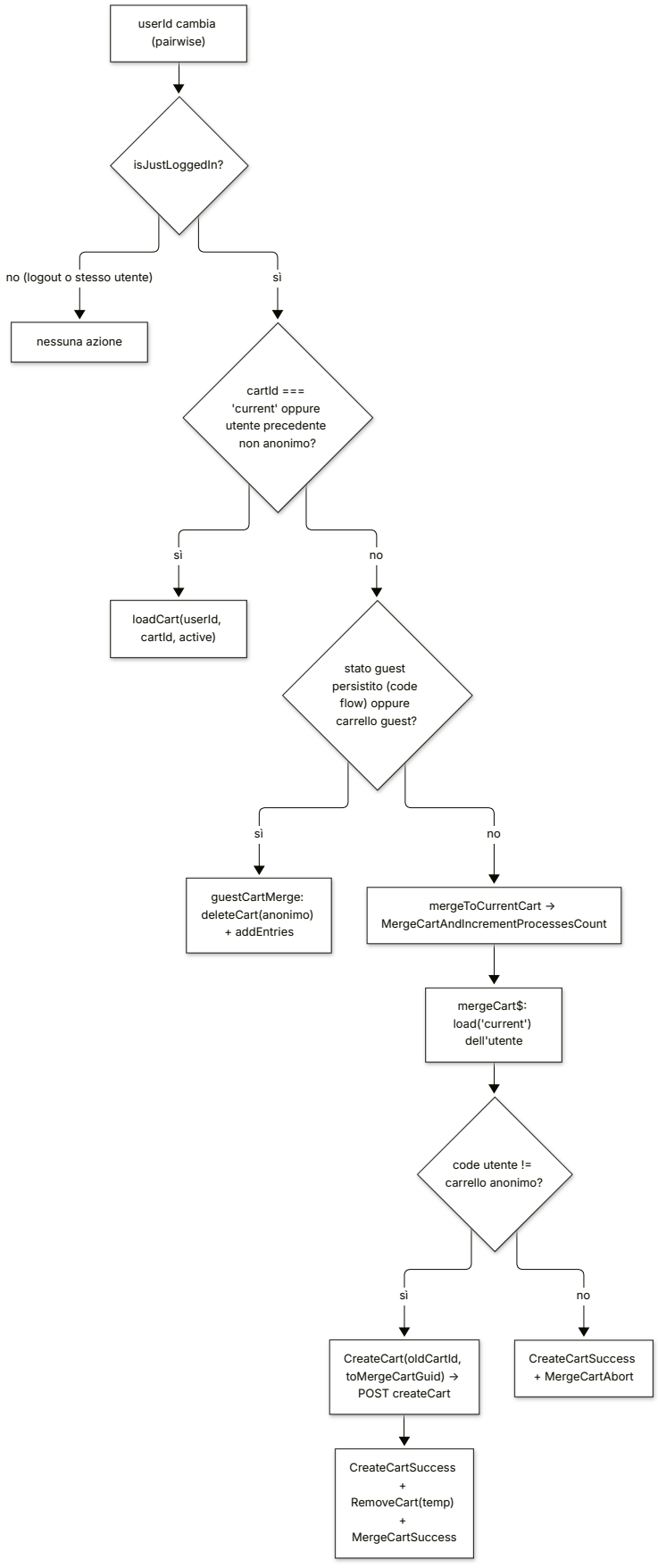


TAVOLA 25.4 **4.4 Login: decisione tra load, guest merge e merge**



26

Checkout

Dal carrello all'ordine: indirizzo, delivery mode, pagamento, revisione. Le tre sotto-feature della libreria.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 12 — Feature module e lazy loading in Spartacus, 20 — Facade, command & query, eventi, 21 — Livello OCC, 14 — Routing. Collegati: 25 — Carrello, 27 — Ordini (il place order vive lì), 16 — CMS rendering (i guard dei componenti CMS), 11 — Configurazione (merge dei config).

1. A cosa serve e perché esiste

Il checkout è il percorso che porta da un carrello attivo a un ordine: scelta dell'indirizzo di consegna, della modalità di consegna, del metodo di pagamento, revisione e conferma. In Spartacus vive nella libreria `@spartacus/checkout`, che ha un `public_api.ts` di radice vuoto: tutto è esposto tramite *secondary entry point* (`feature-libs/checkout/public_api.ts:L11-L11`). La libreria è divisa in tre sotto-feature:

SOTTO-FEATURE	MODULO "PESANTE" (LAZY)	CONTENUTO
`base`	`CheckoutModule` = componenti + core + OCC (`feature-libs/checkout/base/checkout.module.ts:L12-L15`)	checkout B2C: step, guard, facade, adapter OCC, eventi
`b2b`	`CheckoutB2BModule` = componenti + core + OCC B2B (`feature-libs/checkout/b2b/checkout-b2b.module.ts:L12-L19`)	payment type (account/card), cost center, PO number
`scheduled-replenishment`	`CheckoutScheduledReplenishmentModule`, solo componenti (`feature-libs/checkout/scheduled-replenishment/checkout-scheduled-replenishment.module.ts:L10-L13`)	ordini ricorrenti (schedule + place order esteso)

Ogni sotto-feature ha a sua volta i livelli `root` (eager: config, facade astratte, eventi, listener), `core` (servizi command/query, connector), `occ` (adapter) e `components` (componenti CMS, guard, servizi di UI). È lo schema delle feature lib descritto nel capitolo 12.

Tre scelte progettuali dominano tutto il capitolo:

1. **Nessun NgRx per lo stato del checkout.** Il checkout usa Command & Query (capitolo 20): una sola query, `checkoutQuery$`, carica dal backend lo stato del carrello rilevante (indirizzo, delivery mode, payment info) (`feature-libs/checkout/base/core/facade/checkout-query.service.ts:L42-L54`); ogni scrittura è un `Command` che, al successo, dispatcha un evento di dominio (`feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L34-L64`). Dei listener eager trasformano quegli eventi in reset/reload della query (`feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L109-L122`). Restano solo ponti verso lo store legacy dell'utente, marcati `TODO:#deprecation-checkout` (`feature-libs/checkout/base/root/events/checkout-legacy-store-event.listener.ts:L17-L70`).
2. **Gli step sono configurazione.** La sequenza degli step è un array `checkout.steps` di `CheckoutStep` (id, chiave i18n, rotta semantica, tipo) (`feature-libs/checkout/base/root/config/default-checkout-config.ts:L10-L42`); `CheckoutStepService` ne ricava navigazione avanti/indietro e step attivo (`feature-libs/checkout/base/components/services/checkout-step.service.ts:L34-L100`). Le pagine CMS degli step sono pagine di contenuto normali: i "vincoli" del flusso stanno nei **guard dei componenti CMS** (`feature-libs/checkout/base/components/checkout-orchestrator/checkout-orchestrator.module.ts:L18-L25`).
3. **Il place order non appartiene al checkout.** Il componente di conferma chiama `OrderFacade.placeOrder(termsChecked)` di `@spartacus/order/root` (`feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L35-L35`, `feature-libs/order/root/facade/order.`

facade.ts:L47-L47); il checkout reagisce poi all'OrderPlacedEvent (feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L24-L41).

Infine core-libs/core/src/checkout è quasi vuoto: contiene solo i token converter condivisi POINT_OF_SERVICE_NORMALIZER e PAYMENT_DETAILS_NORMALIZER, riesportati da @spartacus/core (core-libs/core/src/checkout/connectors/checkout/converters.ts:L11-L13, core-libs/core/src/checkout/connectors/payment/converters.ts:L11-L13, core-libs/core/public_api.ts:L13-L13).

2. Mappa dei file

PATH	RUOLO
feature-libs/checkout/public_api.ts	Barrel di radice vuoto (:L11-L11).
feature-libs/checkout/base/checkout.module.ts	CheckoutModule: components + core + OCC (:L12-L15).
feature-libs/checkout/base/root/feature-name.ts	CHECKOUT_FEATURE = 'checkout', CHECKOUT_CORE_FEATURE = 'checkoutCore' (:L7-L8).
feature-libs/checkout/base/root/checkout-root.module.ts	CHECKOUT_BASE_CMS_COMPONENTS, defaultCheckoutComponentsConfig(), CheckoutRootModule (:L20-L60).
feature-libs/checkout/base/root/config/checkout-config.ts	CheckoutConfig astratta, DeliveryModePreferences, augmentation di Config (:L12-L55).
feature-libs/checkout/base/root/config/default-checkout-config.ts	4 step base, express, defaultDeliveryMode, guest (:L10-L42).
feature-libs/checkout/base/root/config/default-checkout-routing-config.ts	Rotte semantiche checkout* (:L9-L20).
feature-libs/checkout/base/root/model/checkout-step.model.ts	CheckoutStep, CheckoutStepType, CheckoutStepState (:L7-L27).
feature-libs/checkout/base/root/model/checkout-flow.model.ts	CheckoutFlow (:L10-L31).
feature-libs/checkout/base/root/model/checkout-state.model.ts	CheckoutState (:L10-L14).
feature-libs/checkout/base/root/facade/*.facade.ts	5 facade astratte lazy (facadeFactory).
feature-libs/checkout/base/root/events/checkout.events.ts	Gerarchia Checkout*Event (:L12-L203).
feature-libs/checkout/base/root/events/checkout-event.module.ts	CheckoutEventModule: istanzia eager 6 listener (:L15-L27).
feature-libs/checkout/base/root/events/*-event.listener.ts	Listener: query, delivery address, delivery mode, payment, place order, legacy store.
feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts	CheckoutCartInterceptor ("Cart not found" durante il checkout) (:L26-L95).
feature-libs/checkout/base/core/checkout-core.module.ts	Facade providers, connector, CheckoutPageMetaResolver (:L17-L33).

PATH	RUOLO
feature-libs/checkout/base/core/facade/facade-providers.ts	Facade astratta → service concreto via useExisting (:L21-L47).
feature-libs/checkout/base/core/facade/checkout-query.service.ts	La query centrale checkoutQuery$ (:L28-L89).
feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts	Command create/set/clear address (:L30-L165).
feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts	Query delivery modes supportate + command set/clear.
feature-libs/checkout/base/core/facade/checkout-payment.service.ts	Query card types + command create/set/delete payment.
feature-libs/checkout/base/core/facade/checkout-billing-address.service.ts	Command set billing address (:L28-L53).
feature-libs/checkout/base/core/connectors/**	5 adapter astratti + connector pass-through + token converter.
feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts	Meta resolver del template MultiStepCheckoutSummaryPageTemplate (:L40-L76).
feature-libs/checkout/base/occ/checkout-occ.module.ts	Binding adapter astratti → Occ* (:L24-L50).
feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts	Endpoint OCC (:L9-L35).
feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts	Silent Order Post (SOP) (:L46-L95).
feature-libs/checkout/base/components/services/checkout-step.service.ts	CheckoutStepService (:L19-L204).
feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts	Scelta del flow per payment provider (:L13-L45).
feature-libs/checkout/base/components/services/checkout-config.service.ts	Lettura express/guest/delivery-mode preferito (:L15-L104).
feature-libs/checkout/base/components/services/express-checkout.service.ts	Catena express checkout (:L27-L176).
feature-libs/checkout/base/components/guards/	CheckoutAuthGuard, CartNotEmptyGuard, NotCheckoutAuthGuard, CheckoutGuard, CheckoutStepsSetGuard.
feature-libs/checkout/base/components/checkout-*/	Componenti CMS standalone: orchestrator, delivery address/mode, payment method + form, billing address, review, place order, progress, login, order summary.
feature-libs/checkout/b2b/root/	CheckoutB2BRootModule, augmented-types, step B2B, eventi e facade cost center / payment type.
feature-libs/checkout/b2b/core/	Service cost center / payment type, BadCostCenterRequestHandler.
feature-libs/checkout/b2b/occ/	Adapter OCC cost center / payment type, endpoint B2B.
feature-libs/checkout/b2b/components/	Override guard, componenti payment type / cost center, delivery address e review B2B.

PATH	RUOLO
feature-libs/checkout/scheduled-replenishment/	Root module (CMS list + listener), CheckoutReplenishmentFormService, place order esteso.
core-libs/core/src/checkout/**	Solo POINT_OF_SERVICE_NORMALIZER, PAYMENT_DETAILS_NORMALIZER.
projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts	Integrazione nell'app demo (:L41-L103).
projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts	Chunk lazy che aggrega base + estensioni (:L18-L33).

File aggiuntivi

PATH	RUOLO
feature-libs/checkout/_index.scss	entry SCSS della lib: importa @spartacus/styles/scss/core e gli stili di base, b2b, scheduled-replenishment

3. Come funziona passo per passo

3.1 Bootstrap: root module eager, tutto il resto lazy

1. L'app importa CheckoutRootModule in modo eager; nell'app demo, se environment.b2b, anche CheckoutB2BRootModule e CheckoutScheduledReplenishmentRootModule (projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L41-L87). La demo abilita anche il guest checkout con provideConfig({ checkout: { guest: true } }) (projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L99-L103).
2. CheckoutRootModule importa CheckoutEventModule e fornisce l'interceptor, la config di routing, la config del checkout e, tramite factory, la config featureModules (feature-libs/checkout/base/root/checkout-root.module.ts:L51-L60).
3. CheckoutEventModule è un NgModule vuoto il cui costruttore inietta i 6 listener: iniettarli basta per istanziarli (sono providedIn: 'root') e quindi per far partire le loro subscription già al bootstrap (feature-libs/checkout/base/root/events/checkout-event.module.ts:L15-L27).
4. defaultCheckoutComponentsConfig() dichiara la feature checkout con 14 tipi CMS (CHECKOUT_BASE_CMS_COMPONENTS) e la dipendenza da CART_BASE_FEATURE; checkoutCore è un *alias* di checkout ("by default core is bundled together with components") (feature-libs/checkout/base/root/checkout-root.module.ts:L20-L49).
5. Quando una pagina CMS contiene uno di quei tipi, il caricatore delle feature scarica il chunk configurato: nell'app demo CheckoutWrapperModule, che importa CheckoutModule più, condizionalmente, B2B, replenishment, digital payments e S/4 service (projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L18-L33).
6. Le facade del checkout sono classi astratte fornite in root via facadeFactory con feature: CHECKOUT_CORE_FEATURE (feature-libs/checkout/base/root/facade/checkout-delivery-address.facade.ts:L12-L28). Se qualcuno (es. un listener) chiama una facade prima che la feature sia caricata, il proxy risolve checkoutCore → alias checkout e scarica lo stesso chunk (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57).

7. Quattro facade base usano `async: true`, che aggiunge `delay(0)` al resolver ("Required to make lazy NgRx store feature ready", marcato `TODO:#deprecation-checkout`) (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L53-L56, core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L36); `CheckoutQueryFacade` e le facade B2B no (feature-libs/checkout/base/root/facade/checkout-query.facade.ts:L13-L21, feature-libs/checkout/b2b/root/facade/checkout-cost-center.facade.ts:L15-L20).
8. Nel chunk lazy, `CheckoutCoreModule` lega ogni facade astratta al service concreto con `useExisting` (feature-libs/checkout/base/core/checkout-core.module.ts:L17-L33, feature-libs/checkout/base/core/facade/facade-providers.ts:L21-L47) e `CheckoutOccModule` lega gli adapter astratti alle implementazioni `Occ*` (feature-libs/checkout/base/occ/checkout-occ.module.ts:L24-L50).

Una conseguenza importante: le chiamate a metodi di una facade proxy sono **eager**. `call()` costruisce un `connectable` con `ReplaySubject` e lo connette subito, quindi il metodo reale viene invocato anche se nessuno si sottoscrive al risultato (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L92). Lo stesso vale per `Command.execute` (core-libs/core/src/util/command-query/command.service.ts:L126-L131). Diversi punti del checkout contano su questo (vedi §9).

3.2 Lo stato: una query, molte proiezioni

1. `CheckoutQueryService.checkoutQuery$` è una `Query<CheckoutState | undefined>` creata con `QueryService.create`: il loader esegue `checkoutPreconditions()` e poi `CheckoutConnector.getCheckoutDetails(userId, cartId)`; si ricarica su `CheckoutQueryReloadEvent` e si resetta su `CheckoutQueryResetEvent` (feature-libs/checkout/base/core/facade/checkout-query.service.ts:L32-L54).
2. `checkoutPreconditions()` legge (`take(1)`) userId, id del carrello attivo e `isGuestCart`, e lancia `Error('Checkout conditions not met')` se manca un dato o se l'utente è anonimo senza guest cart (feature-libs/checkout/base/core/facade/checkout-query.service.ts:L66-L84). Lo stesso metodo è duplicato identico in ogni service, base e B2B (feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L133-L151, feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L149-L167, feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L138-L156, feature-libs/checkout/base/core/facade/checkout-billing-address.service.ts:L58-L76, feature-libs/checkout/b2b/core/facade/checkout-cost-center.service.ts:L65-L83, feature-libs/checkout/b2b/core/facade/checkout-payment-type.service.ts:L98-L116).
3. L'adapter OCC esegue un GET su `users/${userId}/carts/${cartId}?fields=deliveryAddress(FULL),deliveryMode(FULL),paymentInfo(FULL)` (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L29-L30); in B2B l'endpoint include anche `costCenter(FULL),purchaseOrderNumber,paymentType(FULL)` (feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L11-L29). Il pattern comune degli adapter è `catchError → tryNormalizeHttpError, backOff({ shouldRetry: isJaloError })`, converter (feature-libs/checkout/base/occ/adapters/occ-checkout.adapter.ts:L35-L50).
4. Lo stato della query parte da `{data: undefined, error: false, loading: true}`; in errore diventa `{loading: false, error, data: undefined}` (core-libs/core/src/util/command-query/query.service.ts:L61-L102). Su `reloadOn` il dato è preservato (solo `loading: true`), su `resetOn` torna a `initialState` (core-libs/core/src/util/command-query/query.service.ts:L105-L128). La query carica solo finché ha subscriber (core-libs/core/src/util/command-query/query.service.ts:L130-L133).
5. Le "selezioni" correnti non hanno query proprie: `getDeliveryAddressState()`, `getSelectedDeliveryModeState()` e `getPaymentDetailsState()` sono proiezioni `map` di `getCheckoutDetailsState()` (feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L153-L160, feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L180-L186, feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L168-L172). Ana-

logamente in B2B `getCostCenterState()` proietta `state.data?.costCenter` (feature-libs/checkout/b2b/core/facade/checkout-cost-center.service.ts:L85-L92).

6. Due query separate esistono solo per liste che non sono "selezioni": le delivery mode supportate (feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L41-L66) e i card type (senza precondizioni, perché non dipendono dal carrello) (feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L52-L57); in B2B i payment type.

3.3 Le scritture: command + evento + listener

Ogni command segue la stessa forma: precondizioni → chiamata al connector → `tap` che dispatcha un evento. Esempi:

- `createDeliveryAddressCommand` chiama `createAddress`, ricopia nella risposta `titleCode` e `region.isocodeShort` del payload, dispatcha `CheckoutDeliveryAddressCreatedEvent`; strategia `CancelPrevious` (feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L34-L64).
- `setDeliveryAddressCommand` richiede `address.id` (altrimenti "Checkout conditions not met"), chiama `setAddress` (PUT con `addressId` in query) e dispatcha `CheckoutDeliveryAddressSetEvent` (feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L66-L94, feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-address.adapter.ts:L75-L133).
- `setDeliveryModeCommand` dispatcha `CheckoutDeliveryModeSetEvent` con `cartCode: cartId` (feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L68-L88); `clearDeliveryModeCommand` dispatcha `CheckoutDeliveryModeClearedEvent` o `CheckoutDeliveryModeClearedErrorEvent` (feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L90-L134).
- `createPaymentMethodCommand` dispatcha `CheckoutPaymentDetailsCreatedEvent`; `setPaymentMethodCommand` richiede `paymentDetails.id` e dispatcha `CheckoutPaymentDetailsSetEvent` (feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L59-L110).
- Eccezioni: `deletePaymentMethodCommand` non ha opzioni (strategia default `Queue`) e dispatcha direttamente `CheckoutQueryResetEvent` (feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L112-L123, core-libs/core/src/util/command-query/command.service.ts:L107-L117); `CheckoutBillingAddressService` non dispatcha alcun evento (feature-libs/checkout/base/core/facade/checkout-billing-address.service.ts:L28-L53).

I listener (root, eager) chiudono il ciclo:

EVENTO IN INGRESSO	REAZIONE	RIFERIMENTO
`LanguageSetEvent`, `CurrencySetEvent`	`CheckoutQueryReloadEvent`	feature-libs/checkout/base/root/events/checkout-query-event.listener.ts:L38-L47
`LogoutEvent`, `LoginEvent`, `SaveCartSuccessEvent`, `RestoreSavedCartSuccessEvent`, `MergeCartSuccessEvent`, `OrderPlacedEvent`	`CheckoutQueryResetEvent`	feature-libs/checkout/base/root/events/checkout-query-event.listener.ts:L49-L62

EVENTO IN INGRESSO	REAZIONE	RIFERIMENTO
`UpdateUserAddressEvent` / `DeleteUserAddressEvent`	`clearCheckoutDeliveryAddress()` (lazy-load del checkout) + reset delivery modes	`feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L57-L83`
`CheckoutDeliveryAddressCreatedEvent`	`LoadUserAddressesEvent` (non anonimi), messaggio globale, reset delivery modes, reset query	`feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L85-L107`
`CheckoutDeliveryAddressSetEvent`	reset delivery modes + reset query	`feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L109-L122`
`CheckoutDeliveryAddressClearedEvent`, `DeleteCartEvent`	reset query	`feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L124-L142`
`CheckoutDeliveryModeSetEvent`, `CheckoutDeliveryModeClearedEvent`, `CheckoutDeliveryModeClearedErrorEvent`	reset query + `LoadCartEvent` (totali carrello)	`feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L45-L100`
`CheckoutSupportedDeliveryModesQueryResetEvent`	`LoadCartEvent`	`feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L107-L126`
`LanguageSetEvent`, `CurrencySetEvent`	`CheckoutSupportedDeliveryModesQueryReloadEvent`	`feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L128-L140`

EVENTO IN INGRESSO	REAZIONE	RIFERIMENTO
`LogoutEvent`, `LoginEvent`	`CheckoutSupportedDeliveryModesQueryResetEvent`	feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L142-L154
`CheckoutPaymentDetailsCreatedEvent`	`LoadUserPaymentMethodsEvent` (non anonimi), messaggio, reset query	feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L44-L60
`CheckoutPaymentDetailsSetEvent`	reset query	feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L62-L68
`LanguageSetEvent`, `CurrencySetEvent`	`CheckoutPaymentCardTypesQueryReloadEvent`	feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L70-L82
`OrderPlacedEvent`	`RemoveCartEvent` + reset query	feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L24-L41
`LoadUserAddressesEvent` / `LoadUserPaymentMethodsEvent`	azioni NgRx legacy `UserActions.*`	feature-libs/checkout/base/root/events/checkout-legacy-store-event.listener.ts:L17-L70

La gerarchia degli eventi (`CheckoutEvent` astratto con `userId`, `cartId`, `cartCode`; sottofamiglie delivery address / delivery mode / payment; `CheckoutQueryReloadEvent` e `CheckoutQueryResetEvent` che estendono direttamente `CxEvent`) è in feature-libs/checkout/base/root/events/checkout.events.ts:L12-L203.

3.4 Gli step: configurazione, flow e `CheckoutStepService`

1. `CheckoutConfig` è una classe astratta `providedIn: 'root', useExisting: Config` con chiave `checkout` (`steps`, `express`, `defaultDeliveryMode`, `guest`, `guestUseSavedAddress`, `flows`), e augmenta `Config` di `@spartacus/core` (feature-libs/checkout/base/root/config/checkout-config.ts:L18-L55).
2. I default base sono 4 step — `deliveryAddress`, `deliveryMode`, `paymentDetails`, `reviewOrder` — con `express: false`, `defaultDeliveryMode: [FREE]`, `guest: false` (feature-libs/checkout/base/root/config/

default-checkout-config.ts:L10-L42). Ogni step punta a una rotta semantica; le rotte default sono checkoutLogin (checkout-login, authFlow: true), checkout, checkoutDeliveryAddress, checkoutDeliveryMode, checkoutPaymentDetails, checkoutReviewOrder (feature-libs/checkout/base/root/config/default-checkout-routing-config.ts:L9-L20).

3. CheckoutFlowOrchestratorService (root) legge baseSite.baseStore.paymentProvider una volta (subscribe nel costruttore) e getCheckoutFlow() restituisce checkout.flows[provider] se esiste, altrimenti l'intero oggetto checkout (feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L20-L44). CheckoutFlow ripete le stesse chiavi di checkout tranne flows (feature-libs/checkout/base/root/model/checkout-flow.model.ts:L10-L31).
4. CheckoutStepService (root) nel costruttore chiama resetSteps(): prende gli step dal flow dell'orchestratore, scarta quelli disabled, li clona con Object.assign ed emette su steps$ (feature-libs/checkout/base/components/services/checkout-step.service.ts:L57-L63, feature-libs/checkout/base/components/services/checkout-step.service.ts:L88-L100).
5. activeStepIndex$ confronta router.state.context.id con /${paths[0]} di ogni step (feature-libs/checkout/base/components/services/checkout-step.service.ts:L34-L55). next()/back() usano invece l'ActivatedRoute del componente; back() dal primo step torna a 'cart' (feature-libs/checkout/base/components/services/checkout-step.service.ts:L65-L73), e il testo del bottone è checkout.backToCart o common.back (feature-libs/checkout/base/components/services/checkout-step.service.ts:L81-L86).
6. disableEnableStep(type, disabled) muta lo step in allSteps e riemette su steps$ solo quelli abilitati (feature-libs/checkout/base/components/services/checkout-step.service.ts:L102-L113); getNextCheckoutStepUrl/getPreviousCheckoutStepUrl saltano gli step disabilitati (feature-libs/checkout/base/components/services/checkout-step.service.ts:L130-L166).
7. CheckoutConfigService legge express, guest e defaultDeliveryMode dal flow (calcolato una volta alla costruzione) (feature-libs/checkout/base/components/services/checkout-config.service.ts:L19-L37). getPreferredDeliveryMode() ordina in place per costo e applica le preferenze in cascata con findMatchingDeliveryMode ricorsivo, con fallback deliveryModes[0].code (feature-libs/checkout/base/components/services/checkout-config.service.ts:L58-L104).

3.5 Navigazione a /checkout: i guard

I guard non sono sulle rotte Angular: sono dichiarati nei cmsComponents e applicati da CmsGuardsService, che li istanzia con getLastValueSync(unifiedInjector.get(guardClass)), cioè dall'ultimo injector (anche lazy) che fornisce quel token (core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48, core-libs/core/src/lazy-loading/unified-injector.ts:L52-L67). Il mapping è (feature-libs/checkout/base/components/checkout-orchestrator/checkout-orchestrator.module.ts:L18-L25, feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.module.ts:L35-L42, feature-libs/checkout/base/components/checkout-login/checkout-login.module.ts:L37-L44, feature-libs/checkout/base/components/checkout-order-summary/checkout-order-summary.module.ts:L16-L22, feature-libs/checkout/base/components/checkout-progress/checkout-progress.module.ts:L33-L40):

COMPONENTE CMS	GUARD
CheckoutOrchestrator	CheckoutAuthGuard, CartNotEmptyGuard, CheckoutGuard
CheckoutDeliveryAddress	CheckoutAuthGuard, CartNotEmptyGuard, CartValidationGuard
CheckoutDeliveryMode, CheckoutPaymentDetails, CheckoutPlaceOrder, CheckoutReviewOrder, review overview/payment/shipping	CheckoutAuthGuard, CartNotEmptyGuard
CheckoutProgress (+ mobile top/bottom)	CheckoutAuthGuard, CartNotEmptyGuard, CheckoutStepsSetGuard

COMPONENTE CMS	GUARD
GuestCheckoutLoginComponent	NotCheckoutAuthGuard
CheckoutOrderSummary	nessuno

1. CheckoutAuthGuard attende activeCartFacade.isStable(); utente loggato → true; anonimo con guest cart → true; altrimenti handleAnonymousUser() (feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L38-L57). Questo salva l'URL corrente e, con il toggle authorizationCodeFlowByDefault, scrive IS_GUEST_USER_CHECKOUT_KEY in localStorage (se guest abilitato) e redirige a login; senza toggle, se il guest è abilitato redirige a login?forced=true (feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L59-L77).
2. CartNotEmptyGuard redirige a home se totalItems è falsy (feature-libs/checkout/base/components/guards/cart-not-empty.guard.ts:L24-L39).
3. CheckoutGuard (solo sull'orchestrator) redirige alla prima rotta degli step; se express è attivo e il carrello non è guest, prova ExpressCheckoutService.trySetDefaultCheckoutDetails() e, se riesce, redirige al review order (feature-libs/checkout/base/components/guards/checkout.guard.ts:L22-L30, feature-libs/checkout/base/components/guards/checkout.guard.ts:L41-L70). Il componente CheckoutOrchestratorComponent ha template vuoto: esiste solo per agganciare questi guard alla pagina /checkout (feature-libs/checkout/base/components/checkout-orchestrator/checkout-orchestrator.component.ts:L9-L18).
4. ExpressCheckoutService costruisce nel costruttore una catena: indirizzo di default (o primo) → delivery mode preferita → pagamento di default (o primo); ogni anello invoca la facade e poi verifica lo stato dalla query (feature-libs/checkout/base/components/services/express-checkout.service.ts:L35-L46, feature-libs/checkout/base/components/services/express-checkout.service.ts:L48-L169).
5. NotCheckoutAuthGuard (pagina checkout-login) redirige a home se loggato e al carrello se c'è già un guest cart (feature-libs/checkout/base/components/guards/not-checkout-auth.guard.ts:L29-L44).

3.6 Ingresso in uno step: CheckoutStepsSetGuard

1. Nel costruttore (servizio root) sottoscrive hasDeliveryItems(): se il carrello non ha articoli da spedire disabilita DELIVERY_ADDRESS e DELIVERY_MODE e regola nameMultiLine di PAYMENT_DETAILS/REVIEW_ORDER (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L49-L70, feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L174-L182).
2. canActivate(route) cerca in steps$ (take(1)) lo step la cui rotta coincide con l'URL; se non lo trova, in dev mode logga Missing step with route ... e redirige a checkout (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L73-L104).
3. Se lo trova, verifica **solo lo step immediatamente precedente** con isStepSet(steps[currentIndex - 1]) (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L92-L93).
4. isStepSet dispatcha per step.type[0]; per PAYMENT_DETAILS, se lo step DELIVERY_MODE è disabilitato, chiama setDeliveryMode('pickup') prima di verificare il pagamento (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132).
5. Le verifiche filtrano !state.loading e, se il dato manca, restituiscono un UrlTree verso la rotta dello step mancante (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L134-L172).

3.7 Gli step nei componenti

- **Indirizzo.** ngOnInit carica gli indirizzi utente (se non guest) (feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L114-L119); selectDefaultAddress

auto-seleziona una sola volta il `defaultAddress` (feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L313-L330). `setAddress = setDeliveryAddress` seguito (switchMap) da `clearCheckoutDeliveryMode()` (feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L357-L374); `addAddress = createAndSetAddress` → clear delivery mode → `next()` (feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L213-L246).

- **Delivery mode.** `supportedDeliveryModes$` in un `tap` imposta sempre una modalità (quella selezionata se ancora nella lista, altrimenti la preferita) chiamando `changeMode`, e nasconde dalla lista il codice `pickup` (feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.ts:L79-L100). Il componente ha `composition.inner: ['PickupInStoreDeliveryModeComponent']` (feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.module.ts:L37-L49).
- **Pagamento.** `CheckoutPaymentMethodComponent.ngOnInit` carica i metodi di pagamento utente (se non guest) e memorizza l'indirizzo di consegna come fallback del billing (feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-method.component.ts:L131-L147); `setPaymentDetails` usa `billingAddress ?? this.deliveryAddress`, chiama `createPaymentDetails` e al complete `next()` (feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-method.component.ts:L278-L301). Il form carta emette `billingAddress: null` se "same as delivery" (feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-form/checkout-payment-form.component.ts:L273-L312).
- **SOP (Silent Order Post).** `OccCheckoutPaymentAdapter.createPaymentDetails` fa GET `payment/sop/request`, POST form-urlencoded al `postUrl` del provider, parsing dell'HTML di risposta (primo `<form>`, i suoi `input`), POST `payment/sop/response`, normalizzazione con `PAYMENT_DETAILS_NORMALIZER`; forza `savePaymentInfo = true` (feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L46-L95, feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L312-L330). I parametri sono rimappati con le `mappingLabels` (`hybris_card_number`, `hybris_billTo_*`) (feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L265-L310).
- **Review.** `CheckoutReviewSubmitComponent` raggruppa gli step in "delivery" e "payment" tramite metodi protetti sovrascrivibili (feature-libs/checkout/base/components/checkout-review-submit/checkout-review-submit.component.ts:L101-L110).
- **Place order.** Form `termsAndConditions: [false, Validators.requiredTrue]` (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L76-L78). `submitForm()` con il toggle `enableCartSlowNetworkResilience` piazza l'ordine solo se `isStable()` è true (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L124-L142); `launchPlaceOrder()` apre lo spinner `LAUNCH_CALLER.PLACE_ORDER_SPINNER`, chiama `orderFacade.placeOrder(valid)`, al successo naviga a `orderConfirmation`, in errore chiude lo spinner (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L144-L170). Lato order, `placeOrderCommand` dispatcha `OrderPlacedEvent` con `cartCode: cartId` (feature-libs/order/core/facade/order.service.ts:L26-L54).
- **Guest login.** `CheckoutLoginComponent.onSubmit` chiama `activeCartFacade.addEmail(email)` e, quando il carrello diventa guest, `authRedirectService.redirect()` (feature-libs/checkout/base/components/checkout-login/checkout-login.component.ts:L58-L73).
- **Progress.** Stato step: `DISABLED` se indice > attivo, `SELECTED` se attivo, altrimenti `COMPLETED` (feature-libs/checkout/base/components/checkout-progress/checkout-progress.component.ts:L67-L87).

3.8 Interceptor "Cart not found"

`CheckoutCartInterceptor` legge una sola volta lo stato router e osserva solo gli errori della richiesta; se la rotta semantica inizia (case-insensitive) con `checkout` e l'errore è 400 con `errors[0].type === 'CartError'` e `reason === 'notFound'`, naviga a `cart` e ricarica il carrello indicato in `errors[0].subject` (feature-libs/

checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L37-L94). È l'unico interceptor del checkout ed è registrato via useExisting (feature-libs/checkout/base/root/http-interceptors/index.ts:L11-L17).

3.9 B2B

1. CheckoutB2BRootModule importa per primo ./model/augmented-types ("Import augmented types first") e ridefinisce i cmsComponents della **stessa** feature checkout (base + CheckoutCostCenterComponent, CheckoutPaymentType) (feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L8-L52).
2. L'augmentation aggiunge CheckoutStepType.PAYMENT_TYPE = 'paymentType' sia come tipo (declare module) sia a runtime (assegnazione sull'enum), più costCenter, purchaseOrderNumber, paymentType a CheckoutState (feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23). È possibile perché il public_api root esporta CheckoutStepType e CheckoutState nel blocco delimitato dal commento AUGMENTABLE_TYPES_START/END (feature-libs/checkout/base/root/public_api.ts:L14-L17).
3. I default B2B sono 5 step con paymentType (rotta checkout/payment-type) per primo (feature-libs/checkout/b2b/root/config/default-b2b-checkout-config.ts:L13-L51, feature-libs/checkout/b2b/root/config/default-checkout-b2b-routing-config.ts:L9-L15). Poiché deepMerge sostituisce gli array, gli step B2B **rimpiazzano** quelli base, mentre gli oggetti vengono fusi (la dependencies della feature base resta) (core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/config/utils/deep-merge.ts:L31-L46).
4. CheckoutB2BComponentsModule (nel chunk lazy) sostituisce CheckoutAuthGuard → CheckoutB2BAuthGuard e CheckoutStepsSetGuard → CheckoutB2BStepsSetGuard con useExisting (feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37); l'override ha effetto grazie a getLastValueSync sull'UnifiedInjector (§3.5).
5. CheckoutB2BAuthGuard richiede inoltre, per utenti con roles, il ruolo B2BUserRole.CUSTOMER, altrimenti warning checkoutB2B.invalid.accountType e redirect home (feature-libs/checkout/b2b/components/guards/checkout-b2b-auth.guard.ts:L51-L88).
6. CheckoutB2BStepsSetGuard disabilita PAYMENT_DETAILS quando il pagamento è ACCOUNT, poi verifica lo step precedente; per DELIVERY_ADDRESS con ACCOUNT richiede indirizzo + cost center, senza ACCOUNT richiede indirizzo e costCenter === undefined (feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L57-L97, feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L139-L177).
7. Componenti: CheckoutPaymentTypeComponent auto-imposta il primo tipo se il selezionato non è disponibile e abilita/disabilita PAYMENT_DETAILS (feature-libs/checkout/b2b/components/checkout-payment-type/checkout-payment-type.component.ts:L85-L130); CheckoutCostCenterComponent è nascosto se non ACCOUNT e imposta il primo cost center attivo se il carrello non ne ha (feature-libs/checkout/b2b/components/checkout-cost-center/checkout-cost-center.component.ts:L44-L88); B2BCheckoutDeliveryAddressComponent con ACCOUNT mostra gli indirizzi del cost center (feature-libs/checkout/b2b/components/checkout-delivery-address/checkout-delivery-address.component.ts:L74-L93).
8. Set di cost center o payment type → reset delivery modes + reset query (feature-libs/checkout/b2b/root/events/checkout-cost-center-event.listener.ts:L26-L38, feature-libs/checkout/b2b/root/events/checkout-payment-type-event.listener.ts:L39-L51). BadCostCenterRequestHandler (registrato come HttpErrorHandler multi) mostra checkoutB2B.invalidCostCenter su 400 con almeno un errore EntityValidationError (scartati prima i NO_LONGER_VALID) e URL il cui path termina con costcenter **e** che contiene il query param costCenterId (feature-libs/checkout/b2b/core/http-interceptors/bad-request/bad-cost-center-request.handler.ts:L50-L63, feature-libs/checkout/b2b/core/http-interceptors/bad-request/bad-cost-center-request.model.ts:L9-L9, feature-libs/checkout/b2b/core/

checkout-b2b-core.module.ts:L14-L26, feature-libs/checkout/b2b/core/http-interceptors/bad-request/bad-cost-center-request.handler.ts:L22-L71).

3.10 Scheduled replenishment

1. `CheckoutScheduledReplenishmentRootModule` ridefinisce ancora i `cmsComponents` di `checkout` = B2B + `CheckoutScheduleReplenishmentOrder` (feature-libs/checkout/scheduled-replenishment/root/checkout-scheduled-replenishment-root.module.ts:L13-L37).
2. `CheckoutReplenishmentFormService` (root) tiene i dati del form (default: daily, 14 giorni, lunedì, data odierna ISO) e `orderType$` (default `PLACE_ORDER`) (feature-libs/checkout/scheduled-replenishment/components/services/checkout-replenishment-form.service.ts:L39-L52), che torna a `PLACE_ORDER` su `CheckoutDeliveryAddressSetEvent`, `CheckoutDeliveryAddressClearedEvent`, `CheckoutDeliveryModeSetEvent`, `CheckoutDeliveryModeClearedEvent`, `CheckoutPaymentDetailsCreatedEvent`, `CheckoutPaymentDetailsSetEvent`, login/logout e save/restore/merge cart; non su `CheckoutDeliveryAddressCreatedEvent`, e non esiste un evento di clear del pagamento (feature-libs/checkout/scheduled-replenishment/components/services/checkout-replenishment-form.service.ts:L58-L78).
3. `CheckoutScheduledReplenishmentPlaceOrderComponent` estende il componente base e sceglie tra `orderFacade.placeOrder` e `scheduledReplenishmentOrderFacade.scheduleReplenishmentOrder(formData, termsChecked)`, navigando a `orderConfirmation` o `replenishmentConfirmation` (feature-libs/checkout/scheduled-replenishment/components/checkout-place-order/checkout-place-order.component.ts:L105-L170); ridefinisce `CheckoutPlaceOrder` nel CMS config (feature-libs/checkout/scheduled-replenishment/components/checkout-place-order/checkout-place-order.module.ts:L35-L42).
4. `ReplenishmentOrderScheduledEvent` → `RemoveCartEvent` + `CheckoutQueryResetEvent` (feature-libs/checkout/scheduled-replenishment/root/events/checkout-scheduled-replenishment-event.listener.ts:L24-L41).

---

4. Diagrammi

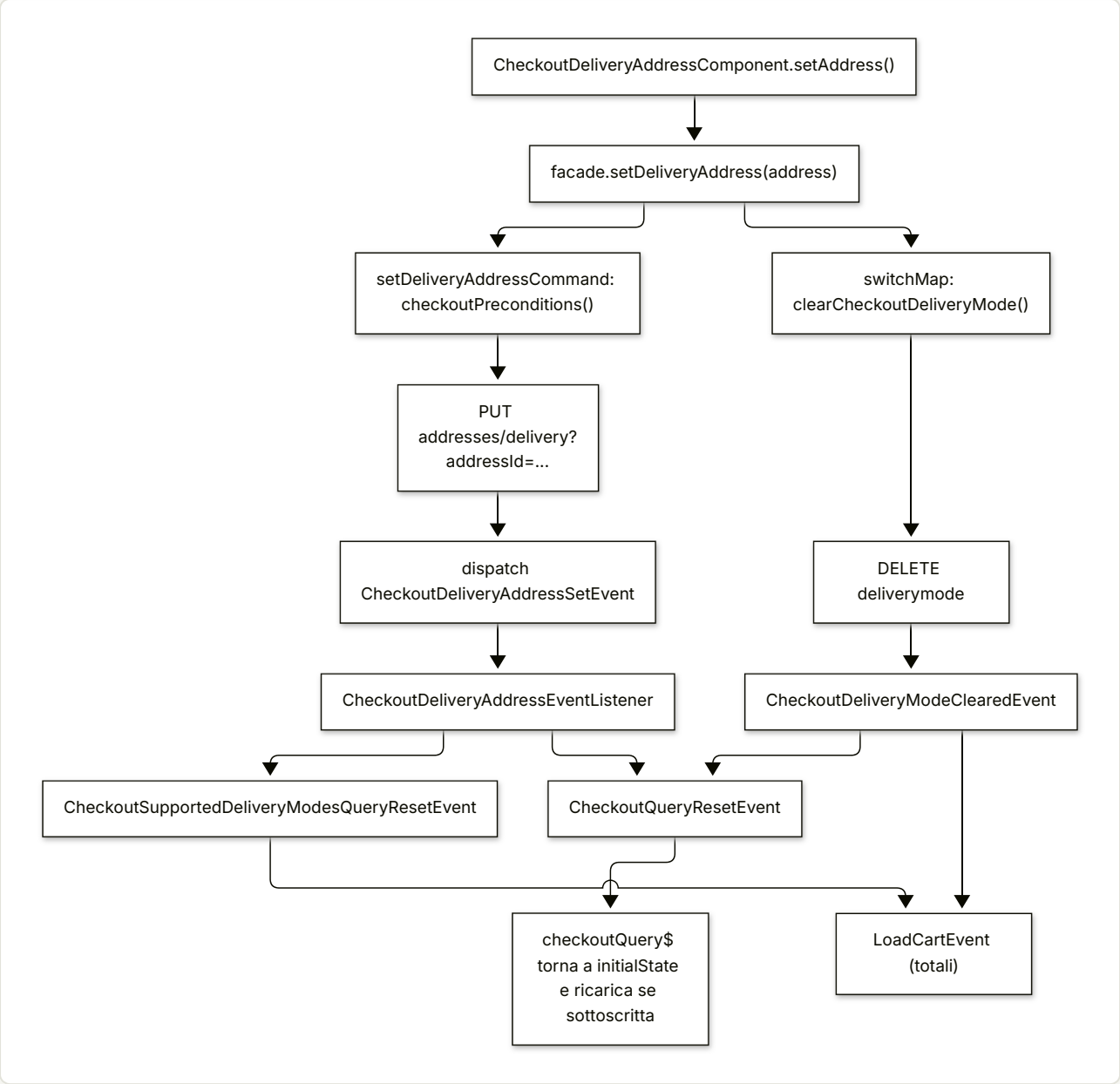
4.1 Bootstrap e lazy loading

Fatti: §3.1 (feature-libs/checkout/base/root/checkout-root.module.ts:L37-L60, feature-libs/checkout/base/root/events/checkout-event.module.ts:L15-L27, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57, feature-libs/checkout/base/core/facade/facade-providers.ts:L21-L47).

■ **TAVOLA 26.1** diagramma a tutta pagina, in fondo al capitolo

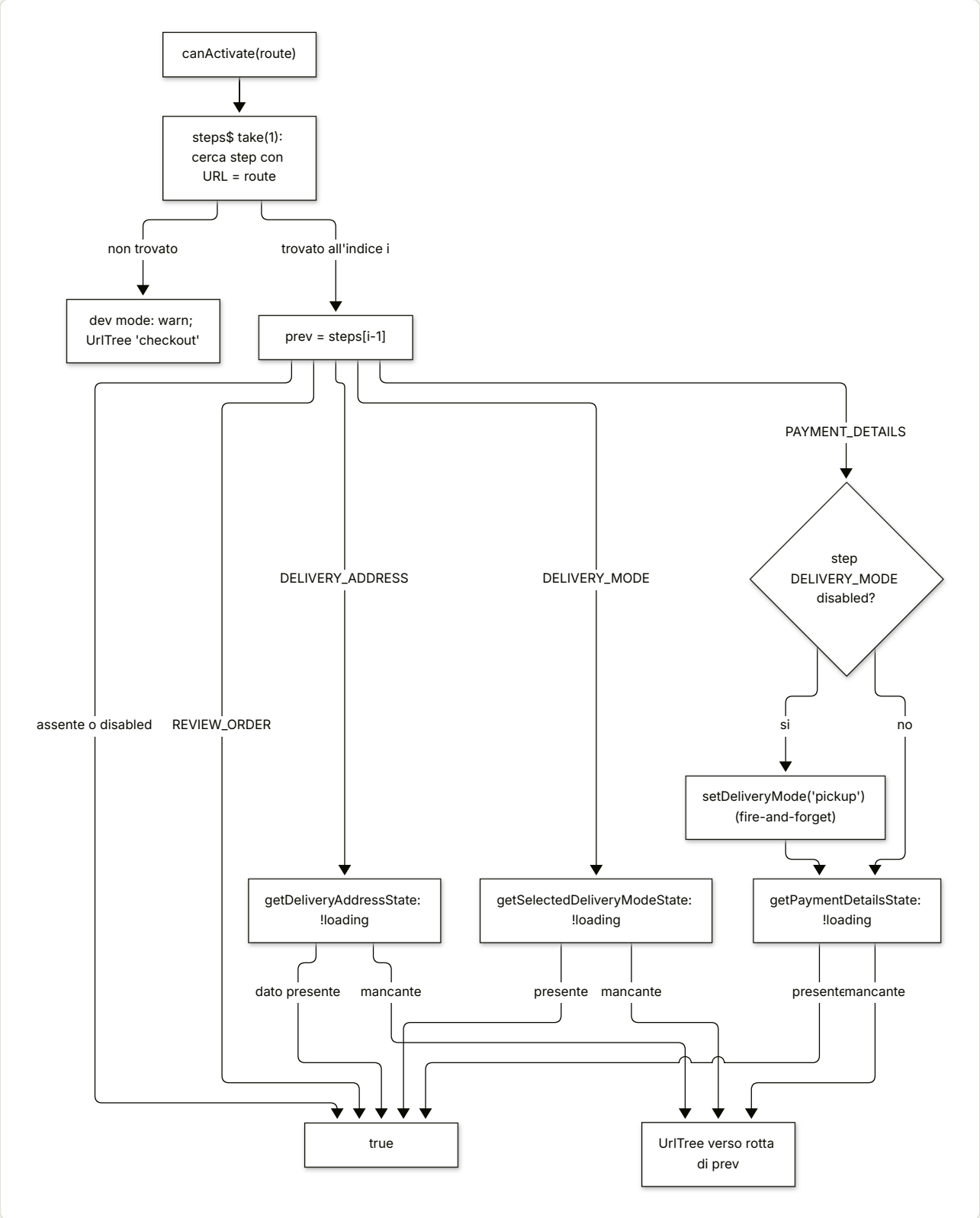
4.2 Ciclo command → evento → query (scelta indirizzo)

Fatti: §3.3, §3.7 (feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L357-L374, feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L66-L94, feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L109-L122, feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L64-L126, core-libs/core/src/util/command-query/query.service.ts:L117-L128).



4.3 Decisione di `CheckoutStepsSetGuard`

Fatti: §3.6 (`feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L73-L172`).



4.4 Silent Order Post (nuova carta)

Fatti: §3.7 (`feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L46-L95`, `feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L44-L60`, `feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-method.component.ts:L278-L301`).

■ **TAVOLA 26.2** diagramma a tutta pagina, in fondo al capitolo

---

5. Estratti di codice reale

5.1 Mapping lazy e root module

feature-libs/checkout/base/root/checkout-root.module.ts:L37-L60

TS

```ts
export function defaultCheckoutComponentsConfig() {
  const config: CmsConfig = {
    featureModules: {
      [CHECKOUT_FEATURE]: {
        cmsComponents: CHECKOUT_BASE_CMS_COMPONENTS,
        dependencies: [CART_BASE_FEATURE],
      },
      // by default core is bundled together with components
      [CHECKOUT_CORE_FEATURE]: CHECKOUT_FEATURE,
    },
  };
  return config;
}

@NgModule({
  imports: [CheckoutEventModule],
  providers: [
    ...interceptors,
    provideDefaultConfig(defaultCheckoutRoutingConfig),
    provideDefaultConfig(defaultCheckoutConfig),
    provideDefaultConfigFactory(defaultCheckoutComponentsConfig),
  ],
})
export class CheckoutRootModule {}
```

5.2 Una facade lazy

feature-libs/checkout/base/root/facade/checkout-delivery-address.facade.ts:L12-L28

TS

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: CheckoutDeliveryAddressFacade,
      feature: CHECKOUT_CORE_FEATURE,
      methods: [
        'getDeliveryAddressState',
```

```
        'createAndSetAddress',
        'setDeliveryAddress',
        'clearCheckoutDeliveryAddress',
      ],
      // TODO:#deprecation-checkout - remove once we remove ngrx
      async: true,
    }),
})
export abstract class CheckoutDeliveryAddressFacade {
```

5.3 La query centrale e le precondizioni

feature-libs/checkout/base/core/facade/checkout-query.service.ts:L32-L54

```ts
  protected getCheckoutQueryReloadEvents(): QueryNotifier[] {
    return [CheckoutQueryReloadEvent];
  }
  /**
   * Returns the reset events for the checkout query.
   */
  protected getCheckoutQueryResetEvents(): QueryNotifier[] {
    return [CheckoutQueryResetEvent];
  }

  protected checkoutQuery$: Query<CheckoutState | undefined> =
    this.queryService.create<CheckoutState | undefined>(
      () =>
        this.checkoutPreconditions().pipe(
          switchMap(([userId, cartId]) =>
            this.checkoutConnector.getCheckoutDetails(userId, cartId)
          )
        ),
      {
        reloadOn: this.getCheckoutQueryReloadEvents(),
        resetOn: this.getCheckoutQueryResetEvents(),
      }
    );
```

feature-libs/checkout/base/core/facade/checkout-query.service.ts:L66-L84

```ts
  protected checkoutPreconditions(): Observable<[string, string]> {
    return combineLatest([
      this.userIdService.takeUserId(),
      this.activeCartFacade.takeActiveCartId(),
      this.activeCartFacade.isGuestCart(),
    ]).pipe(
      take(1),
      map(([userId, cartId, isGuestCart]) => {
```

```
      if (
        !userId ||
        !cartId ||
        (userId === OCC_USER_ID_ANONYMOUS && !isGuestCart)
      ) {
        throw new Error('Checkout conditions not met');
      }
      return [userId, cartId];
    })
  );
}
```

5.4 Un command con evento e una proiezione

feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L66-L94

```ts
protected setDeliveryAddressCommand: Command<Address, unknown> =
  this.commandService.create<Address>(
    (address) =>
      this.checkoutPreconditions().pipe(
        switchMap(([userId, cartId]) => {
          const addressId = address.id;
          if (!addressId) {
            throw new Error('Checkout conditions not met');
          }
          return this.checkoutDeliveryAddressConnector
            .setAddress(userId, cartId, addressId)
            .pipe(
              tap(() => {
                this.eventService.dispatch(
                  {
                    userId,
                    cartId,
                    address,
                  },
                  CheckoutDeliveryAddressSetEvent
                );
              })
            );
        })
      ),
    {
      strategy: CommandStrategy.CancelPrevious,
    }
  );
```

feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L153-L160

```ts
getDeliveryAddressState(): Observable<QueryState<Address | undefined>> {
  return this.checkoutQueryFacade.getCheckoutDetailsState().pipe(
    map((state) => ({
      ...state,
      data: state.data?.deliveryAddress,
    }))
  );
}
```

5.5 Listener che invalida la query

feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L109-L122

```ts
protected onDeliveryAddressSet(): void {
  this.subscriptions.add(
    this.eventService
      .get(CheckoutDeliveryAddressSetEvent)
      .subscribe(({ userId, cartId }) => {
        this.eventService.dispatch(
          { userId, cartId },
          CheckoutSupportedDeliveryModesQueryResetEvent
        );

        this.eventService.dispatch({}, CheckoutQueryResetEvent);
      })
  );
}
```

5.6 Step e flow

feature-libs/checkout/base/components/services/checkout-step.service.ts:L88-L100

```ts
resetSteps(): void {
  let steps = this.checkoutConfig.checkout?.steps ?? [];

  if (this.checkoutFlowOrchestratorService) {
    steps =
      this.checkoutFlowOrchestratorService.getCheckoutFlow()?.steps ?? [];
  }

  this.allSteps = steps
    .filter((step) => !step.disabled)
    .map((checkoutStep) => Object.assign({}, checkoutStep));
  this.steps$.next(this.allSteps);
}
```

feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L20-L44

```ts
constructor() {
  this.getPaymentProvider().subscribe((paymentProvider) => {
    this.paymentProviderName = paymentProvider;
  });
}

protected paymentProviderName: string | undefined = undefined;

getPaymentProvider(): Observable<string | undefined> {
  return this.baseSiteService.get().pipe(
    take(1),
    map((baseSite) => baseSite?.baseStore?.paymentProvider)
  );
}

getCheckoutFlow(): CheckoutFlow | undefined {
  if (this.paymentProviderName) {
    const flow =
      this.checkoutConfig.checkout?.flows?.[this.paymentProviderName];
    if (flow) {
      return flow;
    }
  }
  return this.checkoutConfig.checkout;
}
```

5.7 `CheckoutGuard` ed express checkout

feature-libs/checkout/base/components/guards/checkout.guard.ts:L41-L70

```ts
canActivate(): Observable<GuardResult> {
  const expressCheckout$ = this.expressCheckoutService
    .trySetDefaultCheckoutDetails()
    .pipe(
      switchMap((expressCheckoutPossible) => {
        const reviewOrderRoute =
          this.checkoutStepService.getCheckoutStepRoute(
            CheckoutStepType.REVIEW_ORDER
          );
        return expressCheckoutPossible && reviewOrderRoute
          ? of(
              this.router.parseUrl(
                this.routingConfigService.getRouteConfig(reviewOrderRoute)
                  ?.paths?.[0] as string
              )
            )
          : this.firstStep$;
      })
```

```
    );

  return this.activeCartFacade
    .isGuestCart()
    .pipe(
      switchMap((isGuestCart) =>
        this.checkoutConfigService.isExpressCheckout() && !isGuestCart
          ? expressCheckout$
          : this.firstStep$
      )
    );
}
```

5.8 Il guard degli step (cuore)

feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132

```ts
protected isStepSet(step: CheckoutStep): Observable<GuardResult> {
  if (step && !step.disabled) {
    switch (step.type[0]) {
      case CheckoutStepType.DELIVERY_ADDRESS: {
        return this.isDeliveryAddress(step);
      }
      case CheckoutStepType.DELIVERY_MODE: {
        return this.isDeliveryModeSet(step);
      }
      case CheckoutStepType.PAYMENT_DETAILS: {
        if (
          this.checkoutStepService.getCheckoutStep(
            CheckoutStepType.DELIVERY_MODE
          )?.disabled
        ) {
          this.checkoutDeliveryModesFacade.setDeliveryMode('pickup');
        }

        return this.isPaymentDetailsSet(step);
      }
      case CheckoutStepType.REVIEW_ORDER: {
        break;
      }
    }
  }
  return of(true);
}
```

5.9 Interceptor

feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L37-L61

```ts
    return this.routingService.getRouterState().pipe(
      take(1),
      switchMap((state: RouterState) => {
        return next.handle(request).pipe(
          tap({
            error: (response) => {
              if (
                response instanceof HttpErrorResponse &&
                this.isUserInCheckoutRoute(state.state?.semanticRoute)
              ) {
                if (this.isCartNotFoundError(response)) {
                  this.routingService.go({ cxRoute: 'cart' });

                  const cartCode = this.getCartIdFromError(response);
                  if (cartCode) {
                    this.multiCartFacade.reloadCart(cartCode);
                  }
                }
              }
            },
          })
        );
      })
    );
  }
```

5.10 SOP nell'adapter OCC

feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L55-L94

```ts
    return this.getProviderSubInfo(userId, cartId).pipe(
      map((data) => {
        const labelsMap = this.convertToMap(data.mappingLabels.entry) as {
          [key: string]: string;
        };
        return {
          url: data.postUrl,
          parameters: this.getParamsForPaymentProvider(
            paymentDetails,
            data.parameters.entry,
            labelsMap
          ),
          mappingLabels: labelsMap,
        };
      }),
      mergeMap((sub) =>
        // create a subscription directly with payment provider
        this.createSubWithProvider(sub.url, sub.parameters).pipe(
```

```
        map((response) => this.extractPaymentDetailsFromHtml(response)),
        mergeMap((fromPaymentProvider) => {
          fromPaymentProvider['defaultPayment'] =
            paymentDetails.defaultPayment ?? false;
          fromPaymentProvider['savePaymentInfo'] = true;
          return this.createDetailsWithParameters(
            userId,
            cartId,
            fromPaymentProvider
          ).pipe(
            catchError((error) => {
              throw tryNormalizeHttpError(error, this.logger);
            }),
            backOff({
              shouldRetry: isJaloError,
            }),
            this.converter.pipeable(PAYMENT_DETAILS_NORMALIZER)
          );
        })
      )
    )
  );
```

5.11 Place order con gate di stabilità

feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L124-L142

```ts
  submitForm(): void {
    if (!this.checkoutSubmitForm.valid) {
      this.checkoutSubmitForm.markAllAsTouched();
      return;
    }
    if (!this.featureToggles.enableCartSlowNetworkResilience) {
      this.launchPlaceOrder();
      return;
    }
    this.activeCartFacade
      .isStable()
      .pipe(take(1))
      .subscribe((isStable) => {
        if (!isStable) {
          return;
        }
        this.launchPlaceOrder();
      });
  }
```

5.12 B2B: augmentation e override dei guard

feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23

```ts
declare module '@spartacus/checkout/base/root' {
  enum CheckoutStepType {
    PAYMENT_TYPE = 'paymentType',
  }

  interface CheckoutState {
    costCenter?: CostCenter;
    purchaseOrderNumber?: string;
    paymentType?: PaymentType;
  }
}

(CheckoutStepType as any)['PAYMENT_TYPE'] = 'paymentType';
```

feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37

```ts
  providers: [
    {
      provide: CheckoutAuthGuard,
      useExisting: CheckoutB2BAuthGuard,
    },
    {
      provide: CheckoutStepsSetGuard,
      useExisting: CheckoutB2BStepsSetGuard,
    },
  ],
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
checkout.steps	CheckoutStep[]	4 step base / 5 step B2B	feature-libs/checkout/base/root/config/default-checkout-config.ts:L12-L37; feature-libs/checkout/b2b/root/config/default-b2b-checkout-config.ts:L15-L46	feature-libs/checkout/base/components/services/checkout-step.service.ts:L88-L100

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
checkout.express	boolean	false	feature-libs/checkout/base/root/config/default-checkout-config.ts:L38-L38	feature-libs/checkout/base/components/services/checkout-config.service.ts:L23-L23, feature-libs/checkout/base/components/services/checkout-config.service.ts:L33-L33; feature-libs/checkout/base/components/guards/checkout.guard.ts:L65-L65
checkout.defaultDeliveryMode	(DeliveryModePreferences | string)[]	[FREE]	feature-libs/checkout/base/root/config/default-checkout-config.ts:L39-L39	feature-libs/checkout/base/components/services/checkout-config.service.ts:L25-L26, feature-libs/checkout/base/components/services/checkout-config.service.ts:L35-L35
checkout.guest	boolean	false (app demo: true)	feature-libs/checkout/base/root/config/default-checkout-config.ts:L40-L40; projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L99-L103	feature-libs/checkout/base/components/services/checkout-config.service.ts:L24-L24, feature-libs/checkout/base/components/services/checkout-config.service.ts:L34-L34; feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L62-L69
checkout.guestUseSavedAddress	boolean	non definito	feature-libs/checkout/base/root/config/checkout-config.ts:L43-L43	feature-libs/checkout/base/components/services/checkout-config.service.ts:L93-L99
checkout.flows[paymentProvider]	CheckoutFlow	non definito	feature-libs/checkout/base/root/config/checkout-config.ts:L47-L49	feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L35-L44

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes.checkout*	RoutingConfig	vedi §3.4	feature-libs/checkout/base/root/config/default-checkout-routing-config.ts:L9-L20; feature-libs/checkout/b2b/root/config/default-checkout-b2b-routing-config.ts:L9-L15	feature-libs/checkout/base/components/services/checkout-step.service.ts:L188-L192, guard
featureModules.checkout / checkoutCore	CmsConfig	vedi §3.1, §3.9, §3.10	feature-libs/checkout/base/root/checkout-root.module.ts:L37-L49	lazy loading CMS / facadeFactory
backend.occ.endpoints.*	OccEndpoints	13 base + B2B	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L12-L35; feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L16-L29	adapter OCC
launch.PLACE_ORDER_SPINNER	LayoutConfig	inline, SpinnerComponent, POPOVER_CENTER_BACKDROP	feature-libs/checkout/base/components/checkout-place-order/default-place-order-spinner-layout.config.ts:L13-L21	feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L145-L148
i18n.chunks (checkout, checkoutB2B, checkoutScheduledReplenishment)	TranslationChunksConfig	17 lingue	feature-libs/checkout/base/assets/translations/translations.ts:L9-L36; feature-libs/checkout/b2b/assets/translations/translations.ts:L9-L11; feature-libs/checkout/scheduled-replenishment/assets/translations/translations.ts:L9-L12	app (provideConfig), vedi 37 — i18n

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `enableCartSlowNetworkResilience`	feature toggle	—	core (vedi 42)	feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L129-L129; feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.html:L36-L62
toggle `authorizationCodeFlowByDefault`	feature toggle	—	core	feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L61-L61
toggle `enableFormFieldMaxLength`	feature toggle	—	core	feature-libs/checkout/base/components/checkout-billing-address/checkout-billing-address-form.service.ts:L24-L26
toggle `b2bCheckoutShippingAddressFilter`	feature toggle	—	core	feature-libs/checkout/b2b/components/checkout-delivery-address/checkout-delivery-address.component.ts:L86-L88

Endpoint OCC base (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L9-L35):

ENDPOINT	URL	USATO DA
`setDeliveryAddress`, `createDeliveryAddress`, `removeDeliveryAddress`	`users/${userId}/carts/${cartId}/addresses/delivery`	`OccCheckoutDeliveryAddressAdapter` (PUT con `addressId`, POST, DELETE) (feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-address.adapter.ts:L37-L133)
`setDeliveryMode`, `clearDeliveryMode`, `deliveryMode`	`.../deliverymode`	`OccCheckoutDeliveryModesAdapter` (PUT con `deliveryModeId`, DELETE) (feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-modes.adapter.ts:L38-L116); `deliveryMode` non usato nel repo: nessun `buildUrl('deliveryMode'…)` in `feature-libs`, `integration-libs`, `core-libs`, `projects`; la chiave compare solo nel tipo (feature-libs/checkout/base/occ/model/occ-checkout-endpoints.model.ts:L45-L45)
`deliveryModes`	`.../deliverymodes`	GET supportate
`setCartPaymentDetails`	`.../paymentdetails`	PUT (`paymentDetailsId`), DELETE (feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L97-L146)
`paymentProviderSubInfo`, `createPaymentDetails`	`.../payment/sop/request?responseUrl=sampleUrl`, `.../payment/sop/response`	SOP

ENDPOINT	URL	USATO DA
cardTypes	cardtypes	GET card types (feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L148-L165)
getCheckoutDetails	users/${userId}/carts/${cartId}?fields=deliveryAddress(FULL),deliveryMode(FULL),paymentInfo(FULL)	OccCheckoutAdapter
setBillingAddress	.../addresses/billing	PUT con indirizzo nel body (feature-libs/checkout/base/occ/adapters/occ-checkout-billing-address.adapter.ts:L22-L55)

Esempio di configurazione applicativa (il frammento `provideConfig({ checkout: { guest: true } })` è quello reale dell'app demo, projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L99-L103). Per un flow per payment provider la forma attesa è `checkout.flows['<nome provider>'] = { steps: [...], express, guest, ... }` secondo l'interfaccia `CheckoutFlow` (feature-libs/checkout/base/root/model/checkout-flow.model.ts:L10-L31); nessun default ne definisce uno in questa lib.

---

7. Commenti e note notevoli dal sorgente

- `// TODO:#deprecation-checkout - remove once we remove ngrx` sull'`async: true` delle facade: il `delay(0)` è un residuo dell'era NgRx (feature-libs/checkout/base/root/facade/checkout-delivery-address.facade.ts:L24-L25).
- `// by default core is bundled together with components`: spiega l'alias `checkoutCore → checkout`; separare core e componenti in chunk diversi richiede di ridefinire quella chiave (feature-libs/checkout/base/root/checkout-root.module.ts:L44-L45).
- JSDoc dell'interceptor: "he is redirected to checkout and the cart is reloaded" — ma il codice naviga a `cart` (feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L20-L25 vs feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L48-L48).
- "we want to LL the checkout (if not already loaded), in order to clear the checkout data that's potentially set on the back-end": il listener usa la facade proprio per forzare il lazy loading (feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L74-L74).
- "As we know the cart is not anonymous (precondition checked), we can safely use the cartId, which is actually the cart.code." (feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L104-L107; stesso commento in feature-libs/order/core/facade/order.service.ts:L38-L41).
- JSDoc di `PLACE_ORDER_GATE_SAFETY_VALVE_MS` e `isCartUpdating$`: il bottone è disabilitato finché il carrello non è stabile, per evitare il "phantom cart" (CXSPA-10582), con valvola di sicurezza di 10 secondi (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L47-L53, feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L83-L100). Attenzione: la promessa del JSDoc ("cannot lock the user out of placing the order") vale solo per il bottone. La valvola agisce su `isCartUpdating$`, cioè sul `[disabled]`; `submitForm()` ricontrolla `isStable()` con `take(1)` ed esce in silenzio se è `false` (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L133-L141, feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.html:L36-L40). Inoltre il `timer` è nell'inizializzatore del campo: i 10 s partono dalla sottoscrizione di `isCartUpdating$` (render del componente), non dal click.
- `TODO:#9574 - we should be able to remove the spread of CHECKOUT_BASE_CMS_COMPONENTS` in B2B e replenishment: conseguenza della sostituzione degli array nel merge (feature-libs/checkout/b2b/root/

checkout-b2b-root.module.ts:L24-L28, feature-libs/checkout/scheduled-replenishment/root/checkout-scheduled-replenishment-root.module.ts:L14-L18).

- `// Import augmented types first` nel root B2B: l'assegnazione runtime su `CheckoutStepType` deve avvenire prima che le config B2B usino `PAYMENT_TYPE` (feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L8-L9).
- `// TODO:#checkout - handle breaking changes in schematics` sul parametro `UnifiedInjector` di `CmsGuardsService`, introdotto proprio per gli override lazy dei guard (core-libs/storefront/cms-structure/services/cms-guards.service.ts:L25-L26).
- TODO CXSPA-1276: `verifyAddress()` del billing form non è usato (feature-libs/checkout/base/components/checkout-billing-address/checkout-billing-address-form.component.ts:L165-L178).
- Deprecazioni: `getCardRole()` `@deprecated since 221121.8` (feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L388-L398); `useExtractedBillingAddressComponent` obsoleto dal 2211.42 (feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-form/checkout-payment-form.component.ts:L172-L175).

8. Extension point

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
Ordine/insieme degli step	`provideConfig({ checkout: { steps: [...] } })` ridefinendo **l'intero** array	feature-libs/checkout/base/root/config/default-checkout-config.ts:L10-L42, core-libs/core/src/config/utils/deep-merge.ts:L31-L46	Parte III, cap. 53
Nuovo tipo di step	Augmentation di `CheckoutStepType` (tipo + assegnazione runtime, come in B2B) + rotta semantica + componente CMS	feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23, feature-libs/checkout/base/root/public_api.ts:L14-L17	Parte III, cap. 53
Flow per payment provider	`checkout.flows[provider]`	feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L35-L44	Parte III, cap. 53
Stato aggiuntivo nel checkout	Augmentation di `CheckoutState` + override endpoint `getCheckoutDetails` (`fields`)	feature-libs/checkout/b2b/root/model/augmented-types.ts:L16-L20, feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L11-L29	Parte III, cap. 52

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
Sostituire/estendere un guard	`{ provide: CheckoutStepsSetGuard, useExisting: MyGuard }` nel modulo (anche lazy); metodi `protected isStepSet/is*Set` sovrascrivibili	feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37, feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L172	Parte III, cap. 53
Sostituire un componente di step	`cmsComponents: { CheckoutDeliveryAddress: { component: ... } }` (come fa il B2B)	feature-libs/checkout/b2b/components/checkout-delivery-address/checkout-delivery-address.module.ts:L38-L44	Parte III, cap. 51
Raggruppamento review	Override di `getCheckoutDeliverySteps`/`getCheckoutPaymentSteps`	feature-libs/checkout/base/components/checkout-review-submit/checkout-review-submit.component.ts:L101-L110	Parte III, cap. 51
Contenuto extra nel delivery mode	Outlet `CartOutlets.DELIVERY_MODE`	feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.html:L55-L56	Parte III, cap. 51
Eventi di invalidazione della query	Sottoclasse di `CheckoutQueryService` con `getCheckoutQueryReloadEvents()`/`getCheckoutQueryResetEvents()`	feature-libs/checkout/base/core/facade/checkout-query.service.ts:L32-L40	Parte III, cap. 53
Reagire ai passaggi del checkout	`eventService.get(CheckoutDeliveryModeSetEvent)` ecc.	feature-libs/checkout/base/root/events/checkout.events.ts:L12-L203	Parte III, cap. 50
Backend diverso da OCC	Fornire i 5 adapter astratti (`CheckoutAdapter`, `CheckoutDeliveryAddressAdapter`, `CheckoutDeliveryModesAdapter`, `CheckoutPaymentAdapter`, `CheckoutBillingAddressAdapter`)	feature-libs/checkout/base/occ/checkout-occ.module.ts:L24-L50	Parte III, cap. 52

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
Endpoint / converter	Override `backend.occ.endpoints.*`; converter su `CHECKOUT_NORMALIZER`, `DELIVERY_MODE_NORMALIZER`, `PAYMENT_DETAILS_SERIALIZER`, `PAYMENT_CARD_TYPE_NORMALIZER`, `PAYMENT_DETAILS_NORMALIZER`	`feature-libs/checkout/base/core/connectors/checkout/converters.ts:L11-L13`, `feature-libs/checkout/base/core/connectors/checkout-delivery-modes/converters.ts:L11-L13` (usato in `feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-modes.adapter.ts:L83-L83`), `feature-libs/checkout/base/core/connectors/checkout-payment/converters.ts:L10-L16`, `core-libs/core/src/checkout/connectors/payment/converters.ts:L11-L13`	Parte III, cap. 52
Meta della pagina checkout	`CheckoutPageMetaResolver` registrato come `PageMetaResolver` multi	`feature-libs/checkout/base/core/checkout-core.module.ts:L17-L33`, `feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts:L40-L76`	Parte III, cap. 52
Separare core dai componenti	Ridefinire `featureModules.checkoutCore` come modulo autonomo	`feature-libs/checkout/base/root/checkout-root.module.ts:L44-L45`	Parte III, cap. 53

Simboli non pubblici (da non usare come API stabile, vedi Parte III, cap. 54): `CheckoutCartInterceptor` (`http-interceptors` non è esportato dal public_api root, `feature-libs/checkout/base/root/public_api.ts:L7-L17`), `defaultPlaceOrderSpinnerLayoutConfig` (`feature-libs/checkout/base/components/public_api.ts:L22-L23` esporta solo component e module del place order), `BadCostCenterRequestHandler` (`feature-libs/checkout/b2b/core/public_api.ts:L7-L9` esporta solo module, `connectors/index` e `facade/index`). Nessun altro `index.ts`/`public_api.ts` della lib li esporta.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Gli array di config si sostituiscono.** Chi ridefinisce `checkout.steps` deve ridefinire tutta la lista; in B2B gli step B2B rimpiazzano quelli base, e i `cmsComponents` B2B devono ri-spalmare la lista base (`core-libs/core/src/config/utils/deep-merge.ts:L7-L9`, `core-libs/core/src/config/utils/deep-merge.ts:L38-L45`, `feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L23-L31`). Gli oggetti invece si fondono: la `dependencies` della feature base sopravvive alla ridefinizione B2B (`feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L33-L42`).
2. **Il config "puro" non è mai letto direttamente.** `CheckoutFlowOrchestratorService` è root, quindi `if (this.checkoutFlowOrchestratorService)` è sempre vero e tutto passa da `getCheckoutFlow()` (`feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L13-L15`, `feature-libs/checkout/base/components/services/checkout-step.service.ts:L91-L94`).

3. **Race sul payment provider.** `paymentProviderName` è valorizzato in una subscribe; `CheckoutConfigService.checkoutFlow` e `CheckoutStepService.resetSteps()` leggono il flow una sola volta alla costruzione: se il base site non è ancora disponibile, si usa il flow di default (feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L20-L26, feature-libs/checkout/base/components/services/checkout-config.service.ts:L28-L29). Il meccanismo è confermato dal codice: `BaseSiteService.get()` filtra sia il site attivo (`filter((active) => Boolean(active))`) sia la lista dei site (`filter(isNotNullable)`), quindi finché i site non sono caricati la `take(1)` dell'orchestratore non emette e `paymentProviderName` resta `undefined` (core-libs/core/src/site-context/facade/base-site.service.ts:L31-L51, core-libs/core/src/site-context/facade/base-site.service.ts:L56-L70). DA VERIFICARE a runtime solo se, in pratica, il checkout viene istanziato prima del caricamento dei site.
4. **Solo lo step precedente è verificato.** Navigando direttamente al review, `CheckoutStepsSetGuard` controlla solo il pagamento, non indirizzo e delivery mode (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L92-L93). Inoltre il guard è sui componenti progress, non sugli step: una pagina di step senza `CheckoutProgress` non è protetta (feature-libs/checkout/base/components/checkout-progress/checkout-progress.module.ts:L33-L40).
5. **Effetti collaterali nei guard.** `setDeliveryMode('pickup')` è fire-and-forget e funziona solo perché la chiamata facade e `Command.execute` sono eager (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L121-L121, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L75-L82). Il guard B2B non replica questa logica (feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L114-L116).
6. **Il guard degli step muta lo stato globale.** La subscription su `hasDeliveryItems()` vive quanto l'app (servizio root) e muta `nameMultiLine` sugli oggetti di `allSteps` (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L49-L70, feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L174-L182).
7. **Precondizioni fallite = "step non impostato".** Se le precondizioni lanciano, la query va in errore con `data: undefined`; i guard filtrano solo `!loading` e quindi redirigono allo step (core-libs/core/src/util/command-query/query.service.ts:L98-L101, feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L134-L145).
8. **Delivery mode: un PUT a ogni emissione.** Il componente chiama `changeMode` a ogni emissione di `supportedDeliveryModes$`, anche se il codice è già selezionato (feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.ts:L84-L96).
9. **Preferenze delivery mode.** `compareDeliveryCost` restituisce 0 se uno dei costi è 0 (falsy), quindi le modalità gratuite non vengono ordinate; `FREE` guarda solo `deliveryModes[0]`; `findMatchingDeliveryMode` accede a `deliveryModes[0]` senza controllare la lunghezza; `getPreferredDeliveryMode` ordina **in place** l'array ricevuto (feature-libs/checkout/base/components/services/checkout-config.service.ts:L39-L104).
10. `getCheckoutStepIndex` **può restituire** `-1`, e `getCheckoutStep` legge allora `allSteps[-1]` (undefined) (feature-libs/checkout/base/components/services/checkout-step.service.ts:L115-L120, feature-libs/checkout/base/components/services/checkout-step.service.ts:L194-L203).
11. **SOP in SSR.** `DOMParser` è istanziato solo se esiste, ma `extractPaymentDetailsFromHtml` lo usa senza guard (feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L39-L44, feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L315-L315). E `street1` è `line1 + ' ' + line2`: con `line2` undefined diventa la stringa `"... undefined"` (feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L295-L298).
12. **Billing address senza eventi.** Il set del billing address non dispatcha eventi né resetta la query (feature-libs/checkout/base/core/facade/checkout-billing-address.service.ts:L36-L53).
13. **Strategie diverse.** `deletePaymentMethodCommand` usa `Queue`, tutti gli altri `CancelPrevious` (feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L112-L123).

14. **Card types senza filtro** `loading`. `getPaymentCardTypes()` emette `[]` iniziale, `getSupportedDeliveryModes()` invece attende `!loading` (feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L162-L166, feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L173-L178).
15. **Interceptor: JSDoc fuorviante** (vedi §7) e attivo su qualunque rotta semantica che *inizi* con `checkout` (feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L75-L77).
16. **B2B senza** `async`. Le facade B2B non hanno `async: true` a differenza delle base (feature-libs/checkout/b2b/root/facade/checkout-cost-center.facade.ts:L15-L20). `CheckoutCostCenterComponent` accede a `costCenters[0].code` senza controllare la lista e non sottoscrive il risultato del set (feature-libs/checkout/b2b/components/checkout-cost-center/checkout-cost-center.component.ts:L74-L88).
17. **Replenishment.** `replenishmentStartDate` è calcolato una volta alla costruzione del servizio root con `toISOString()` (UTC) (feature-libs/checkout/scheduled-replenishment/components/services/checkout-replenishment-form.service.ts:L39-L46); il place order esteso non chiama `super.ngOnInit()` (feature-libs/checkout/scheduled-replenishment/components/checkout-place-order/checkout-place-order.component.ts:L136-L155).
18. **DOM fuori da Angular.** `CheckoutReviewOverviewComponent` sposta nodi DOM con `requestAnimationFrame` (feature-libs/checkout/base/components/checkout-review/checkout-review-overview/checkout-review-overview.component.ts:L61-L72); delivery mode e payment form usano `document` globale (feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.ts:L187-L191).
19. `@Optional()` **inutile** su proprietà inizializzate con `inject()` (feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L100-L101).

---

10. Come fare debug

- **Il chunk non si carica / "Feature checkoutCore is not configured properly".** Il proxy lancia questo errore se la feature non è configurata (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47): verificare che `CheckoutRootModule` sia importato e che `featureModules.checkout.module` sia fornito dall'app (projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L41-L87).
- **Stato del checkout.** Non c'è uno slice NgRx da ispezionare: nella scheda Network cercare il GET con `fields=deliveryAddress(FULL),deliveryMode(FULL),paymentInfo(FULL)` (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L29-L30); ogni reset della query provoca un nuovo GET se qualcuno è sottoscritto (core-libs/core/src/util/command-query/query.service.ts:L117-L133).
- **Tracciare gli eventi.** In un componente o `APP_INITIALIZER` di debug, `inject(EventService).get(CheckoutEvent)` riceve tutta la famiglia (feature-libs/checkout/base/root/events/checkout.events.ts:L32-L48); `CheckoutQueryResetEvent` e `CheckoutQueryReloadEvent` vanno ascoltati a parte perché estendono `CxEvent` direttamente (feature-libs/checkout/base/root/events/checkout.events.ts:L12-L203).
- **"Checkout conditions not met".** Viene dalle precondizioni (utente anonimo senza guest cart, carrello mancante) o da `address.id`/`paymentDetails.id` assenti (feature-libs/checkout/base/core/facade/checkout-query.service.ts:L66-L84, feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L71-L74), o da un billing address vuoto/assente passato a `setBillingAddress` (feature-libs/checkout/base/core/facade/checkout-billing-address.service.ts:L40-L42).
- **Redirect inattesi tra step.** Cercare in console il warning dev-mode `Missing step with route ...` (feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L95-L99); verificare quale step è disabilitato (`steps$`), quale guard è effettivamente attivo (in B2B gli override `useExisting`, feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37) e lo stato della query (errore = step mancante, §9.7).

- **Flow sbagliato.** Controllare `baseSite.baseStore.paymentProvider` e la chiave in `checkout.flows` (feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L28-L44).
- **Bottone Place Order disabilitato.** Con `enableCartSlowNetworkResilience` il bottone è disabilitato mentre `isStable()` è `false`, al massimo per 10 s dalla sottoscrizione di `isCartUpdating$` (cioè dal render del componente) (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L94-L100). Scaduta la valvola il bottone torna attivo, ma `submitForm()` ricontrolla `isStable()` e, se è ancora `false`, esce senza piazzare l'ordine e senza feedback (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L133-L141). La valvola sblocca quindi solo il bottone, non il place order. Stesso schema nel place order replenishment (feature-libs/checkout/scheduled-replenishment/components/checkout-place-order/checkout-place-order.component.ts:L94-L102).
- **Test.** La lib usa Vitest (jsdom, `pool: 'forks'`) (feature-libs/checkout/vitest.config.ts:L16-L66): `nx test-vitest checkout` o `npx vitest run --config feature-libs/checkout/vitest.config.ts <spec>` (comandi da `.claude/CLAUDE.md`; il nome progetto Nx `checkout` e il target `test-vitest` sono in feature-libs/checkout/project.json:L2-L2 e feature-libs/checkout/project.json:L20-L20).

---

11. Domande di autoverifica

1. **Perché un listener può far scaricare il chunk del checkout?** Perché inietta una facade proxy (`facadeFactory` con feature `checkoutCore`, alias di `checkout`) e la chiamata a un metodo risolve la feature e la carica; la chiamata è eager. — feature-libs/checkout/base/root/checkout-root.module.ts:L44-L45, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L92, feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L74-L75
2. **Dove si trova l'indirizzo di consegna selezionato nel client?** Non in uno store: è la proiezione `state.data?.deliveryAddress` dell'unica query `checkoutQuery$`, caricata da `getCheckoutDetails`. — feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L153-L160, feature-libs/checkout/base/core/facade/checkout-query.service.ts:L42-L54
3. **Cosa succede dopo un `setDeliveryAddress` riuscito?** Evento `CheckoutDeliveryAddressSetEvent` → reset delivery modes supportate (→ `LoadCartEvent`) e reset della query; il componente inoltre cancella la delivery mode. — feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L66-L94, feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L109-L122, feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L357-L374
4. **Aggiungo uno step con `provideConfig({ checkout: { steps: [nuovo] } })`: cosa ottengo?** Un checkout con il solo step nuovo, perché `deepMerge` sostituisce gli array: va ridefinita l'intera lista. — core-libs/core/src/config/utils/deep-merge.ts:L31-L46
5. **Perché in B2B il guard `CheckoutStepsSetGuard` viene sostituito anche se è dichiarato nel CMS config base?** `CmsGuardsService` risolve i guard con `getLastValueSync(unifiedInjector.get(...))`, cioè dall'ultimo injector (anche lazy) che fornisce il token; il modulo B2B fornisce `useExisting: CheckoutB2BStepsSetGuard`. — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L41-L46, feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37
6. **Un utente arriva direttamente al review order senza delivery mode ma con un pagamento: il guard lo blocca?** No: `CheckoutStepsSetGuard` verifica solo lo step immediatamente precedente (payment details). — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L92-L93
7. **Quando il `CheckoutGuard` porta direttamente al review?** Se `isExpressCheckout()` è vero, il carrello non è guest e `ExpressCheckoutService` riesce a impostare indirizzo, delivery mode e pagamento di default. —

feature-libs/checkout/base/components/guards/checkout.guard.ts:L41-L70, feature-libs/checkout/base/components/services/express-checkout.service.ts:L48-L175

8. **Chi piazza l'ordine e come il checkout lo viene a sapere?** `OrderFacade.placeOrder` della lib order; `OrderService` dispatcha `OrderPlacedEvent`, a cui il checkout reagisce con `RemoveCartEvent` e reset della query. — feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L144-L166, feature-libs/order/core/facade/order.service.ts:L26-L54, feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L24-L41
9. **Come fa il B2B ad aggiungere `PAYMENT_TYPE` a un enum di un'altra libreria?** Con `declare module '@spartacus/checkout/base/root'` per il tipo e con un'assegnazione runtime sull'oggetto enum, importata per prima nel root module. — feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23, feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L8-L9
10. **Cosa fa il checkout se una richiesta restituisce "Cart not found" durante uno step?** L'interceptor, se la rotta semantica inizia con `checkout` e l'errore è 400 `CartError/notFound`, naviga al carrello e ricarica il carrello indicato in `subject`. — feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L37-L94

TAVOLA 26.1 **4.1 Bootstrap e lazy loading**

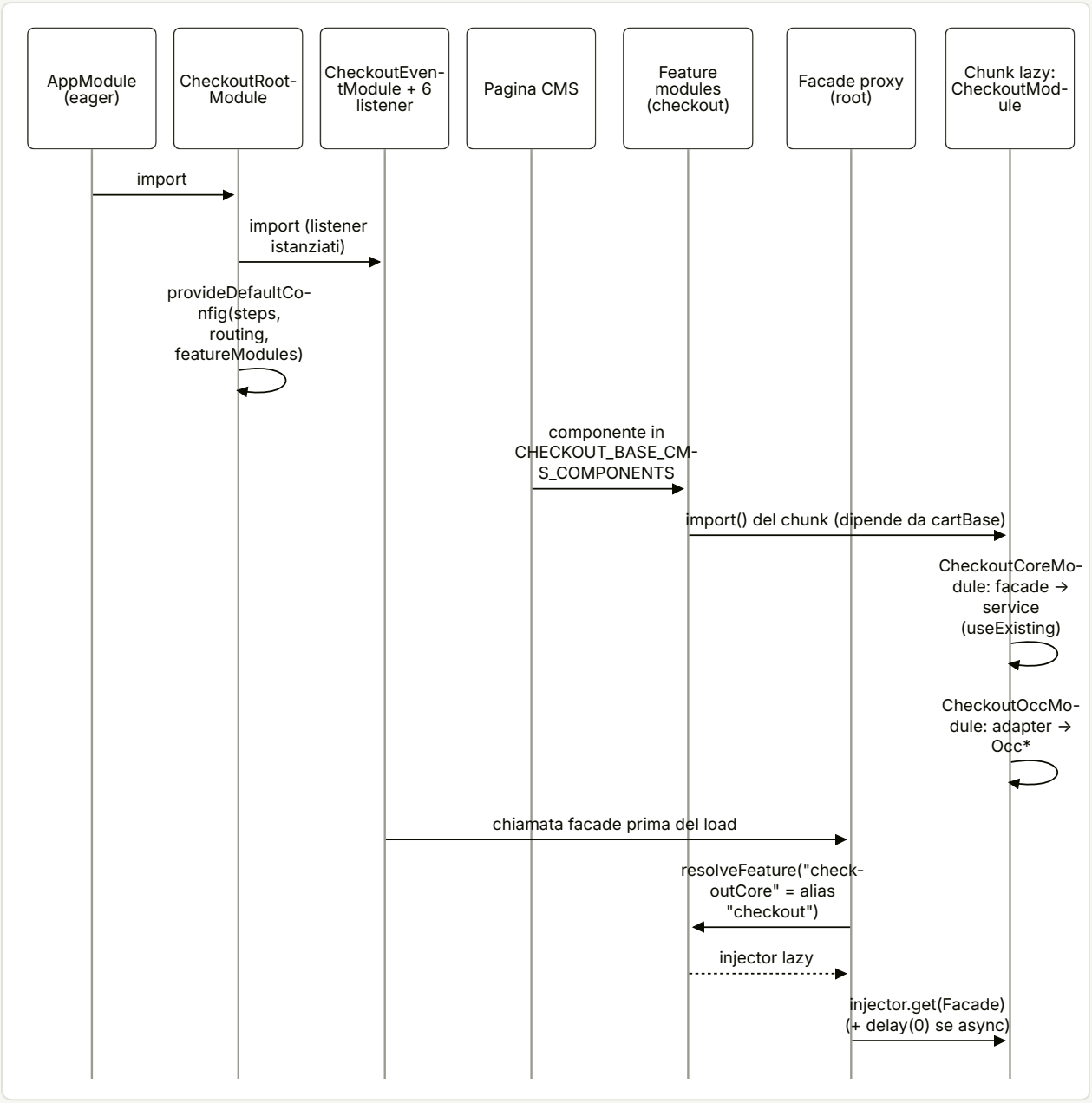
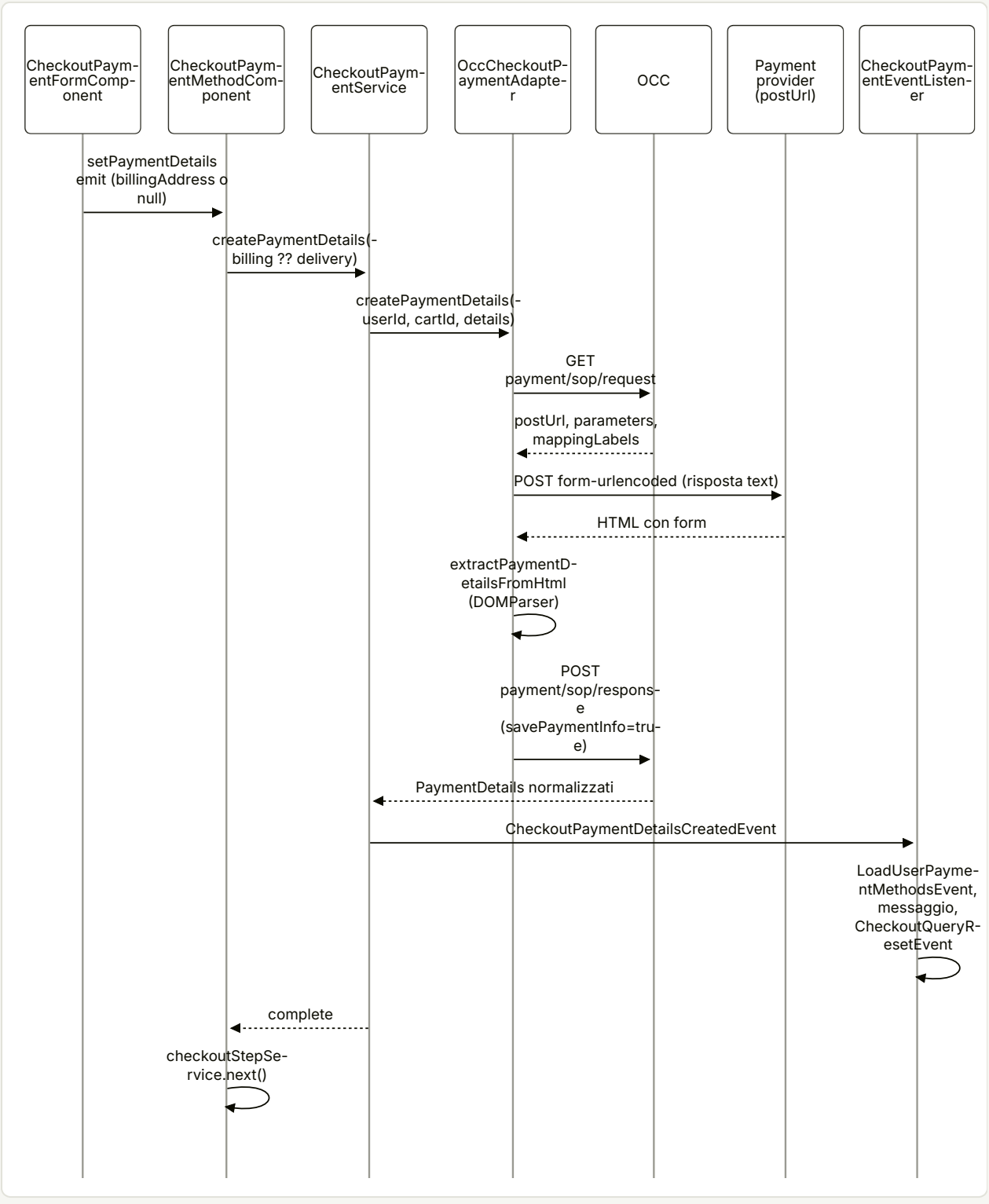


TAVOLA 26.2 **4.4 Silent Order Post (nuova carta)**



27

Ordini

Place order, storia e dettaglio ordini, resi, ordini ricorrenti, reorder, allegati.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 12 — Feature module e lazy loading in Spartacus, 19 — State management, 20 — Facade, command & query, eventi, 21 — Livello OCC, 14 — Routing. Collegati: 26 — Checkout (chi chiama il place order), 25 — Carrello (reorder e rimozione del carrello), 16 — CMS rendering (guard e `providers` dei componenti CMS), 17 — Outlet, 18 — Layout, breakpoint, launch dialog, 28 — B2B organization (endpoint `orgUsers`).

1. A cosa serve e perché esiste

La libreria `@spartacus/order` copre tutto ciò che accade **dopo** (e nell'istante in cui) il carrello diventa un ordine:

- il **place order** e la pagina di conferma (`OrderFacade`);
- la **storia ordini** e il **dettaglio ordine** in "My Account", con il tracking delle spedizioni (`OrderHistoryFacade`);
- la **cancellazione** e il **reso** (return request), con le relative liste e dettagli (`OrderReturnRequestFacade`);
- gli **ordini ricorrenti** (scheduled replenishment) (`ScheduledReplenishmentOrderFacade`, `ReplenishmentOrderHistoryFacade`);
- il **reorder** (ricrea un carrello da un ordine) (`ReorderOrderFacade`);
- gli **allegati** dell'ordine (`OrderAttachmentsFacade`);
- due estensioni "laterali": il **document flow** (`@spartacus/order/document-flow`, documenti successivi SAP) e le **fatture PDF** (libreria separata `@spartacus/pdf-invoices`).

Il punto di ingresso pesante è `OrderModule`, che importa nell'ordine core, OCC e componenti (`feature-libs/order/order.module.ts:L12-L15`). Come per le altre feature lib (vedi capitolo 12) esiste un livello `root` eager — configurazione di default, rotte, facade astratte, eventi, token — e un livello lazy caricato alla prima necessità.

Tre scelte progettuali caratterizzano la libreria e ritornano in tutto il capitolo:

1. **Due modelli di stato convivono.** Il place order, la schedulazione di un replenishment, il reorder e gli allegati usano il pattern *Command* o chiamate dirette al connector, senza NgRx: l'ordine appena piazzato vive in un `BehaviorSubject` (`feature-libs/order/core/facade/order.service.ts:L22-L24`). Storia ordini, dettaglio, resi, replenishment e tracking usano invece uno store NgRx di feature `order` con 9 slice (6 loader, 2 entity loader e lo stato semplice `consignmentTracking`) (`feature-libs/order/core/store/order-state.ts:L37-L51`, `feature-libs/order/core/store/reducers/index.ts:L60-L60`).
2. **I componenti di dettaglio sono riusati cambiando la sorgente dati via DI.** `OrderOverviewComponent`, `OrderDetailItemsComponent` ecc. leggono da `OrderDetailsService`; la pagina di conferma ordine e quella di dettaglio replenishment sostituiscono quel servizio nei `providers` del mapping CMS con `OrderFacade` o `ReplenishmentOrderDetailsService` (`feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L96-L129`, `feature-libs/order/components/replenishment-order-details/replenishment-order-details.module.ts:L51-L87`).
3. **Le rotte sono semantiche e senza path.** `OrderRootModule` registra 11 rotte con `path: null` e un `data.cxRoute`; i path reali vengono dalla routing config di default (`feature-libs/order/root/order-root.module.ts:L82-L172`, `feature-libs/order/root/config/default-order-routing-config.ts:L9-L54`), secondo il meccanismo descritto nel capitolo 14.

2. Mappa dei file

PATH	RUOLO
feature-libs/order/order.module.ts	OrderModule: core + OCC + components (:L12-L15).
feature-libs/order/public_api.ts	Entry @spartacus/order, riesporta solo il modulo (:L7-L16).
feature-libs/order/root/feature-name.ts	ORDER_FEATURE = 'order', ORDER_CORE_FEATURE = 'orderCore' (:L7-L8).
feature-libs/order/root/order-root.module.ts	ORDER_CMS_COMPONENTS (36 tipi), defaultOrderComponentsConfig(), 11 rotte path: null, OrderRootModule (:L28-L180).
feature-libs/order/root/config/order-config.ts	OrderConfig astratta + augmentation di Config (:L10-L23).
feature-libs/order/root/config/default-order-config.ts	previewMimeTypes (13 MIME), showOrderQuoteLink: false (:L9-L28).
feature-libs/order/root/config/default-order-routing-config.ts	12 rotte semantiche e paramsMapping (:L9-L54).
feature-libs/order/root/facade/*.facade.ts	7 facade astratte lazy (facadeFactory su orderCore).
feature-libs/order/root/events/order.events.ts	OrderEvent, OrderPlacedEvent, ReplenishmentOrderScheduledEvent, DownloadOrderInvoicesEvent (:L14-L72).
feature-libs/order/root/model/order.model.ts	Order, OrderHistory, ReturnRequest, input cancel/return (:L29-L146).
feature-libs/order/root/model/replenishment-order.model.ts	ReplenishmentOrder, Trigger (:L26-L81).
feature-libs/order/root/model/scheduled-replenishment.model.ts	ScheduleReplenishmentForm, recurrencePeriod, ORDER_TYPE (:L7-L35).
feature-libs/order/root/model/order-view.model.ts	View model per My Account v2 (:L17-L38).
feature-libs/order/root/model/order-outlets.model.ts	OrderOutlets (3 outlet) (:L10-L14).
feature-libs/order/root/model/augmented.model.ts	Augmentation di LAUNCH_CALLER (:L9-L21).
feature-libs/order/root/tokens/context.ts	Token contesto entries + USE_MY_ACCOUNT_V2_ORDER (:L9-L20).
feature-libs/order/root/tokens/converters.ts	10 token normalizer/serializer (:L24-L62).
feature-libs/order/root/utils/order-card-utils.ts	Builder di Card (indirizzo, delivery mode, pagamento) (:L14-L72).
feature-libs/order/core/order-core.module.ts	OrderCoreModule: store + facade providers + 6 connector (:L17-L29).
feature-libs/order/core/facade/facade-providers.ts	Facade astratta → service concreto via useExisting (:L26-L63).
feature-libs/order/core/facade/order.service.ts	Place order: BehaviorSubject + Command (:L22-L191).
feature-libs/order/core/facade/order-history.service.ts	Dettaglio, lista, tracking, cancel via NgRx (:L41-L225).
feature-libs/order/core/facade/order-return-request.service.ts	Resi via NgRx (:L63-L79).
feature-libs/order/core/facade/replenishment-order-history.service.ts	Replenishment via NgRx (:L43-L160).

PATH	RUOLO
feature-libs/order/core/facade/scheduled-replenishment-order.service.ts	Schedulazione (Command) (:L32-L97).
feature-libs/order/core/facade/reorder-order.service.ts	Reorder (Command + cancellazione carrello) (:L24-L93).
feature-libs/order/core/facade/order-attachments.service.ts	Allegati, direttamente sul connector (:L16-L45).
feature-libs/order/core/facade/my-account-v2-order-history.service.ts	Aggregazione lista + dettagli + tracking + resi (:L36-L270).
feature-libs/order/core/store/order-state.ts	OrderState, process id, entity type, key builder (:L17-L58).
feature-libs/order/core/store/order-store.module.ts	StoreModule.forFeature / EffectsModule.forFeature (:L14-L21).
feature-libs/order/core/store/reducers/index.ts	getReducers(), reducerToken, reducerProvider (:L39-L85).
feature-libs/order/core/store/effects/index.ts	Array effects (8 classi) (:L16-L25).
feature-libs/order/core/connectors/*.adapter.ts / *.connector.ts	6 adapter astratti + 6 connector pass-through.
feature-libs/order/occ/order-occ.module.ts	Binding adapter → Occ*, 5 converter multi (:L38-L90).
feature-libs/order/occ/config/default-occ-order-config.ts	17 endpoint OCC di default (:L9-L50).
feature-libs/order/occ/adapters/occ-order.adapter.ts	POST place order con backOff (:L38-L115).
feature-libs/order/occ/adapters/occ-order-history.adapter.ts	Dettaglio, lista, tracking, cancel, resi (:L57-L231).
feature-libs/order/occ/adapters/converters/occ-order-normalizer.ts	OccOrderNormalizer (:L25-L81).
feature-libs/order/components/order-components.module.ts	Aggregatore dei 10 moduli componenti (:L23-L43).
feature-libs/order/components/order-details/order-details.service.ts	OrderDetailsService: codice dal router → load → select (:L19-L56).
feature-libs/order/components/order-details/order-details.module.ts	Mapping CMS dettaglio, layout dialog, outlet v2 (:L61-L190).
feature-libs/order/components/order-history/order-history.component.ts	Lista ordini paginata (:L58-L156).
feature-libs/order/components/amend-order/amend-order.service.ts	OrderAmendService astratto (form cancel/reso) (:L23-L153).
feature-libs/order/components/amend-order/cancellations/order-cancellation.service.ts	Cancellazione (:L41-L93).
feature-libs/order/components/amend-order/returns/order-return.service.ts	Reso (:L54-L99).
feature-libs/order/components/guards/order-confirmation.guard.ts	OrderConfirmationGuard (:L24-L36).
feature-libs/order/components/order-confirmation/order-confirmation.module.ts	Mapping CMS conferma + outlet (:L90-L141).

PATH	RUOLO
feature-libs/order/components/return-request-detail/return-request.service.ts	Dettaglio/cancel reso (:L35-L94).
feature-libs/order/document-flow/**	Feature orderDocumentFlow (root/core/occ/components).
feature-libs/pdf-invoices/**	Feature pdfInvoices (root/core/occ/components).
projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts	App demo: module: () => import(...) e flag v2 (:L23-L37).

File aggiuntivi

PATH	RUOLO
feature-libs/order/schematics/add-order/index.ts	addOrderFeatures(): factory degli schematics ng-add/add di @spartacus/order (vedi 3.10).
feature-libs/order/schematics/add-order/index_spec.ts	Test dello schematic: lazy loading, nessuna installazione delle dipendenze, scss, angular.json, eager loading.
feature-libs/order/styles/_index.scss	Entry SCSS degli stili order: importa components/index.
feature-libs/order/styles/components/_index.scss	Importa i cinque gruppi: order-history, order-overview, order-details, replenishment-order, my-account-v2.
feature-libs/order/styles/components/my-account-v2/_index.scss	Allowlist $myaccount-view-order-allowlist (solo cx-my-account-v2-orders) + @extend dei placeholder.
feature-libs/order/styles/components/my-account-v2/_my-account-v2-order.scss	Placeholder %cx-my-account-v2-orders (widget ordini My Account v2).
feature-libs/order/styles/components/order-details/_index.scss	Importa gli stili del dettaglio e definisce $order-details-allowlist (include anche i selettori del document flow).
feature-libs/order/styles/components/order-details/_attachments-dialog.scss	Placeholder %cx-order-attachments-dialog.
feature-libs/order/styles/components/order-details/_my-account-v2-order-details.scss	Placeholder v2: %cx-my-account-v2-order-details-actions, %cx-my-account-v2-consignment-tracking, %cx-my-account-v2-download-invoices.
feature-libs/order/styles/components/order-details/_order-attachments.scss	Placeholder %cx-order-attachments.
feature-libs/order/styles/components/order-details/_order-detail-billing.scss	Placeholder %cx-order-detail-billing.
feature-libs/order/styles/components/order-details/_order-details-actions.scss	Placeholder %cx-order-details-actions.
feature-libs/order/styles/components/order-details/_order-details-items.scss	Placeholder %cx-order-details-items.
feature-libs/order/styles/components/order-details/_order-details-reorder.scss	Placeholder %cx-order-details-reorder.
feature-libs/order/styles/components/order-details/_order-details-shipping.scss	Definisce di nuovo %cx-order-overview (stesso placeholder di _order-overview.scss, vedi 3.10).
feature-libs/order/styles/components/order-details/_reorder-dialog.scss	Selettore diretto cx-reorder-dialog (non placeholder, fuori allowlist).
feature-libs/order/styles/components/order-history/_index.scss	Importa cart-item-list da cart base + gli stili storico/amend/resi/tracking; $order-history-allowlist.

PATH	RUOLO
feature-libs/order/styles/components/order-history/_amend-order-actions.scss	Placeholder %cx-amend-order-actions.
feature-libs/order/styles/components/order-history/_amend-order-items.scss	Placeholder %cx-amend-order-items.
feature-libs/order/styles/components/order-history/_cancel-order.scss	Placeholder %cx-cancel-order.
feature-libs/order/styles/components/order-history/_consignment-tracking.scss	Placeholder %cx-consignment-tracking e %cx-tracking-events.
feature-libs/order/styles/components/order-history/_my-account-v2-order-history.scss	Mixin draw-line + placeholder %cx-my-account-v2-order-history.
feature-libs/order/styles/components/order-history/_order-history.scss	Placeholder %cx-order-history.
feature-libs/order/styles/components/order-history/_order-return-request-list.scss	Placeholder %cx-order-return-request-list.
feature-libs/order/styles/components/order-history/_replenishment-order-history.scss	Placeholder %cx-replenishment-order-history.
feature-libs/order/styles/components/order-history/_return-request-items.scss	Placeholder %cx-return-request-items.
feature-libs/order/styles/components/order-history/_return-request-overview.scss	Placeholder %cx-return-request-overview.
feature-libs/order/styles/components/order-history/_return-request-totals.scss	Placeholder %cx-return-request-totals.
feature-libs/order/styles/components/order-overview/_index.scss	Allowlist $order-overview-allowlist (cx-order-overview).
feature-libs/order/styles/components/order-overview/_order-overview.scss	Placeholder %cx-order-overview.
feature-libs/order/styles/components/replenishment-order/_index.scss	Allowlist $replenishment-order-allowlist (cancellation + dialog).
feature-libs/order/styles/components/replenishment-order/_replenishment-order-cancellation.scss	Placeholder %cx-replenishment-order-cancellation e %cx-replenishment-order-cancellation-dialog.
feature-libs/pdf-invoices/_index.scss	Entry SCSS di @spartacus/pdf-invoices: importa @spartacus/styles/scss/core e styles/index.
feature-libs/pdf-invoices/styles/_index.scss	Importa _pdf-invoices.scss.
feature-libs/pdf-invoices/styles/_pdf-invoices.scss	Stili di cx-invoices-list con selettore diretto (nessuna allowlist).
feature-libs/pdf-invoices/assets/public_api.ts	Entry @spartacus/pdf-invoices/assets: riesporta translations/translations.
feature-libs/pdf-invoices/pdf-invoices.module.ts	PDFInvoicesModule = components + core + occ (:L12-L19).
feature-libs/pdf-invoices/public_api.ts	Entry @spartacus/pdf-invoices: riesporta solo il modulo (:L10-L10).
feature-libs/pdf-invoices/schematics/add-pdf-invoices/index.ts	addPDFInvoicesFeature(): factory degli schematics di pdf-invoices (vedi 3.10).

PATH	RUOLO
feature-libs/pdf-invoices/schematics/add-pdf-invoices/index_spec.ts	Test dello schematic pdf-invoices (lazy, scss, angular.json, eager).
feature-libs/pdf-invoices/vitest.config.ts	Config Vitest della libreria: jsdom, pool forks, coverage v8 con soglie 85/65, report junit.

3. Come funziona passo per passo

3.1 Registrazione eager e caricamento lazy

1. **Root eager.** `OrderRootModule` fornisce tre default config: componenti (factory), routing e `defaultOrderConfig` (feature-libs/order/root/order-root.module.ts:L174-L178). La factory dichiara la feature `order` con i 36 tipi CMS di `ORDER_CMS_COMPONENTS` e la dipendenza da `CART_BASE_FEATURE`, e mappa `orderCore` come **alias stringa** di `order` (feature-libs/order/root/order-root.module.ts:L67-L80).
2. **Il `module` lo mette l'app.** Una feature è "configurata" solo se ha la proprietà `module` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35); il root module non la fornisce. Nell'app demo `OrderFeatureModule` aggiunge `module: () => import('./order-wrapper.module')` (projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L26-L33), e `OrderWrapperModule` importa `OrderModule` più estensioni condizionali (S/4 service, OMF, OPF) (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27).
3. **Facade proxy.** Ogni facade è una classe astratta `providedIn: 'root'` con `useFactory → facadeFactory({ feature: ORDER_CORE_FEATURE, methods: [...] })` (feature-libs/order/root/facade/order.facade.ts:L14-L30). Il proxy è un'istanza di una sottoclasse anonima a cui vengono aggiunti **solo** i metodi elencati (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L117-L127).
4. **Prima chiamata.** `resolveFeature('orderCore')` risolve l'alias con un ciclo `while` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100), risolve le dipendenze (`cartBase`) e importa il modulo (core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73). Il proxy prende poi la facade dall'injector del modulo lazy (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52), dove `facadeProviders` la lega al servizio concreto con `useExisting` (feature-libs/order/core/facade/facade-providers.ts:L26-L63).
5. **`async: true`.** `OrderHistoryFacade`, `OrderReturnRequestFacade` e `ReplenishmentOrderHistoryFacade` usano `async: true` (feature-libs/order/root/facade/order-history.facade.ts:L41-L41, feature-libs/order/root/facade/order-return-request.facade.ts:L37-L37, feature-libs/order/root/facade/replenishment-order-history.facade.ts:L39-L39), che aggiunge `delay(0)` alla risoluzione (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L53-L56).
6. **Metodi `void` via proxy.** `call()` collega subito un `connectable` con `ReplaySubject`: il metodo reale viene eseguito appena il modulo è risolto anche se il chiamante non si sottoscrive (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L92). Per questo `loadOrderDetails(code)` chiamato sul proxy produce davvero il dispatch.
7. **Core.** `OrderCoreModule` importa `OrderStoreModule` e fornisce i `facadeProviders` più 6 connector (feature-libs/order/core/order-core.module.ts:L17-L29). Lo store è registrato con `StoreModule.forFeature(ORDER_FEATURE, reducerToken)` e `EffectsModule.forFeature(effects)` (feature-libs/order/core/store/order-store.module.ts:L14-L21).
8. **OCC.** `OrderOccModule` fornisce `defaultOccOrderConfig`, lega 6 adapter astratti alle implementazioni `Occ*` e registra 5 converter multi (feature-libs/order/occ/order-occ.module.ts:L38-L90).

3.2 Rotte e contesto

- Rotte di default: `orders` → `my-account/orders`, `orderDetails` → `my-account/order/:orderCode`, `orderGuest` → `guest/order/:orderCode`, cancel/return e relative conferme sotto `my-account/order/...`, `returnRequestDetails` → `my-account/return-request/:returnCode`, replenishment sotto `my-account/my-replenishment(s)`, `replenishmentConfirmation` → `replenishment/confirmation` (nessuna rotta `path: null` in `OrderRootModule`, feature-libs/order/root/order-root.module.ts:L82-L172), `orderConfirmation` → `order-confirmation` (feature-libs/order/root/config/default-order-routing-config.ts:L9-L54).
- `paramsMapping` collega i parametri URL alle proprietà del modello: `orderCode` → `code`, `returnCode` → `rma`, `replenishmentOrderCode` → `replenishmentOrderCode` (feature-libs/order/root/config/default-order-routing-config.ts:L17-L17, :L41-L41, :L48-L48). Per questo `OrderHistoryComponent.goToOrderDetail` può passare l'intero ordine come `params` (feature-libs/order/components/order-history/order-history.component.ts:L119-L129).
- Le rotte `orderGuest`, `orderDetails`, `orders`, `replenishment*`, `returnRequestDetails` hanno `[AuthGuard, CmsPageGuard]`; cancel, return, le loro conferme e `orderConfirmation` hanno solo `CmsPageGuard` (feature-libs/order/root/order-root.module.ts:L104-L131, :L160-L171). Per cancel/return l'`AuthGuard` è nel mapping CMS (feature-libs/order/components/amend-order/cancellations/cancel-order/cancel-order.module.ts:L40-L42).
- `orderDetails` e `orderConfirmation` impostano `cxContext[ORDER_ENTRIES_CONTEXT]` sui token `OrderDetailsOrderEntriesContextToken` / `OrderConfirmationOrderEntriesContextToken` (feature-libs/order/root/order-root.module.ts:L97-L102, :L165-L170). I token sono legati ai contesti concreti rispettivamente in `OrderComponentsModule` e `OrderConfirmationModule` (feature-libs/order/components/order-components.module.ts:L23-L43, feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L132-L135); il primo legge le entry da `OrderHistoryFacade`, il secondo da `OrderFacade` (feature-libs/order/components/page-context/order-details-order-entries.context.ts:L20-L29, feature-libs/order/components/page-context/order-confirmation-order-entries.context.ts:L20-L31).

3.3 Place order e pagina di conferma

1. Il checkout chiama `orderFacade.placeOrder(this.checkoutSubmitForm.valid)` e su `next` naviga a `cxRoute: 'orderConfirmation'` (feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L149-L170).
2. `OrderService.placeOrder` esegue `placeOrderCommand`, creato con `CommandStrategy.CancelPrevious` (feature-libs/order/core/facade/order.service.ts:L26-L54, :L119-L121).
3. `checkoutPreconditions()` prende con `take(1)` userId, id del carrello attivo e flag guest, e lancia `Error('Checkout conditions not met')` se manca uno dei due id o se l'utente è anonimo con carrello non guest (feature-libs/order/core/facade/order.service.ts:L99-L117).
4. `OccOrderAdapter.placeOrder` fa POST con body vuoto, `cartId`/`termsChecked` in query string, `Content-Type: application/x-www-form-urlencoded`, header `USE_CLIENT_TOKEN` per l'utente anonimo (guest) e `backOff({ shouldRetry: isJaloError })`; la risposta passa per `ORDER_NORMALIZER` (feature-libs/order/occ/adapters/occ-order.adapter.ts:L38-L66, :L106-L115).
5. `OccOrderNormalizer` converte entries, entry delle consignment e unconsigned entries con `PRODUCT_NORMALIZER` e `ORDER_ENTRY_PROMOTIONS_NORMALIZER`, aggiungendo `orderCode` (feature-libs/order/occ/adapters/converters/occ-order-normalizer.ts:L25-L81).
6. Al successo: `placedOrder$.next(order)` e dispatch di `OrderPlacedEvent` con `cartCode: cartId` (feature-libs/order/core/facade/order.service.ts:L32-L46). Il JSDoc dell'evento spiega che `cartId` è `code` per utenti loggati e `guid` per anonimi (feature-libs/order/root/events/order.events.ts:L16-L29).

7. `CheckoutPlaceOrderEventListener` reagisce all'evento dispatchando `RemoveCartEvent` e `CheckoutQueryResetEvent` (feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L24-L40).
8. Sulla pagina `/order-confirmation` ogni componente CMS di conferma è protetto da `OrderConfirmationGuard`, che lascia passare solo se `orderFacade.getOrderDetails()` emette un oggetto non vuoto, altrimenti reindirizza a `orders` (feature-libs/order/components/guards/order-confirmation.guard.ts:L24-L36).
9. Overview e Billing della conferma riusano `OrderOverviewComponent` / `OrderDetailBillingComponent` con `OrderDetailsService` sostituito da `OrderFacade` (feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L96-L129). Poiché `OrderFacade` non ha `isOrderDetailsLoading`, i componenti verificano `typeof ... === 'function'` e altrimenti usano `of(false)` (feature-libs/order/components/order-details/order-overview/order-overview.component.ts:L66-L70, feature-libs/order/components/order-details/order-detail-items/order-detail-items.component.ts:L78-L81).
10. `OrderConfirmationModule` aggiunge `OrderConfirmationShippingComponent` all'outlet `CONSIGNMENT_DELIVERY_INFO` (feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L136-L139).
11. Il thank-you message aggiunge un global message `MSG_TYPE_ASSISTIVE` con il codice ordine (o `replenishmentOrderCode`) per gli screen reader (feature-libs/order/components/order-confirmation/order-confirmation-thank-you-message/order-confirmation-thank-you-message.component.ts:L65-L93).
12. In `ngOnDestroy` thank-you message, items e totals chiamano `orderFacade.clearPlacedOrder()` (feature-libs/order/components/order-confirmation/order-confirmation-thank-you-message/order-confirmation-thank-you-message.component.ts:L69-L71, feature-libs/order/components/order-confirmation/order-confirmation-items/order-confirmation-items.component.ts:L41-L43, feature-libs/order/components/order-confirmation/order-confirmation-totals/order-confirmation-totals.component.ts:L26-L28).

Varianti:

- `placePaymentAuthorizedOrder(termsChecked, cartId?)`: con `cartId` esplicito salta il command e le precondizioni sul carrello attivo, controlla solo lo userId e chiama direttamente il connector (feature-libs/order/core/facade/order.service.ts:L123-L156). L'adapter aggiunge un secondo `backOff` su `isServerError` con `maxTries: DEFAULT_SERVER_ERROR_RETRIES_COUNT` (feature-libs/order/occ/adapters/occ-order.adapter.ts:L91-L103).
- **Guest:** `OrderGuestRegisterFormComponent` (host `ngSkipHydration`) registra l'ospite con `userRegisterFacade.registerGuest(guid, password)` (feature-libs/order/components/order-confirmation/order-guest-register-form/order-guest-register-form.component.ts:L30-L91).

3.4 Dettaglio ordine (My Account)

1. Rotta `my-account/order/:orderCode` con `[AuthGuard, CmsPageGuard]` (feature-libs/order/root/order-root.module.ts:L92-L103).
2. I componenti CMS di dettaglio (8 tipi, tutti con `guards: [AuthGuard]`) leggono da `OrderDetailsService` (feature-libs/order/components/order-details/order-details.module.ts:L128-L172).
3. `OrderDetailsService` (`providedIn: 'root'`) deriva `orderCode$` dal param del router con `distinctUntilChanged`; `orderLoad$` chiama `loadOrderDetails(orderCode)` o `clearOrderDetails()` ed è `shareReplay({ bufferSize: 1, refCount: true })` (feature-libs/order/components/order-details/order-details.service.ts:L30-L44).
4. `getOrderDetails()` fa `switchMap` da `orderLoad$` a `orderHistoryFacade.getOrderDetails()`: il caricamento parte solo quando qualcuno si sottoscrive (feature-libs/order/components/order-details/order-

details.service.ts:L51-L55).

5. `OrderHistoryService.loadOrderDetails` prende lo userId e dispatcha `LoadOrderDetails({ userId, orderCode })` (feature-libs/order/core/facade/order-history.service.ts:L50-L59).
6. `OrderDetailsEffect.loadOrderDetails$` chiama `OrderHistoryConnector.get` e mappa in `LoadOrderDetailsSuccess/Fail` con `tryNormalizeHttpError` (feature-libs/order/core/store/effects/order-details.effect.ts:L35-L57).
7. `OccOrderHistoryAdapter.load` fa GET su `orderDetail` (`?fields=FULL`), con `USE_CLIENT_TOKEN` se l'utente è anonimo, e normalizza con `ORDER_NORMALIZER` (feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L57-L68). Con `showOrderQuoteLink` attivo l'URL è la fusione degli scope `orderDetail` e `quoteCode` tramite `OccFieldsService.getOptimalUrlGroups` (feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L212-L231).
8. Il success finisce nella slice `orderDetail` (`loaderReducer(ORDER_DETAILS, ...)`) (feature-libs/order/core/store/reducers/index.ts:L45-L48) e il selettore `getOrderDetails` lo restituisce (feature-libs/order/core/store/selectors/order-details.selectors.ts:L18-L21).
9. Al cambio di lingua o valuta `resetOrderDetails$` ricarica l'ordine corrente (codice letto dallo store con `withLatestFrom`), altrimenti `EMPTY` (feature-libs/order/core/store/effects/order-details.effect.ts:L87-L120).

Sopra questo flusso:

- **Azioni.** Il template mostra i link `orderCancel` se `order.cancellable` e `orderReturn` se `order.returnable` (feature-libs/order/components/order-details/order-detail-actions/order-detail-actions.component.html:L13-L39).
- **Consegne.** `OrderDetailItemsComponent` raggruppa consegne in pickup/delivery tramite `MyAccountV2OrderConsignmentsService` (feature-libs/order/components/order-details/order-detail-items/order-detail-items.component.ts:L61-L72), che ordina per stato: in lavorazione, completate (`DELIVERY_COMPLETED`, `PICKUP_COMPLETE`), cancellate (feature-libs/order/components/order-details/my-account-v2-order-consignments.service.ts:L50-L81, feature-libs/order/components/order-details/order-detail-items/order-consigned-entries/order-consigned-entries.model.ts:L7-L8).
- **Tracking.** `ConsignmentTrackingComponent.openTrackingDialog` chiama `loadConsignmentTracking(orderCode, consignment.code)` e apre il dialog `CONSIGNMENT_TRACKING` con `tracking$` e `shipDate` (feature-libs/order/components/order-details/order-detail-items/consignment-tracking/consignment-tracking.component.ts:L61-L87); il pulsante esiste per gli stati SHIPPED, IN_TRANSIT, DELIVERY_COMPLETED, DELIVERY_REJECTED, DELIVERING (feature-libs/order/components/order-details/order-detail-items/consignment-tracking/consignment-tracking.component.ts:L35-L41). Il template mostra il tracking dietro `*cxFeature="'consignmentTracking'"` (feature-libs/order/components/order-details/order-detail-items/order-consigned-entries/order-consigned-entries.component.html:L19-L19), con feature level `'1.2'` impostato dallo stesso modulo (feature-libs/order/components/order-details/order-details.module.ts:L173-L175).
- **Link al preventivo.** `OrderOverviewComponent` mostra il link a `quoteDetails` solo se `orderConfig.showOrderQuoteLink && quoteCode` (feature-libs/order/components/order-details/order-overview/order-overview.component.html:L178-L195).

3.5 Lista ordini

1. `OrderHistoryComponent.orders$ = getOrderHistoryList(5)` (feature-libs/order/components/order-history/order-history.component.ts:L61-L71).
2. `getOrderHistoryList` implementa "load on subscribe": se lo stato loader non è né loading né success né error chiama `loadOrderList(pageSize)`, poi mappa `value` (feature-libs/order/core/facade/order-

history.service.ts:L71-L87).

3. `loadOrderList` richiede utente loggato (`takeUserId(true)`), legge in modo sincrono `replenishmentOrderCode` dai params del router e dispatcha `LoadUserOrders`; l'errore è inghiottito con un TODO (feature-libs/order/core/facade/order-history.service.ts:L102-L130). `takeUserId(true)` lancia per l'utente anonimo (core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L57-L69).
4. `OrdersEffect.loadUserOrders$` sceglie la sorgente: con `replenishmentOrderCode` usa `loadReplenishmentDetailsHistory`, altrimenti `getHistory` (feature-libs/order/core/store/effects/orders.effect.ts:L33-L68).
5. `OccOrderHistoryAdapter.loadHistory` aggiunge `pageSize`, `currentPage`, `sort` in query solo se truthy (feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L70-L95).
6. Cambio sort/pagina → `loadOrderList(PAGE_SIZE, currentPage, sortCode)`; `ngOnDestroy` → `clearOrderList()` (feature-libs/order/components/order-history/order-history.component.ts:L98-L155).
7. `resetUserOrders$` svuota la lista al cambio lingua (feature-libs/order/core/store/effects/orders.effect.ts:L70-L78).

3.6 Cancellazione e reso (amend order)

1. `OrderAmendService` è astratto: `getEntries()` e `save()` sono astratti; il form viene costruito per ordine e ricostruito se cambia `orderCode` (feature-libs/order/components/amend-order/amend-order.service.ts:L34-L84).
2. Il form ha un controllo per entry (chiave `entryNumber`) con `Validators.min(0)` e `max(getMaxAmendQuantity(entry))`; il gruppo `entries` ha il validatore `ValidateQuantityToCancel` (somma > 0, errore `cxNoSelectedItemToCancel`) (feature-libs/order/components/amend-order/amend-order.service.ts:L23-L32, :L86-L108). Il massimo è `cancellableQuantity` o `returnableQuantity` con fallback su `quantity` (feature-libs/order/components/amend-order/amend-order.service.ts:L140-L148).
3. Il servizio concreto entra tramite i `providers` del mapping CMS: `{ provide: OrderAmendService, useExisting: OrderCancellationService }` (feature-libs/order/components/amend-order/cancellations/cancel-order/cancel-order.module.ts:L38-L51).
4. `AmendOrderActionsComponent.continue` naviga a `forwardRoute` con `params: { code: orderCode }` se il form è valido, altrimenti `markAllAsTouched` (feature-libs/order/components/amend-order/amend-order-actions/amend-order-actions.component.ts:L35-L45).
5. Le pagine di conferma hanno guard `[AuthGuard, OrderCancellationGuard]` (o `OrderReturnGuard`); la guard reindirizza a `orders` se il form non è valido (feature-libs/order/components/amend-order/cancellations/cancel-order-confirmation/cancel-order-confirmation.module.ts:L29-L31, feature-libs/order/components/amend-order/cancellations/order-cancellation.guard.ts:L24-L38).
6. **Cancel.** `OrderCancellationService.getEntries` esclude `entryNumber === -1` e le entry senza `cancellableQuantity > 0` (feature-libs/order/components/amend-order/cancellations/order-cancellation.service.ts:L41-L54). `save()` costruisce gli input, resetta il form, chiama `cancelOrder`, attende `getCancelOrderSuccess()` con `first(Boolean)`, poi resetta il process state, mostra `cancelSuccess` e naviga a `orders` (feature-libs/order/components/amend-order/cancellations/order-cancellation.service.ts:L56-L93).
7. Il flag di successo viene dal process store: `ProcessSelectors.getProcessSuccessFactory(CANCEL_ORDER_PROCESS_ID)` (feature-libs/order/core/facade/order-history.service.ts:L191-L204); l'azione `CancelOrder` estende `EntityLoadAction` su `PROCESS_FEATURE` (feature-libs/order/core/store/actions/order-details.action.ts:L60-L95).
8. `cancelOrder$` chiama il connector (POST JSON su `cancelOrder`) e in errore aggiunge un global message per ciascun `error.error.errors[].message` (feature-libs/order/core/store/effects/order-details.

effect.ts:L59-L85, feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L110-L127).

9. **Reso.** `ReturnOrderComponent` tiene solo le entry con una consignment entry dello stesso codice prodotto e usa `shippedQuantity` come nuovo `returnableQuantity` solo nelle entry passate alla UI (feature-libs/order/components/amend-order/returns/return-order/return-order.component.ts:L42-L73); il `Validators.max` del form resta sul valore originale (feature-libs/order/components/amend-order/amend-order.service.ts:L96-L106). `OrderReturnService.save` chiama `createOrderReturnRequest`, attende `getReturnRequestSuccess()`, legge il `ReturnRequest` creato e naviga a `returnRequestDetails` con `params: { rma }` (feature-libs/order/components/amend-order/returns/order-return.service.ts:L54-L99).
10. `createReturnRequest` serializza con `ORDER_RETURN_REQUEST_INPUT_SERIALIZER` e normalizza con `ORDER_RETURN_REQUEST_NORMALIZER` (feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L129-L151). Create e load del reso condividono l'entity type `RETURN_REQUEST_DETAILS`, quindi scrivono nella stessa slice `orderReturn` (feature-libs/order/core/store/actions/order-return-request.action.ts:L55-L117).
11. **Dettaglio reso.** `ReturnRequestService.getReturnRequest` legge `returnCode` dal router e carica il dettaglio se assente o con `rma` diverso e non in loading (feature-libs/order/components/return-request-detail/return-request.service.ts:L35-L59). La cancellazione invia `{ status: 'CANCELLING' }` (PATCH su `cancelReturn`) (feature-libs/order/components/return-request-detail/return-request.service.ts:L65-L69, feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L193-L210); `backToList()` torna a `orders` con `state: { activeTab: 1 }` (feature-libs/order/components/return-request-detail/return-request.service.ts:L85-L94).

3.7 Reorder

1. `OrderDetailReorderComponent` disabilita il pulsante se nessuna entry ha un prodotto ancora in catalogo e apre il dialog `REORDER` con `{ orderCode }` (feature-libs/order/components/order-details/order-detail-reorder/order-detail-reorder.component.ts:L46-L74).
2. `ReorderDialogComponent.createCartFromOrder` chiama `reorderOrderFacade.reorder`, poi `multiCartFacade.reloadCart(OCC_CART_ID_CURRENT, { active: true })` e mostra le `cartModifications`; in errore aggiunge `reorder.dialog.error` e chiude (feature-libs/order/components/order-details/order-detail-reorder/reorder-dialog/reorder-dialog.component.ts:L77-L96).
3. `ReorderOrderService` usa un command `CancelPrevious`: prima `reorderPreconditions()`, poi `ReorderOrderConnector.reorder(orderId, userId)` (feature-libs/order/core/facade/reorder-order.service.ts:L26-L37).
4. Le precondizioni falliscono con `'Must be logged in to reorder'` se manca lo userId; se esiste un carrello attivo lo cancellano e attendono che l'entità non sia in loading e sia assente o in errore (feature-libs/order/core/facade/reorder-order.service.ts:L56-L92).
5. `OccReorderOrderAdapter` fa POST JSON con body vuoto su `reorder` con `orderCode` in query (feature-libs/order/occ/adapters/occ-reorder-order.adapter.ts:L31-L51). L'endpoint non ha default nella libreria order: esiste in `defaultB2bOccConfig` (feature-libs/order/occ/model/occ-order-endpoints.model.ts:L86-L93, core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L31-L36).

3.8 Replenishment

1. **Schedulazione** (dal checkout scheduled replenishment): `ScheduledReplenishmentOrderService` usa un command `CancelPrevious`, chiama il connector con `(cartId, form, termsChecked, userId)` e al successo fa `orderFacade.setPlacedOrder(replenishmentOrder)` e dispatcha `ReplenishmentOrderScheduledEvent` (feature-libs/order/core/facade/scheduled-replenishment-order.service.ts:L32-L68). Le precondizioni sono quelle del place order con messaggio `'Order conditions not met'` (feature-libs/order/core/facade/scheduled-replenishment-order.service.ts:L79-L97).

2. L'adapter serializza il form con `REPLENISHMENT_ORDER_FORM_SERIALIZER` (`feature-libs/order/occ/adapters/occ-scheduled-replenishment-order.adapter.ts:L39-L82`); il serializer trasforma `replenishmentStartDate` `YYYY-mm-dd` aggiungendo ora locale, `:00` e offset del fuso (`feature-libs/order/occ/adapters/converters/occ-scheduled-replenishment-order-form-serializer.ts:L34-L46`).
3. **Storico e dettaglio** passano per NgRx: `ReplenishmentOrderHistoryService` usa `takeUserId(true)` per caricare dettagli, lista e cancellare (`feature-libs/order/core/facade/replenishment-order-history.service.ts:L43-L57`). `ReplenishmentOrderDetailsService` replica il pattern di `OrderDetailsService` sul param `replenishmentOrderCode` (`feature-libs/order/components/replenishment-order-details/replenishment-order-details.service.ts:L25-L55`).
4. **Ordini generati.** Nella pagina di dettaglio replenishment, `ReplenishmentDetailOrderHistoryComponent` è mappato su `OrderHistoryComponent` (`feature-libs/order/components/replenishment-order-details/replenishment-order-details.module.ts:L51-L87`); `loadOrderList` trova `replenishmentOrderCode` nei params e `OrdersEffect` usa l'endpoint `replenishmentOrderDetailsHistory` (`feature-libs/order/core/facade/order-history.service.ts:L107-L123`, `feature-libs/order/core/store/effects/orders.effect.ts:L39-L47`).
5. **Cancellazione.** `ReplenishmentOrderHistoryComponent.openDialog` apre `REPLENISHMENT_ORDER` (`feature-libs/order/components/replenishment-order-history/replenishment-order-history.component.ts:L132-L144`); l'adapter fa PATCH con body `{}` (`feature-libs/order/occ/adapters/occ-replenishment-order-history.adapter.ts:L43-L87`); `CANCEL_REPLENISHMENT_ORDER_SUCCESS` aggiorna sia la lista (sostituendo l'elemento) sia il dettaglio (`feature-libs/order/core/store/reducers/replenishment-orders.reducer.ts:L30-L49`, `feature-libs/order/core/store/reducers/replenishment-order-details.reducer.ts:L16-L20`).

3.9 Allegati, My Account v2, document flow, PDF invoices

- **Allegati.** `OrderAttachmentsService` chiama il connector dopo `takeUserId().pipe(take(1))`, senza store (`feature-libs/order/core/facade/order-attachments.service.ts:L16-L45`). Il dialog carica `sapAttachments` (fallback `[]`) (`feature-libs/order/components/order-details/order-attachments/attachments-dialog/order-attachments-dialog.component.ts:L72-L90`); al click scarica il blob (`responseType: 'blob' as 'json'`, `feature-libs/order/occ/adapters/occ-order-attachments.adapter.ts:L31-L65`) e, se il MIME è in `previewMimeTypes`, lo apre in una nuova scheda, altrimenti forza il download (`feature-libs/order/components/order-details/order-attachments/attachments-dialog/order-attachments-dialog.component.ts:L95-L145`).
- **My Account v2.** Il token `USE_MY_ACCOUNT_V2_ORDER` (default `false`, `feature-libs/order/root/tokens/context.ts:L17-L20`) attiva tre cambiamenti: mapping di `AccountOrderDetailsActionsComponent` → `MyAccountV2OrderDetailsActionsComponent` (`feature-libs/order/components/order-details/order-details.module.ts:L80-L87`, `:L180-L182`), un `MODULE_INITIALIZER` che mette `MyAccountV2ConsignmentTrackingComponent` in `OrderOutlets.ORDER_CONSIGNMENT` con `OutletPosition.REPLACE` (`feature-libs/order/components/order-details/order-details.module.ts:L61-L78`, `:L183-L187`), e `AccountOrderHistoryComponent` → `MyAccountV2OrderHistoryComponent` (`feature-libs/order/components/order-history/order-history.module.ts:L34-L41`, `:L65-L77`). `MyAccountV2OrderHistoryService` arricchisce ogni ordine della lista con dettaglio completo (`LoadOrderById`, slice entity `orderById`), tracking delle consegne con `trackingID` e resi associati (`feature-libs/order/core/facade/my-account-v2-order-history.service.ts:L46-L174`, `:L198-L213`).
- **Download fatture v2.** È event-driven: `MyAccountV2OrderDetailsActionsComponent.showDialog` dispatcha `DownloadOrderInvoicesEvent`; `MyAccountV2DownloadInvoicesEventListener` apre il dialog `DOWNLOAD_ORDER_INVOICES` (`feature-libs/order/components/order-details/my-account-v2/order-details-actions/my-account-v2-order-details-actions.component.ts:L31-L35`, `feature-libs/order/components/order-details/my-account-v2/download-invoices/my-account-v2-download-invoices-event.listener.`

ts:L24-L41), che incorpora `InvoicesListComponent` di `@spartacus/pdf-invoices` (feature-libs/order/components/order-details/my-account-v2/download-invoices/my-account-v2-download-invoices.component.ts:L41-L65).

- **Document flow.** Feature separata `orderDocumentFlow` con un solo tipo CMS, `AccountOrderDocumentFlowComponent`, senza `dependencies` (feature-libs/order/document-flow/root/order-document-flow-root.module.ts:L11-L22). `OrderDocumentFlowFacade` espone `getOrderSubsequentDocuments` e `getOrderSubsequentDocumentEntries` (feature-libs/order/document-flow/root/facade/order-document-flow.facade.ts:L17-L45); il componente riusa `OrderDetailsService` e apre il dialog `ORDER_DOCUMENT_FLOW` (feature-libs/order/document-flow/components/order-document-flow/order-document-flow.component.ts:L29-L46), che mette in cache le entry per documento (feature-libs/order/document-flow/components/order-document-flow-dialog/order-document-flow-dialog.component.ts:L111-L177). Gli endpoint sono `subsequentDocuments` e `subsequentDocumentsEntries` (feature-libs/order/document-flow/occ/config/default-occ-document-flow-config-factory.ts:L10-L24).
- **PDF invoices.** Feature `pdfInvoices` con alias `pdfInvoicesCore` e CMS `AccountOrderDetailsPDFInvoicesComponent` (feature-libs/pdf-invoices/root/pdf-invoices-root.module.ts:L14-L33). `PDFInvoicesFacade` è lazy sulla feature `pdfInvoices` (feature-libs/pdf-invoices/root/facade/pdf-invoices.facade.ts:L16-L22). `PDFInvoicesService` memorizza `userId` e `orderId` (dal param `orderCode`) nel costruttore e li usa come fallback (feature-libs/pdf-invoices/core/services/pdf-invoices.service.ts:L25-L76). `PDFInvoicesBadRequestHandler` (400) traduce gli `UnknownIdentifierError` su `Order`/`Invoice` in messaggi dedicati (feature-libs/pdf-invoices/core/http-interceptors/bad-request/pdf-invoices-badrequest.handler.ts:L20-L90). `InvoicesListComponent` usa page size 5 e forza `fields: DEFAULT` sugli aggiornamenti successivi (feature-libs/pdf-invoices/components/invoices-list/invoices-list.component.ts:L67-L155); il download passa per `FileDownloadService.download(url, '<invoiceId>.pdf')` (feature-libs/pdf-invoices/components/invoices-list/invoices-list.component.ts:L171-L182).

3.10 Installazione via schematics e stili

- **Schematics.** `addOrderFeatures()` legge il `package.json` dell'app, valida l'installazione Spartacus, risolve le dipendenze tra feature con `analyzeCrossFeatureDependencies` e concatena `analyzeApplication`, `addFeatures`, `addPackageJsonDependenciesForLibrary(peerDependencies, ...)` e `finalizeInstallation` (feature-libs/order/schematics/add-order/index.ts:L25-L43). La stessa funzione è registrata sia come `ng-add` (privato, nascosto, alias `install`) sia come `add` (feature-libs/order/schematics/collection.json:L4-L16). Il test verifica che i moduli feature di cart base e user **non** vengano generati (feature-libs/order/schematics/add-order/index_spec.ts:L99-L107). `addPDFInvoicesFeature()` ha corpo identico (feature-libs/pdf-invoices/schematics/add-pdf-invoices/index.ts:L25-L45), ma il suo spec non contiene il test "should NOT install the required feature dependencies" (feature-libs/pdf-invoices/schematics/add-pdf-invoices/index_spec.ts:L79-L123).
- **Stili.** Ogni gruppo SCSS segue lo schema allowlist + `@extend %<selettore> !optional`, saltando i selettori presenti in `$skipComponentStyles` (feature-libs/order/styles/components/order-details/_index.scss:L11-L37). L'allowlist del dettaglio elenca anche `cx-order-document-flow` e `cx-order-document-flow-dialog`, i cui placeholder stanno però in feature-libs/order/document-flow/styles/_document-flow.scss:L1-L1. Il placeholder `%cx-order-overview` è definito due volte, in feature-libs/order/styles/components/order-overview/_order-overview.scss:L1-L1 e in feature-libs/order/styles/components/order-details/_order-details-shipping.scss:L1-L1: Sass unisce le regole di entrambi. `cx-reorder-dialog` (feature-libs/order/styles/components/order-details/_reorder-dialog.scss:L1-L1) e `cx-invoices-list` (feature-libs/pdf-invoices/styles/_pdf-invoices.scss:L1-L1) usano invece selettori diretti, quindi `$skipComponentStyles` non li può escludere.

4. Diagrammi

4.1 Struttura a livelli e caricamento lazy

■ **TAVOLA 27.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/order/root/order-root.module.ts:L67-L80, core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L100, projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L26-L33, feature-libs/order/core/order-core.module.ts:L17-L29, feature-libs/order/core/facade/facade-providers.ts:L26-L63, feature-libs/order/occ/order-occ.module.ts:L38-L90.

4.2 Place order → conferma

■ **TAVOLA 27.2** diagramma a tutta pagina, in fondo al capitolo

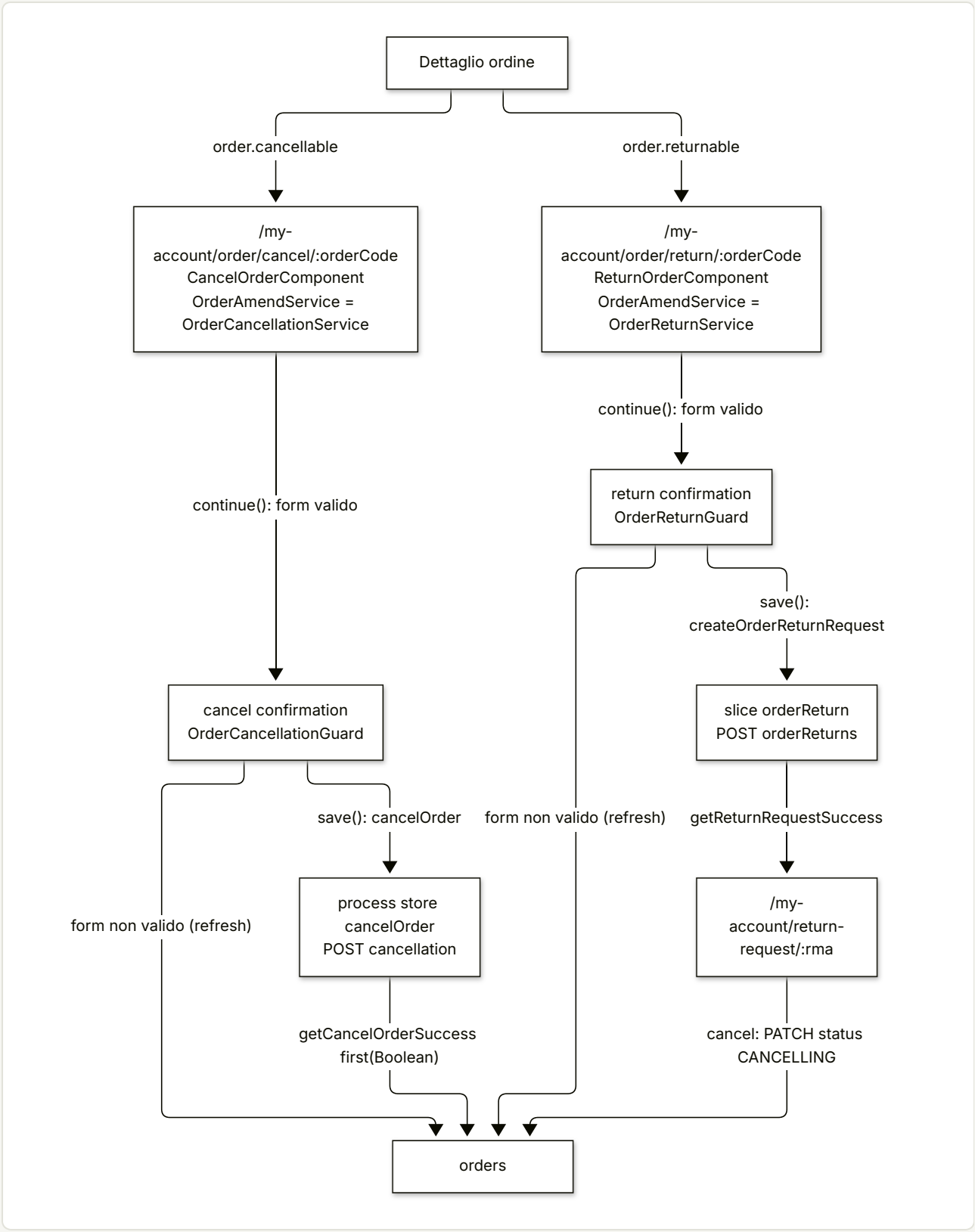
Riferimenti: feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.component.ts:L149-L170, feature-libs/order/core/facade/order.service.ts:L26-L117, feature-libs/order/occ/adapters/occ-order.adapter.ts:L38-L66, feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L24-L40, feature-libs/order/components/guards/order-confirmation.guard.ts:L24-L36.

4.3 Dettaglio ordine via NgRx

■ **TAVOLA 27.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/order/components/order-details/order-details.service.ts:L30-L55, feature-libs/order/core/facade/order-history.service.ts:L41-L59, feature-libs/order/core/store/effects/order-details.effect.ts:L35-L120, feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L57-L68, feature-libs/order/core/store/reducers/index.ts:L45-L48.

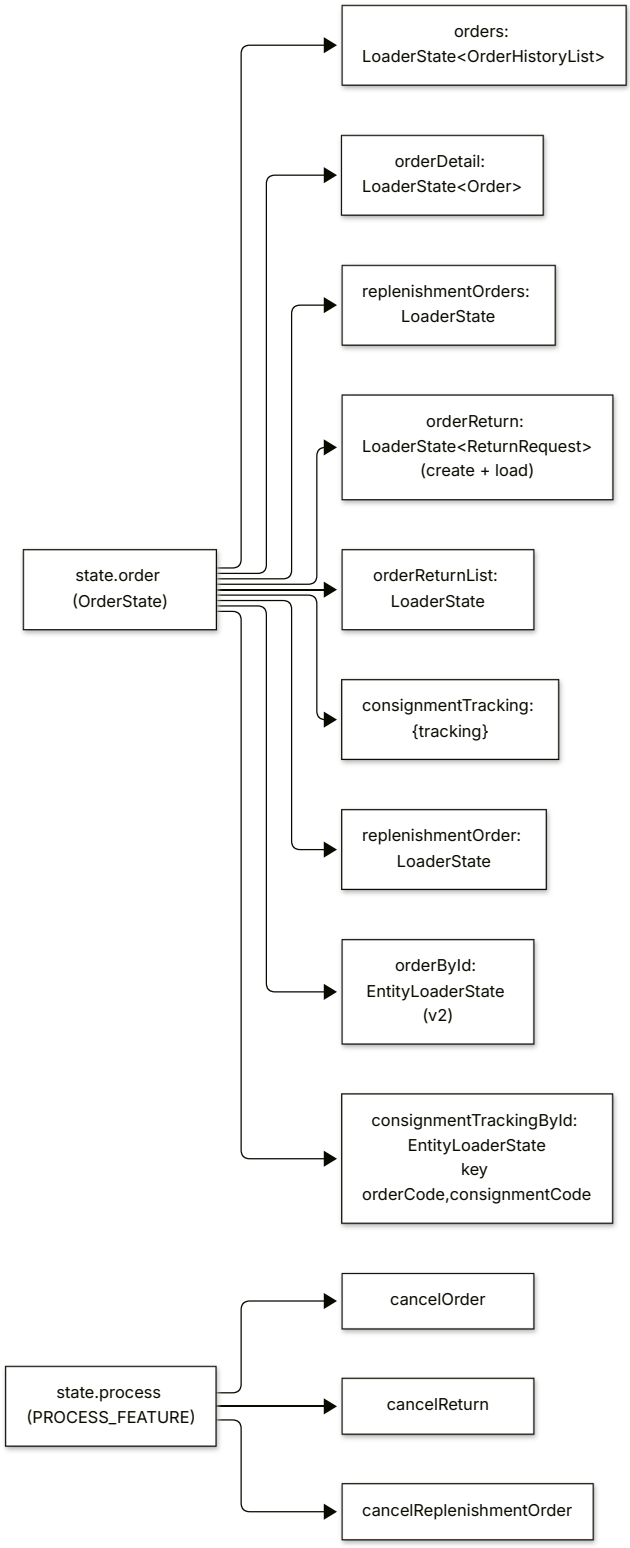
4.4 Cancellazione e reso



Riferimenti: `feature-libs/order/components/order-details/order-detail-actions/order-detail-actions.component.html:L13-L39`, `feature-libs/order/components/amend-order/cancellations/cancel-order/cancel-order.module.ts:L38-L51`, `feature-libs/order/components/amend-order/cancellations/order-cancellation.guard.ts:L24-L38`, `feature-libs/order/components/amend-order/cancellations/order-cancellation.`

service.ts:L56-L93, feature-libs/order/components/amend-order/returns/order-return.service.ts:L54-L99, feature-libs/order/components/return-request-detail/return-request.service.ts:L65-L83.

4.5 Le slice dello store `order`



Riferimenti: feature-libs/order/core/store/order-state.ts:L19-L58, feature-libs/order/core/store/reducers/index.ts:L39-L77, feature-libs/order/core/store/actions/order-details.action.ts:L60-L95.

---

5. Estratti di codice reale

5.1 Feature, alias core e dipendenza dal carrello

feature-libs/order/root/order-root.module.ts:L67-L80

```ts
// TODO: Inline this factory when we start releasing Ivy compiled libraries
export function defaultOrderComponentsConfig(): CmsConfig {
  const config: CmsConfig = {
    featureModules: {
      [ORDER_FEATURE]: {
        cmsComponents: ORDER_CMS_COMPONENTS,
        dependencies: [CART_BASE_FEATURE],
      },
      // by default core is bundled together with components
      [ORDER_CORE_FEATURE]: ORDER_FEATURE,
    },
  };
  return config;
}
```

5.2 Rotta senza path con contesto

feature-libs/order/root/order-root.module.ts:L92-L103

```ts
      {
        // @ts-ignore
        path: null,
        canActivate: [AuthGuard, CmsPageGuard],
        component: PageLayoutComponent,
        data: {
          cxRoute: 'orderDetails',
          cxContext: {
            [ORDER_ENTRIES_CONTEXT]: OrderDetailsOrderEntriesContextToken,
          },
        },
      },
```

5.3 La facade lazy dello storico (e il metodo dimenticato)

feature-libs/order/root/facade/order-history.facade.ts:L19-L43

```ts
export function orderHistoryFacadeFactory() {
  return facadeFactory({
    // @ts-ignore: Deprecated methods will be removed with next major release
    facade: OrderHistoryFacade,
    feature: ORDER_CORE_FEATURE,
    methods: [
      'getOrderDetails',
      'loadOrderDetails',
      'clearOrderDetails',
      'getOrderHistoryList',
      // ...
      'getOrderDetailsLoading',
    ],
    async: true,
  });
}
```

`getQueryParams` è dichiarato `abstract` e `@deprecated` più in basso nella stessa classe ma non compare nei `methods` (feature-libs/order/root/facade/order-history.facade.ts:L144-L148).

5.4 Place order: Command e stato in memoria

feature-libs/order/core/facade/order.service.ts:L22-L54

```ts
@Injectable()
export class OrderService implements OrderFacade {
  protected placedOrder$ = new BehaviorSubject<Order | undefined>(undefined);

  protected placeOrderCommand: Command<boolean, Order> =
    this.commandService.create<boolean, Order>(
      (payload) =>
        this.checkoutPreconditions().pipe(
          switchMap(([userId, cartId]) =>
            this.orderConnector.placeOrder(userId, cartId, payload).pipe(
              tap((order) => {
                this.placedOrder$.next(order);
                this.eventService.dispatch(
                  {
                    userId,
                    cartId,
                    /**
                     * As we know the cart is not anonymous (precondition checked),
                     * we can safely use the cartId, which is actually the cart.code.
                     */
                    cartCode: cartId,
                    order,
                  },
                  OrderPlacedEvent
```

```
              );
            })
          )
        )
      ),
    {
      strategy: CommandStrategy.CancelPrevious,
    }
  );
```

5.5 Precondizioni del checkout

feature-libs/order/core/facade/order.service.ts:L99-L117

```ts
protected checkoutPreconditions(): Observable<[string, string]> {
  return combineLatest([
    this.userIdService.takeUserId(),
    this.activeCartFacade.takeActiveCartId(),
    this.activeCartFacade.isGuestCart(),
  ]).pipe(
    take(1),
    map(([userId, cartId, isGuestCart]) => {
      if (
        !userId ||
        !cartId ||
        (userId === OCC_USER_ID_ANONYMOUS && !isGuestCart)
      ) {
        throw new Error('Checkout conditions not met');
      }
      return [userId, cartId];
    })
  );
}
```

5.6 Adapter OCC del place order

feature-libs/order/occ/adapters/occ-order.adapter.ts:L38-L66

```ts
public placeOrder(
  userId: string,
  cartId: string,
  termsChecked: boolean
): Observable<Order> {
  let headers = new HttpHeaders({
    'Content-Type': 'application/x-www-form-urlencoded',
  });
```

```
  if (userId === OCC_USER_ID_ANONYMOUS) {
    headers = InterceptorUtil.createHeader(USE_CLIENT_TOKEN, true, headers);
  }

  return this.http
    .post<Occ.Order>(
      this.getPlaceOrderEndpoint(userId, cartId, termsChecked.toString()),
      {},
      { headers }
    )
    .pipe(
      catchError((error) => {
        throw tryNormalizeHttpError(error, this.logger);
      }),
      backOff({
        shouldRetry: isJaloError,
      }),
      this.converter.pipeable(ORDER_NORMALIZER)
    );
}
```

5.7 La guard della pagina di conferma

feature-libs/order/components/guards/order-confirmation.guard.ts:L24-L36

```ts
canActivate(): Observable<GuardResult> {
  return this.orderFacade.getOrderDetails().pipe(
    map((orderDetails) => {
      if (orderDetails && Object.keys(orderDetails).length !== 0) {
        return true;
      } else {
        return this.router.parseUrl(
          this.semanticPathService.get('orders') ?? ''
        );
      }
    })
  );
}
```

5.8 Riuso dei componenti di dettaglio sostituendo la sorgente dati

feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L96-L105

```ts
      OrderConfirmationOverviewComponent: {
        component: OrderOverviewComponent,
        providers: [
          {
```

```
            provide: OrderDetailsService,
            useExisting: OrderFacade,
          },
        ],
        guards: [OrderConfirmationGuard],
      },
```

5.9 `OrderDetailsService`: il router guida il caricamento

feature-libs/order/components/order-details/order-details.service.ts:L30-L55

```ts
  this.orderCode$ = this.routingService.getRouterState().pipe(
    map((routingData) => routingData.state.params.orderCode),
    distinctUntilChanged()
  );

  this.orderLoad$ = this.orderCode$.pipe(
    tap((orderCode) => {
      if (orderCode) {
        this.orderHistoryFacade.loadOrderDetails(orderCode);
      } else {
        this.orderHistoryFacade.clearOrderDetails();
      }
    }),
    shareReplay({ bufferSize: 1, refCount: true })
  );
}

isOrderDetailsLoading(): Observable<boolean> {
  return this.orderHistoryFacade.getOrderDetailsLoading();
}

getOrderDetails(): Observable<Order> {
  return this.orderLoad$.pipe(
    switchMap(() => this.orderHistoryFacade.getOrderDetails())
  );
}
```

5.10 "Load on subscribe" della lista ordini

feature-libs/order/core/facade/order-history.service.ts:L71-L87

```ts
getOrderHistoryList(
  pageSize: number
): Observable<OrderHistoryList | undefined> {
  return this.store.pipe(
    select(OrderSelectors.getOrdersState),
```

```
    tap((orderListState) => {
      const attemptedLoad =
        orderListState.loading ||
        orderListState.success ||
        orderListState.error;
      if (!attemptedLoad) {
        this.loadOrderList(pageSize);
      }
    }),
    map((orderListState) => orderListState.value)
  );
}
```

5.11 Lista ordini o ordini generati da un replenishment

feature-libs/order/core/store/effects/orders.effect.ts:L33-L53

```ts
loadUserOrders$: Observable<OrderActions.UserOrdersAction> = createEffect(
  () =>
    this.actions$.pipe(
      ofType(OrderActions.LOAD_USER_ORDERS),
      map((action: OrderActions.LoadUserOrders) => action.payload),
      switchMap((payload) => {
        return (
          payload.replenishmentOrderCode
            ? this.replenishmentOrderConnector.loadReplenishmentDetailsHistory(
                payload.userId,
                payload.replenishmentOrderCode ?? '',
                payload.pageSize,
                payload.currentPage,
                payload.sort
              )
            : this.orderConnector.getHistory(
                payload.userId,
                payload.pageSize,
                payload.currentPage,
                payload.sort
              )
```

5.12 Lo stato della feature

feature-libs/order/core/store/order-state.ts:L37-L47

```ts
export interface OrderState {
  orders: StateUtils.LoaderState<OrderHistoryList>;
  orderDetail: StateUtils.LoaderState<Order>;
  replenishmentOrders: StateUtils.LoaderState<ReplenishmentOrderList>;
```

```
  orderReturn: StateUtils.LoaderState<ReturnRequest>;
  orderReturnList: StateUtils.LoaderState<ReturnRequestList>;
  consignmentTracking: ConsignmentTrackingState;
  replenishmentOrder: StateUtils.LoaderState<ReplenishmentOrder>;
  consignmentTrackingById: StateUtils.EntityLoaderState<ConsignmentTracking>;
  orderById: StateUtils.EntityLoaderState<Order>;
}
```

5.13 Form di amend e prezzo calcolato lato client

feature-libs/order/components/amend-order/amend-order.service.ts:L86-L108

```ts
  private buildForm(order: Order): void {
    this.form = new UntypedFormGroup({});
    this.form.addControl('orderCode', new UntypedFormControl(order.code));

    const entryGroup = new UntypedFormGroup(
      {},
      { validators: [ValidateQuantityToCancel] }
    );
    this.form.addControl('entries', entryGroup);

    (order.entries || []).forEach((entry) => {
      const key = entry?.entryNumber?.toString() ?? '';
      entryGroup.addControl(
        key,
        new UntypedFormControl(0, {
          validators: [
            Validators.min(0),
            Validators.max(this.getMaxAmendQuantity(entry)),
          ],
        })
      );
    });
  }
```

feature-libs/order/components/amend-order/amend-order.service.ts:L119-L135

```ts
  /**
   * As discussed, this calculation is moved to SPA side.
   * The calculation and validation should be in backend facade layer.
   */
  getAmendedPrice(entry: OrderEntry): Price {
    const amendedQuantity = this.getFormControl(this.form, entry).value;
    const amendedPrice = Object.assign({}, entry.basePrice);
    amendedPrice.value =
      Math.round((entry.basePrice?.value ?? 0) * amendedQuantity * 100) / 100;
```

```
  amendedPrice.formattedValue = formatCurrency(
    amendedPrice.value,
    // TODO: user current language
    'en',
    getCurrencySymbol(amendedPrice.currencyIso ?? '', 'narrow'),
    amendedPrice.currencyIso
  );
```

5.14 Cancellazione: attesa sul process state

feature-libs/order/components/amend-order/cancellations/order-cancellation.service.ts:L56-L79

```ts
save(): void {
  const orderCode = this.form.value.orderCode;
  const entries = this.form.value.entries;
  const inputs: CancelOrReturnRequestEntryInput[] = Object.keys(entries)
    .filter((entryNumber) => <number>entries[entryNumber] > 0)
    .map(
      (entryNumber) =>
        ({
          orderEntryNumber: Number(entryNumber),
          quantity: <number>entries[entryNumber],
        }) as CancelOrReturnRequestEntryInput
    );

  this.form.reset();

  this.orderHistoryFacade.cancelOrder(orderCode, {
    cancellationRequestEntryInputs: inputs,
  });

  this.orderHistoryFacade
    .getCancelOrderSuccess()
    .pipe(first(Boolean))
    .subscribe(() => this.afterSave(orderCode));
}
```

5.15 Reorder: cancella il carrello e aspetta

feature-libs/order/core/facade/reorder-order.service.ts:L56-L92

```ts
protected reorderPreconditions(): Observable<string> {
  return combineLatest([
    this.userIdService.takeUserId(),
    this.activeCartFacade.getActiveCartId(),
  ]).pipe(
    take(1),
```

```
    switchMap(([userId, cartId]) => {
      if (!userId) {
        return throwError(() => new Error('Must be logged in to reorder'));
      }

      if (cartId) {
        return this.deleteCartAndWait(cartId, userId);
      }

      return of(userId);
    })
  );
}

protected deleteCartAndWait(
  cartId: string,
  userId: string
): Observable<string> {
  this.multiCartFacade.deleteCart(cartId, userId);
  // Wait for the cart entity to be fully removed from the store
  // before proceeding with the reorder.
  return this.multiCartFacade.getCartEntity(cartId).pipe(
    filter((state) => !state.loading && (!state.value || !!state.error)),
    take(1),
    switchMap((state) =>
      state.error
        ? throwError(() => new Error('Failed to delete cart before reorder'))
        : of(userId)
    )
  );
}
```

5.16 Reso: abbinamento entry ↔ consegna per codice prodotto

feature-libs/order/components/amend-order/returns/return-order/return-order.component.ts:L42-L73

TS

```
entries$: Observable<OrderEntry[]> = combineLatest([
  this.orderAmendService.getEntries(),
  this.consignments$,
]).pipe(
  map(([entries, consignments]) => {
    // Flatten all consignment entries
    const consignmentEntries = consignments.flatMap(
      (consignment) => consignment.entries ?? []
    );
    return entries
      .map<OrderEntry | null>((entry) => {
        // Find matching consignment entry by product code
```

```
          const consignmentEntry = consignmentEntries.find(
            (ce) => ce.orderEntry?.product?.code === entry.product?.code
          );
          // If found, update the max quantity with shippedQuantity
          return consignmentEntry
            ? {
                ...entry,
                returnableQuantity:
                  consignmentEntry.shippedQuantity ??
                  entry.returnableQuantity ??
                  0,
              }
            : null;
        })
        .filter(
          (entry): entry is OrderEntry =>
            !!entry && entry.returnableQuantity !== 0
        );
    })
  );
```

5.17 Outlet registrato condizionalmente (My Account v2)

feature-libs/order/components/order-details/order-details.module.ts:L61-L78

```ts
function registerOrderOutletFactory(): () => void {
  const isMyAccountV2 = inject(USE_MY_ACCOUNT_V2_ORDER);
  const outletService = inject(OutletService);
  const componentFactoryResolver = inject(ComponentFactoryResolver);
  return () => {
    const config: ProvideOutletOptions = {
      component: MyAccountV2ConsignmentTrackingComponent,
      id: OrderOutlets.ORDER_CONSIGNMENT,
      position: OutletPosition.REPLACE,
    };
    if (isMyAccountV2) {
      const template = componentFactoryResolver.resolveComponentFactory(
        config.component
      );
      outletService.add(config.id, template, config.position);
    }
  };
}
```

5.18 Endpoint e URL del dettaglio con quote code

feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L212-L231

```ts
protected getOrderDetailUrl(userId: string, orderCode: string): string {
  return this.orderConfig.showOrderQuoteLink
    ? (() => {
        const scopes = ['orderDetail', 'quoteCode'];
        const scopedDataWithUrls: ScopedDataWithUrl[] = scopes.map(
          (scope) => ({
            scopedData: { scope, userId, orderCode },
            url: this.occEndpoints.buildUrl(scope, {
              urlParams: { userId, orderId: orderCode },
            }),
          })
        );
        const mergedUrl =
          this.occFieldsService.getOptimalUrlGroups(scopedDataWithUrls);
        return Object.keys(mergedUrl)[0];
      })()
    : this.occEndpoints.buildUrl('orderDetail', {
        urlParams: { userId, orderId: orderCode },
      });
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
orderAttachments.previewMimeTypes	string[]	13 MIME (testo, JSON, XML, XHTML, 8 immagini, PDF)	feature-libs/order/root/config/default-order-config.ts:L10-L26	feature-libs/order/components/order-details/order-attachments/attachments-dialog/order-attachments-dialog.component.ts:L107-L112
showOrderQuoteLink	boolean	false	feature-libs/order/root/config/default-order-config.ts:L27-L27	feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L213-L213; feature-libs/order/components/order-details/order-overview/order-overview.component.html:L182-L182
USE_MY_ACCOUNT_V2_ORDER (token)	boolean	false (factory)	feature-libs/order/root/tokens/context.ts:L17-L20	feature-libs/order/components/order-details/order-details.module.ts:L62-L62, :L181-L181; feature-libs/order/components/order-history/order-history.module.ts:L75-L75

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
featureModules.order / orderCore	CmsConfig	36 tipi CMS + dipendenza cartBase; orderCore alias	feature-libs/order/root/order-root.module.ts:L68-L80	core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L60
featureModules.order.module	() => Promise	**nessuno** (lo fornisce l'app)	projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L26-L33	core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35
featureModules.orderDocumentFlow	CmsConfig	AccountOrderDocumentFlowComponent	feature-libs/order/document-flow/root/order-document-flow-root.module.ts:L13-L19	idem
featureModules.pdfInvoices / pdfInvoicesCore	CmsConfig	AccountOrderDetailsPDFInvoicesComponent; alias	feature-libs/pdf-invoices/root/pdf-invoices-root.module.ts:L14-L26	idem
routing.routes.* (12 rotte)	RoutingConfig	vedi §3.2	feature-libs/order/root/config/default-order-routing-config.ts:L9-L54	rotte cxRoute in feature-libs/order/root/order-root.module.ts:L82-L172
backend.occ.endpoints.* order (17)	string	users/${userId}/...	feature-libs/order/occ/config/default-occ-order-config.ts:L9-L50	adapter OCC order
backend.occ.endpoints.reorder, scheduleReplenishmentOrder (+ override placeOrder)	string	nessuno in order; orgUsers/... nella recipe B2B	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L31-L36	feature-libs/order/occ/adapters/occ-reorder-order.adapter.ts:L31-L51; feature-libs/order/occ/adapters/occ-scheduled-replenishment-order.adapter.ts:L39-L82
backend.occ.endpoints.subsequentDocuments, subsequentDocumentsEntries	string	vedi §3.9	feature-libs/order/document-flow/occ/config/default-occ-document-flow-config-factory.ts:L10-L24	feature-libs/order/document-flow/occ/adapters/occ-order-document-flow.adapter.ts:L34-L68
backend.occ.endpoints.pdfInvoicesList Invoices, pdfInvoicesDownload InvoicePDF	string	users/${userId}/orders/${orderId}/invoices...	feature-libs/pdf-invoices/occ/config/default-occ-pdf-invoices-config.ts:L9-L19	feature-libs/pdf-invoices/occ/adapters/occ-pdf-invoices.adapter.ts:L36-L96
features.consignmentTracking	feature level	'1.2'	feature-libs/order/components/order-details/order-details.module.ts:L173-L175	feature-libs/order/components/order-details/order-detail-items/order-consigned-entries/order-consigned-entries.component.html:L19-L19

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
launch.CONSIGNMENT_TRACKING / REORDER / ORDER_ATTACHMENTS / REPLENISHMENT_ORDER / DOWNLOAD_ORDER_INVOICES	LayoutConfig	dialog inline/inlineRoot	feature-libs/order/components/order-details/order-detail-items/default-consignment-tracking-layout.config.ts:L10-L18; feature-libs/order/components/order-details/reoder-layout.config.ts:L10-L18; feature-libs/order/components/order-details/order-attachments/default-order-attachments-dialog-layout.config.ts:L10-L18; feature-libs/order/components/replenishment-order-details/default-replenishment-order-cancellation-layout.config.ts:L10-L18; feature-libs/order/components/order-details/my-account-v2/download-invoices/default-my-account-v2-download-invoices-layout.config.ts:L10-L18	LaunchDialogService
launch.ORDER_DOCUMENT_FLOW	LayoutConfig	OrderDocumentFlowDialogComponent inlineRoot	feature-libs/order/document-flow/components/default-order-document-flow-dialog-layout.config.ts:L10-L18	LaunchDialogService
Toggle addTitleToAddressCard	FeatureToggles	—	core	feature-libs/order/components/order-details/order-overview/order-overview.component.ts:L215-L234
Toggle useEnhancedSecurePasswordValidators, authorizationCodeFlowByDefault	FeatureToggles	—	core	feature-libs/order/components/order-confirmation/order-guest-register-form/order-guest-register-form.component.ts:L47-L53, :L79-L79
i18n chunk order, myAccountV2Order	TranslationChunksConfig	17 lingue	feature-libs/order/assets/translations/translations.ts:L9-L42	app (provideConfig i18n)
i18n chunk documentFlow	TranslationChunksConfig	solo en	feature-libs/order/document-flow/assets/translations/translations.ts:L9-L14	app

Note: OrderConfig è una classe astratta providedIn: 'root' con useExisting: Config che augmenta Config di @spartacus/core (feature-libs/order/root/config/order-config.ts:L10-L23); OrderOccEndpoints augmenta OccEndpoints (feature-libs/order/occ/model/occ-order-endpoints.model.ts:L104-L106). L'app demo B2B applica defaultB2bOccConfig con provideConfig (projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L10-L20).

7. Commenti e note notevoli dal sorgente

- // by default core is bundled together with components — spiega perché orderCore è un alias di order: core e componenti stanno nello stesso chunk (feature-libs/order/root/order-root.module.ts:L75-L75, uguale in feature-libs/pdf-invoices/root/pdf-invoices-root.module.ts:L20-L20).
- // TODO: Inline this factory when we start releasing Ivy compiled libraries (feature-libs/order/root/order-root.module.ts:L67-L67).
- // @ts-ignore: Deprecated methods will be removed with next major release e @deprecated: Method doesn't pass facade method's requirements (returns void or Observable) su getQueryParams (feature-

libs/order/root/facade/order-history.facade.ts:L21-L21, :L144-L148). L'implementazione restituisce sempre `null` (feature-libs/order/core/facade/order-history.service.ts:L220-L225).

- `As we know the cart is not anonymous (precondition checked), we can safely use the cartId, which is actually the cart.code.` (feature-libs/order/core/facade/order.service.ts:L38-L41): attenzione, la precondizione consente anche l'anonimo **con carrello guest** (:L107-L111).
- `// Wait for the cart entity to be fully removed from the store before proceeding with the reorder.` (feature-libs/order/core/facade/reorder-order.service.ts:L81-L82).
- `// TODO: for future releases, refactor this part to thrown errors` — errori inghiottiti in `loadOrderList` e nei servizi resi/replenishment (feature-libs/order/core/facade/order-history.service.ts:L126-L128).
- `As discussed, this calculation is moved to SPA side. The calculation and validation should be in backend facade layer. + // TODO: user current language` (feature-libs/order/components/amend-order/amend-order.service.ts:L119-L132).
- `the order code is not available in the route as long as we're inside a guard, hence we redirect to the common orders page.` (feature-libs/order/components/amend-order/cancellations/order-cancellation.guard.ts:L28-L30).
- `When "Order Return" feature is enabled, this component becomes one tab in TabParagraphContainerComponent.` (feature-libs/order/components/order-history/order-history.component.ts:L88-L91).
- `Converts 2021-10-15 to 2021-10-15T15:38:05-05:00` — JSDoc del serializer replenishment (feature-libs/order/occ/adapters/converters/occ-scheduled-replenishment-order-form-serializer.ts:L34-L38).
- `//guards: inherited from standard config,` — nel mapping v2 i guard non vengono ripetuti perché il merge della config li conserva (feature-libs/order/components/order-details/order-details.module.ts:L84-L84).
- `// Every request that doesn't specify fields should be set to DEFAULT` (feature-libs/pdf-invoices/components/invoices-list/invoices-list.component.ts:L149-L149).
- Typo nei nomi: costante `ClEAR_REPLENISHMENT_ORDER_DETAILS` con "l" minuscola (feature-libs/order/core/store/actions/replenishment-order-details.action.ts:L20-L21), file `reoder-layout.config.ts`, factory PDF chiamata `defaultRequestedDeliveryDateComponentsConfig` (feature-libs/pdf-invoices/root/pdf-invoices-root.module.ts:L14-L14), campo `orderAttachmentsConnector` in `OrderDocumentFlowService` (feature-libs/order/document-flow/core/services/order-document-flow.service.ts:L18-L36).

8. Extension point

COSA	COME	RIFERIMENTO	RICETTA
Sostituire la logica di una facade (es. place order con controlli extra)	Estendere il service concreto esportato da `@spartacus/order/core` e fornire `{ provide: OrderFacade, useClass: ... }` nel modulo lazy (accanto a `OrderModule`)	feature-libs/order/core/facade/facade-providers.ts:L26-L63; feature-libs/order/core/public_api.ts:L7-L12	50 — Ricette: configurazione, DI, interceptor, eventi
Reagire al place order / replenishment	`EventService.get(OrderPlacedEvent)` / `ReplenishmentOrderScheduledEvent`	feature-libs/order/root/events/order.events.ts:L14-L72; esempio feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L24-L40	50

COSA	COME	RIFERIMENTO	RICETTA
Cambiare chiamate HTTP	Fornire un adapter (`OrderHistoryAdapter`, `OrderAdapter`, …) o estendere un `Occ*Adapter` (metodi URL `protected`)	feature-libs/order/core/connectors/order-history.adapter.ts:L19-L112; feature-libs/order/occ/order-occ.module.ts:L38-L90	52 — Ricette: OCC, adapter, converter, endpoint
Arricchire i dati	Aggiungere un converter multi su `ORDER_NORMALIZER` & co. (alcuni token non hanno converter di default)	feature-libs/order/root/tokens/converters.ts:L24-L62; core-libs/core/src/util/converter.service.ts:L80-L88	52
Cambiare campi/endpoint	`provideConfig({ backend: { occ: { endpoints: { orderDetail: ... } } } })`	feature-libs/order/occ/config/default-occ-order-config.ts:L9-L50; feature-libs/order/occ/model/occ-order-endpoints.model.ts:L104-L106	52
Sostituire un componente CMS	`cmsComponents: { AccountOrderDetailsOverviewComponent: { component: ... } }` (i guard restano per merge)	feature-libs/order/components/order-details/order-details.module.ts:L128-L172	51 — Ricette: componenti CMS, outlet, route e layout
Riusare un componente di dettaglio con altra sorgente	`providers: [{ provide: OrderDetailsService, useExisting: ... }]` nel mapping CMS	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L96-L129	51
Inserire contenuto nelle consegne	Outlet `OrderOutlets.ORDER_CONSIGNMENT`, `CONSIGNMENT_DELIVERY_INFO`, `SERVICE_DETAILS`	feature-libs/order/root/model/order-outlets.model.ts:L10-L14	51
Logica di cancel/reso	Estendere `OrderAmendService` / `OrderCancellationService` / `OrderReturnService` e fornirli nel mapping	feature-libs/order/components/amend-order/amend-order.service.ts:L34-L153; feature-libs/order/components/amend-order/cancellations/cancel-order/cancel-order.module.ts:L38-L51	51
Visualizzazione delivery mode	Sostituire `OrderOverviewComponentService.shouldShowDeliveryMode`	feature-libs/order/components/order-details/order-overview/order-overview-component.service.ts:L10-L15	50
Rotte	Override di `routing.routes.orderDetails.paths` ecc.	feature-libs/order/root/config/default-order-routing-config.ts:L9-L54	51
Dialog	Override di `launch.REORDER` ecc. in `LayoutConfig`	feature-libs/order/components/order-details/reoder-layout.config.ts:L10-L18	51

COSA	COME	RIFERIMENTO	RICETTA
UI v2	`{ provide: USE_MY_ACCOUNT_V2_ORDER, useValue: true }`	feature-libs/order/root/tokens/context.ts:L17-L20; projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L34-L37	53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni
Traduzioni e stili	Chunk `order`, `myAccountV2Order`; scss `feature-libs/order/_index.scss`	feature-libs/order/assets/translations/translations.ts:L9-L42; feature-libs/order/_index.scss:L1-L5	53
Upgrade	Non usare simboli non esportati (`OccReorderOrderAdapter`, `OccReorderOrderNormalizer`, `ReturnRequestService` dei componenti, store interno)	feature-libs/order/occ/adapters/index.ts:L7-L12; feature-libs/order/core/public_api.ts:L7-L12	54 — API pubblica vs interna, upgrade

Avvertenze specifiche:

- Un override di facade va fornito **nel modulo lazy**: il proxy root risolve la facade con `injector.get(facadeClass)` sull'injector del modulo caricato (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52). Nell'app demo il punto naturale è `OrderWrapperModule` (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27).
- Aggiungere un metodo a una facade richiede di aggiungerlo anche ai `methods` del factory, altrimenti il proxy non lo espone (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L117-L127).
- Lo store (actions, selectors, `order-state`) è esportato da `@spartacus/order/core`, ma effects e reducers no (feature-libs/order/core/public_api.ts:L7-L12).

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Refresh della conferma = ordine perso.** L'ordine piazzato vive solo nel `BehaviorSubject` (feature-libs/order/core/facade/order.service.ts:L24-L24): ricaricando `/order-confirmation` il servizio riparte vuoto e `OrderConfirmationGuard` reindirizza a `orders` (feature-libs/order/components/guards/order-confirmation.guard.ts:L27-L32).
2. **Il primo componente distrutto svuota l'ordine per tutti.** Tre componenti di conferma chiamano `clearPlacedOrder()` in `ngOnDestroy` (vedi §3.3, passo 12). `DA VERIFICARE`: la sequenza effettiva di distruzione a runtime e se qualche componente ancora vivo riceve `undefined` (nessun test eseguito, `node_modules` assente).
3. `placePaymentAuthorizedOrder` **con** `cartId` **esplicito** non usa il command (niente `CancelPrevious`) né controlla anonimo/guest (feature-libs/order/core/facade/order.service.ts:L127-L153).
4. **Il reorder cancella il carrello attivo** prima di creare quello nuovo (feature-libs/order/core/facade/reorder-order.service.ts:L67-L69, :L76-L80). Se la cancellazione fallisce: `'Failed to delete cart before reorder'`.
5. **Endpoint B2B-only.** `reorder` e `scheduleReplenishmentOrder` esistono solo in `defaultB2bOccConfig` (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L31-L36). Senza, `OccEndpointsService.buildUrl` non lancia: ripiega sul **nome della chiave** come path (`|| endpoint`), quindi la richiesta parte verso `<baseUrl>/reorder?orderCode=...` e fallisce lato server (core-libs/core/src/occ/services/occ-endpoints.service.ts:L196-L224).

6. `getQueryParams` **assente dal proxy.** Non è nei `methods` di `OrderHistoryFacade` (feature-libs/order/root/facade/order-history.facade.ts:L24-L40); `OrderHistoryComponent.getQueryParams` lo chiama (feature-libs/order/components/order-history/order-history.component.ts:L131-L133). Funziona solo perché il componente vive nel modulo lazy e riceve il servizio reale, che restituisce `null`. DA VERIFICARE a runtime quale istanza riceve il componente (dedotto dal funzionamento degli injector dei moduli lazy, non osservato).
7. **Reset asimmetrici al cambio di contesto.** `resetUserOrders$` reagisce solo a `LANGUAGE_CHANGE` (feature-libs/order/core/store/effects/orders.effect.ts:L73-L73), `resetOrderDetails$` anche a `CURRENCY_CHANGE` (feature-libs/order/core/store/effects/order-details.effect.ts:L91-L94).
8. `forEach` **su undefined negli errori di cancel.** `error.error?.errors.forEach(...)` lancia se `error.error` esiste senza `errors` (feature-libs/order/core/store/effects/order-details.effect.ts:L69-L69; analogo feature-libs/order/core/store/effects/replenishment-order-details.effect.ts:L72-L72).
9. **Create e load del reso condividono la slice** `orderReturn` (feature-libs/order/core/store/reducers/index.ts:L53-L55): il "dettaglio" dopo la creazione è la risposta della create; `ReturnRequestService` ricarica solo se `rma` differisce (feature-libs/order/components/return-request-detail/return-request.service.ts:L48-L53).
10. **Cancel di un replenishment non in lista svuota la lista**: se l'indice è `-1` il reducer restituisce `initialState` (feature-libs/order/core/store/reducers/replenishment-orders.reducer.ts:L40-L42).
11. **N+1 richieste in My Account v2.** Un dettaglio per ogni ordine + un tracking per ogni consegna con `trackingID` (feature-libs/order/core/facade/my-account-v2-order-history.service.ts:L54-L67, :L100-L118).
12. **Emissioni ripetute v2.** In `getOrderHistoryList` lo `switchMap` restituisce un array (`orderHistory.orders.map(...)`), quindi emette lo stesso oggetto una volta per ordine (feature-libs/order/core/facade/my-account-v2-order-history.service.ts:L159-L167). `getOrderDetailsWithTracking` fa `push` sulle consegne dentro un `map`: emissioni ripetute del tracking possono duplicarle (feature-libs/order/core/facade/my-account-v2-order-history.service.ts:L52-L67) — DA VERIFICARE con un test.
13. **Reso abbinato per codice prodotto, non per** `entryNumber`: due entry dello stesso prodotto prendono la stessa `shippedQuantity`; le entry senza consegna spariscono (feature-libs/order/components/amend-order/returns/return-order/return-order.component.ts:L53-L71).
14. **La conferma reso non controlla** `form.valid`, a differenza della conferma cancel (feature-libs/order/components/amend-order/returns/return-order-confirmation/return-order-confirmation.component.ts:L47-L50, feature-libs/order/components/amend-order/cancellations/cancel-order-confirmation/cancel-order-confirmation.component.ts:L42-L48).
15. **Prezzo di amend formattato sempre con locale** `'en'` (feature-libs/order/components/amend-order/amend-order.service.ts:L129-L135).
16. **Refresh della conferma cancel/reso** → la guard trova il form vuoto e reindirizza a `orders` (feature-libs/order/components/amend-order/cancellations/order-cancellation.guard.ts:L24-L38, feature-libs/order/components/amend-order/returns/order-return.guard.ts:L24-L38).
17. `OrderDetailsService` **fuori dalla rotta di dettaglio**: è `providedIn: 'root'` e legge `params.orderCode`; se il param manca emette `undefined` e chiama `clearOrderDetails()` (feature-libs/order/components/order-details/order-details.service.ts:L30-L44). Il document flow lo riusa e quindi funziona solo sulla pagina di dettaglio (feature-libs/order/document-flow/components/order-document-flow/order-document-flow.component.ts:L29-L46).
18. **Document flow senza** `dependencies`, pur usando `OrderDetailsService` di `@spartacus/order/components` (feature-libs/order/document-flow/root/order-document-flow-root.module.ts:L13-L19).
19. `PDFInvoicesService` **legge** `userId` **una volta sola** nel costruttore: dopo login/logout resta il valore precedente se non si passa `userId` esplicito (feature-libs/pdf-invoices/core/services/pdf-invoices.

service.ts:L30-L34).

20. **Stream che muoiono dopo un errore.** `invoicesList$` chiude con `EMPTY` dopo un errore e i cambi successivi di pagina/sort non ricaricano (feature-libs/pdf-invoices/components/invoices-list/invoices-list.component.ts:L96-L117); analogo nel dialog del document flow (feature-libs/order/document-flow/components/order-document-flow-dialog/order-document-flow-dialog.component.ts:L92-L137).
21. **Object URL mai revocati** per fatture e allegati (feature-libs/pdf-invoices/components/invoices-list/invoices-list.component.ts:L177-L179, feature-libs/order/components/order-details/order-attachments/attachments-dialog/order-attachments-dialog.component.ts:L130-L145).
22. **Subscription non chiuse** in `MyAccountV2ConsignmentTrackingComponent` e `MyAccountV2OrderDetailsActionsComponent` (feature-libs/order/components/order-details/my-account-v2/consignment-tracking/my-account-v2-consignment-tracking.component.ts:L26-L33, feature-libs/order/components/order-details/my-account-v2/order-details-actions/my-account-v2-order-details-actions.component.ts:L26-L30).
23. `MyAccountV2OrdersComponent.getProduct` **con** `entries = []` accede a `order.entries[0].product` (feature-libs/order/components/my-account-v2/my-account-v2-orders.component.ts:L40-L49).
24. `ngAfterViewChecked` **con** `detectChanges()` **a ogni ciclo** in `MyAccountV2DownloadInvoicesComponent` (feature-libs/order/components/order-details/my-account-v2/download-invoices/my-account-v2-download-invoices.component.ts:L57-L65).
25. **Dialog cancel replenishment.** `onSuccess` chiama `clearCancelReplenishmentOrderProcessState()` anche quando il valore è `false` (feature-libs/order/components/replenishment-order-cancellation-dialog/replenishment-order-cancellation-dialog.component.ts:L88-L105).
26. **Rotte cancel/return senza** `AuthGuard` **di rotta**: la protezione è nel mapping CMS (feature-libs/order/root/order-root.module.ts:L104-L131, feature-libs/order/components/amend-order/cancellations/cancel-order/cancel-order.module.ts:L40-L42). Un mapping custom che dimentica `guards` lascia la pagina accessibile agli anonimi.
27. **Chiave entità vuota.** `LoadOrderByIdSuccess` usa `payload.code ?? ''` (feature-libs/order/core/store/actions/order-by-id.action.ts:L21-L36).
28. **Converter assenti di default.** Per `ORDER_HISTORY_NORMALIZER`, `CONSIGNMENT_TRACKING_NORMALIZER`, `ORDER_RETURNS_NORMALIZER`, `REPLENISHMENT_ORDER_HISTORY_NORMALIZER`, `ORDER_RETURN_REQUEST_INPUT_SERIALIZER` non è registrato nulla: `pipeable` passa lo stream invariato, e per il serializer dell'input del reso, applicato con `convert`, il payload viene inviato così com'è (feature-libs/order/occ/order-occ.module.ts:L63-L87, core-libs/core/src/util/converter.service.ts:L80-L88, :L106-L112, feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L140-L143). Il primo converter che aggiungete è l'unico.

---

10. Come fare debug

- **"Feature orderCore is not configured properly".** Il proxy lancia questo errore se nessuno ha fornito `featureModules.order.module` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47). Verificare il modulo feature dell'app (projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L26-L33).
- **Il modulo lazy non arriva.** Controllare nel tab Network il chunk del wrapper; la dipendenza `cartBase` viene risolta prima (feature-libs/order/root/order-root.module.ts:L71-L74, core-libs/core/src/lazy-loading/feature-modules.service.ts:L54-L68).

- **Redux DevTools.** Feature key `order`, 9 slice (§4.5). Azioni con prefisso `[Order] ...` (es. feature-libs/order/core/store/actions/orders.action.ts:L11-L14). Per cancel guardare `state.process` alle chiavi `cancelOrder`, `cancelReturn`, `cancelReplenishmentOrder`.
- **Dettaglio che non carica.** Verificare che `params.orderCode` sia presente (feature-libs/order/components/order-details/order-details.service.ts:L30-L33) e che qualcuno sia sottoscritto a `getOrderDetails()` (il caricamento è lazy, :L51-L55).
- **Lista vuota senza errori.** `loadOrderList` inghiotte l'errore di `takeUserId(true)` per l'anonimo (feature-libs/order/core/facade/order-history.service.ts:L103-L129); controllare lo stato di login. Se la slice `orders` ha già `error: true`, `getOrderHistoryList` non ritenta (:L76-L83) finché non arriva `clearOrderList()`.
- **Place order fallisce subito.** Messaggio `Checkout conditions not met`: manca userId/cartId o è un anonimo senza guest cart (feature-libs/order/core/facade/order.service.ts:L107-L112). Errori Jalo vengono ritentati da `backOff` prima di arrivare al chiamante (feature-libs/order/occ/adapters/occ-order.adapter.ts:L61-L63).
- **Reorder 404 / URL strano.** Se l'URL contiene `/reorder?orderCode=` senza `orgUsers`, manca la config B2B (Trappola 5).
- **URL del dettaglio con campi inattesi.** Con `showOrderQuoteLink: true` l'URL è il risultato della fusione degli scope (feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L212-L231): stampare `getOrderDetailUrl` in un adapter esteso.
- **Breakpoint utili:** `OrderService.checkoutPreconditions` (feature-libs/order/core/facade/order.service.ts:L99-L117), `OrderConfirmationGuard.canActivate` (feature-libs/order/components/guards/order-confirmation.guard.ts:L24-L36), `OrdersEffect.loadUserOrders$` (feature-libs/order/core/store/effects/orders.effect.ts:L38-L54), `OrderCancellationGuard` (feature-libs/order/components/amend-order/cancellations/order-cancellation.guard.ts:L24-L38), `ReorderOrderService.deleteCartAndWait` (feature-libs/order/core/facade/reorder-order.service.ts:L76-L92).
- **Eventi.** Sottoscrivere temporaneamente `EventService.get(OrderPlacedEvent)` per vedere `userId`, `cartId`, `cartCode`, `order` (feature-libs/order/root/events/order.events.ts:L14-L72).
- **Test.** La libreria ha una config Vitest (`feature-libs/order/vitest.config.ts`); vedi il capitolo 44.

---

11. Domande di autoverifica

1. **Perché `OrderRootModule` non basta da solo per usare `OrderFacade`?** Perché la feature è "configurata" solo se ha `module`, che il root module non fornisce; il proxy lancia `Feature ... is not configured properly`. — core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35; core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47.
2. **Dove si trova l'ordine appena piazzato e cosa succede con un refresh della pagina di conferma?** In un `BehaviorSubject` di `OrderService`, non nello store; al refresh è vuoto e `OrderConfirmationGuard` reindirizza a `orders`. — feature-libs/order/core/facade/order.service.ts:L24-L24; feature-libs/order/components/guards/order-confirmation.guard.ts:L24-L36.
3. **Come fa `OrderOverviewComponent` a funzionare sia nel dettaglio ordine sia nella conferma?** Legge da `OrderDetailsService`; il mapping CMS della conferma fornisce `{ provide: OrderDetailsService, useExisting: OrderFacade }`, e il componente gestisce l'assenza di `isOrderDetailsLoading` con un controllo `typeof`. — feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L96-L105; feature-libs/order/components/order-details/order-overview/order-overview.component.ts:L66-L70.

4. **Quando parte la richiesta della lista ordini?** Quando qualcuno si sottoscrive a `getOrderHistoryList` e la slice `orders` non è né loading né success né error. — `feature-libs/order/core/facade/order-history.service.ts:L71-L87`.
5. **Come vengono caricati gli ordini generati da un replenishment?** Lo stesso `OrderHistoryComponent` è montato nella pagina di dettaglio replenishment; `loadOrderList` legge `replenishmentOrderCode` dal router e `OrdersEffect` chiama `loadReplenishmentDetailsHistory`. — `feature-libs/order/core/facade/order-history.service.ts:L107-L123`; `feature-libs/order/core/store/effects/orders.effect.ts:L39-L47`.
6. **Cosa fa il reorder al carrello attivo dell'utente?** Lo cancella e aspetta che l'entità sia rimossa (o in errore) prima di chiamare il backend. — `feature-libs/order/core/facade/reorder-order.service.ts:L56-L92`.
7. **Perché un reorder in uno storefront B2C senza config B2B fallisce?** L'endpoint `reorder` non ha default nella libreria order; `buildUrl` ripiega sul nome della chiave come path. — `feature-libs/order/occ/model/occ-order-endpoints.model.ts:L86-L93`; `core-libs/core/src/occ/services/occ-endpoints.service.ts:L220-L224`.
8. **Come sa `OrderCancellationService.save` che la cancellazione è andata a buon fine?** Attende `getCancelOrderSuccess()` con `first(Boolean)`, che legge il process state `cancelOrder`. — `feature-libs/order/components/amend-order/cancellations/order-cancellation.service.ts:L75-L78`; `feature-libs/order/core/facade/order-history.service.ts:L191-L204`.
9. **Qual è la quantità massima selezionabile per il reso di una entry?** Il validatore del form usa `returnableQuantity` (o `quantity`) dell'entry originale (`buildForm` su `order.entries`, letto da `OrderDetailsService`). `ReturnOrderComponent` sostituisce `returnableQuantity` con la `shippedQuantity` della consignment entry con lo stesso codice prodotto solo nelle entry passate alla UI: il contatore e `[max]` usano quindi `shippedQuantity`, mentre `Validators.max` resta sul valore originale. — `feature-libs/order/components/amend-order/amend-order.service.ts:L68-L70`, `:L96-L106`, `:L140-L148`; `feature-libs/order/components/amend-order/returns/return-order/return-order.component.ts:L42-L73`; `feature-libs/order/components/amend-order/amend-order-items/amend-order-items.component.html:L127-L127`.
10. **Cosa cambia attivando `USE_MY_ACCOUNT_V2_ORDER`?** Mapping v2 per le azioni di dettaglio e per la storia ordini, e un `MODULE_INITIALIZER` che sostituisce il contenuto dell'outlet `ORDER_CONSIGNMENT` con il tracking v2. — `feature-libs/order/components/order-details/order-details.module.ts:L61-L87`, `:L180-L187`; `feature-libs/order/components/order-history/order-history.module.ts:L65-L77`.

TAVOLA 27.1 **4.1 Struttura a livelli e caricamento lazy**

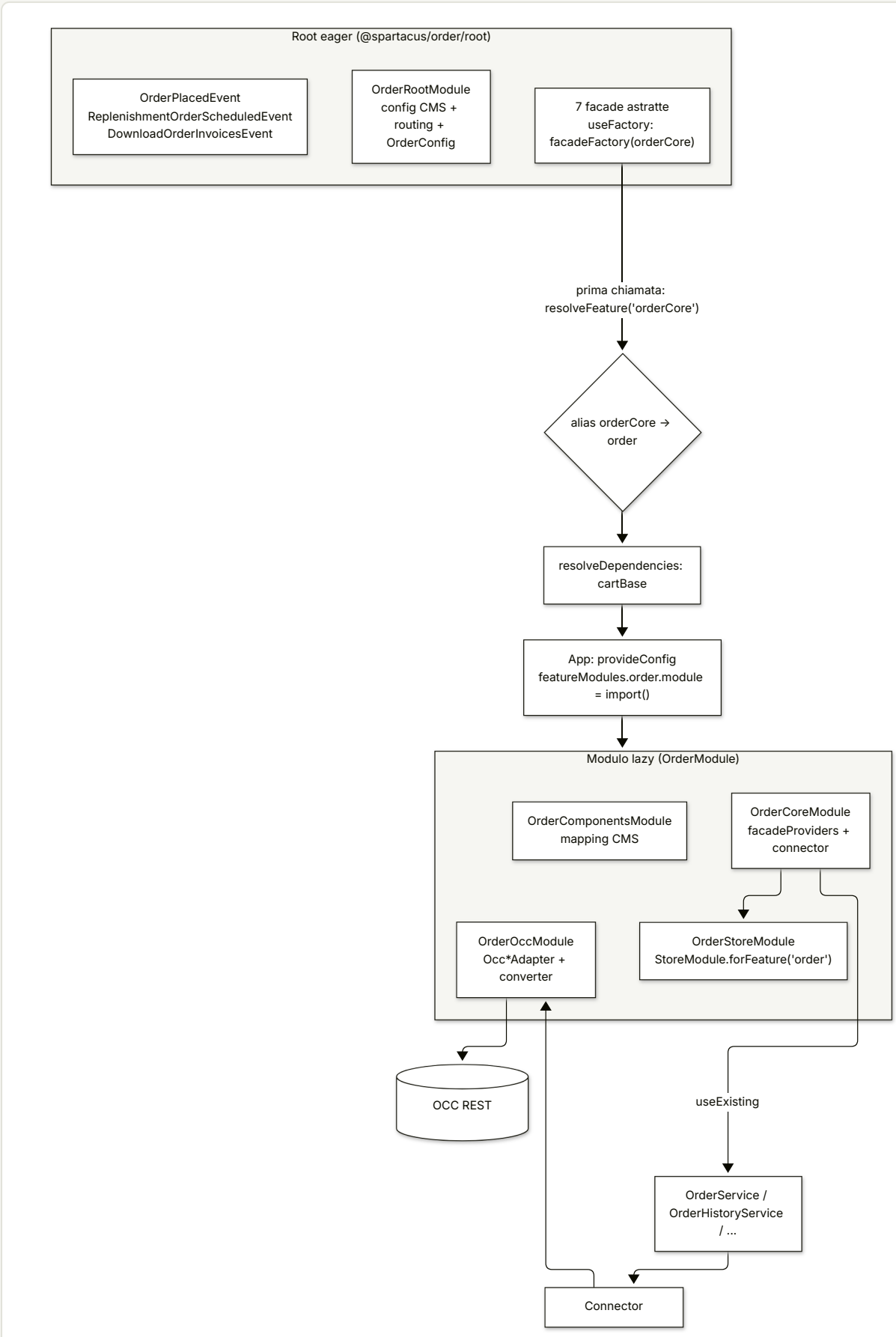


TAVOLA 27.2 **4.2 Place order → conferma**

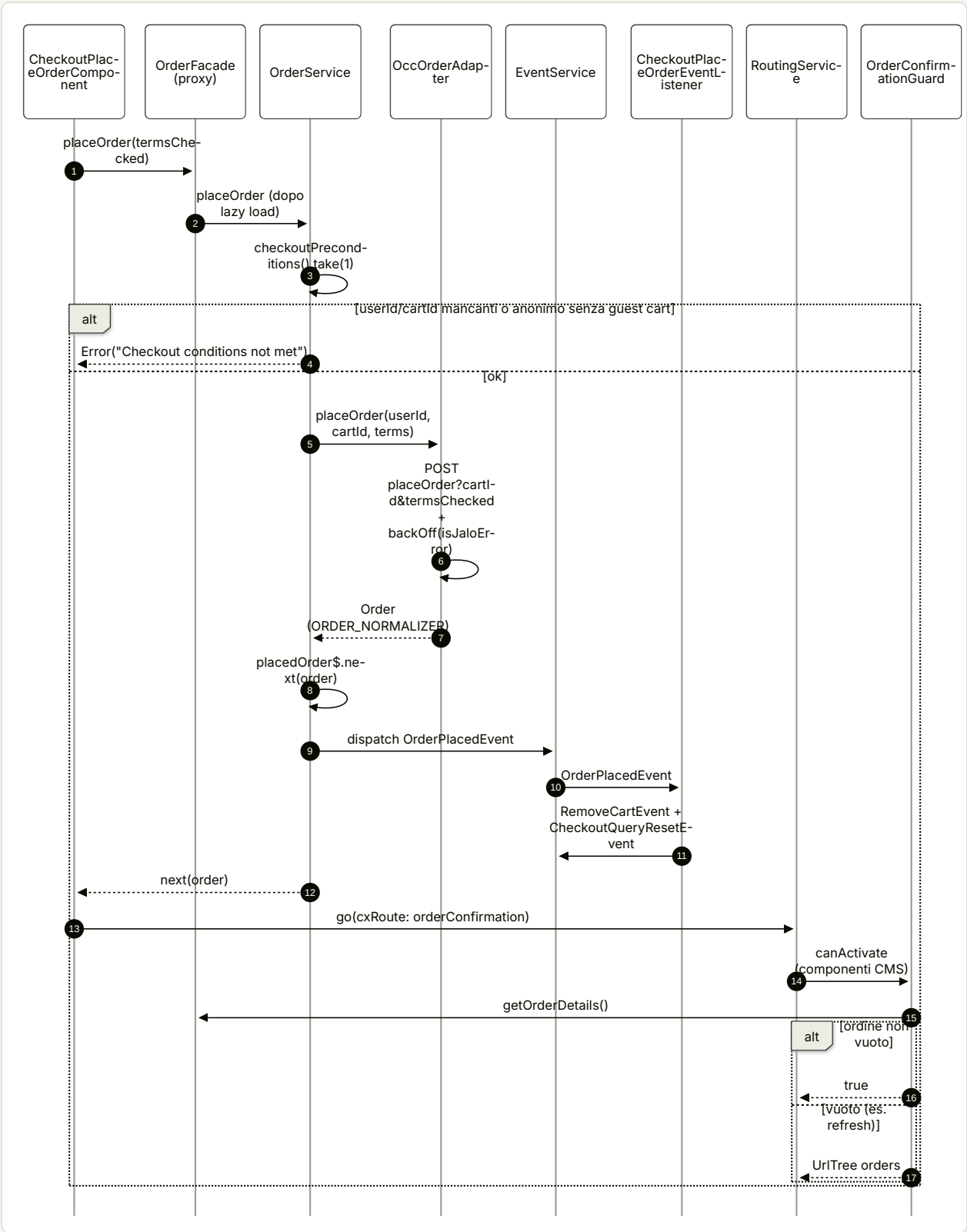
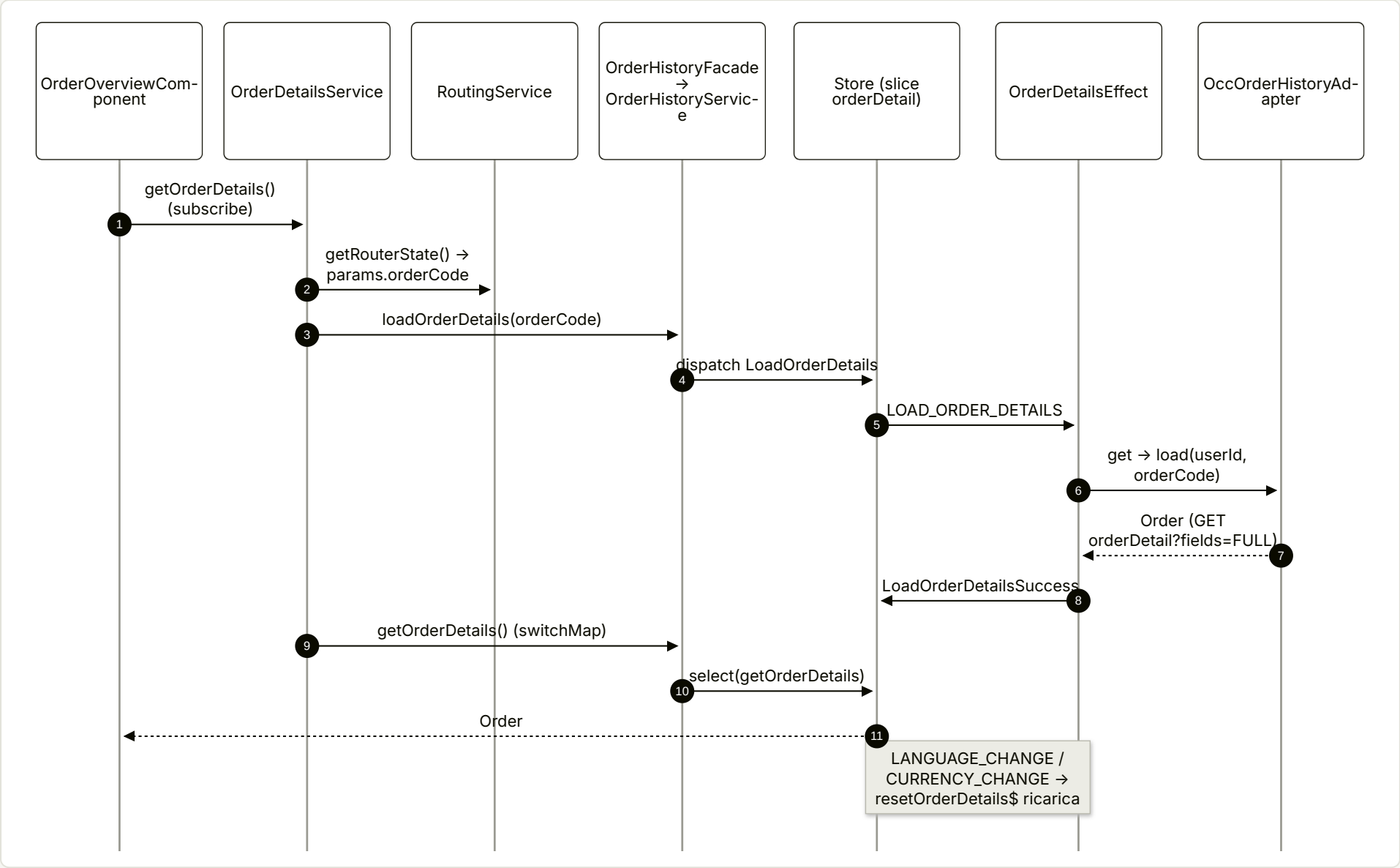


TAVOLA 27.3 **4.3 Dettaglio ordine via NgRx**



28

B2B: organization

Amministrazione aziendale, approvazioni, ordini dell'unità, estratto conto, registrazione B2B.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

`@spartacus/organization` è la libreria B2B di Spartacus. Copre le funzioni che un acquirente aziendale si aspetta oltre al carrello e al checkout: amministrare la propria azienda (unità, utenti, gruppi di utenti, budget, cost center, limiti di acquisto), approvare ordini, vedere gli ordini dell'unità, consultare l'estratto conto e registrarsi come utente B2B. È divisa in cinque sotto-pacchetti, ciascuno con il proprio entry point `root` (eager) e il proprio modulo lazy:

SOTTO-PACCHETTO	FEATURE LAZY	COSA FA
`administration`	`organizationAdministration`	CRUD delle sei entità aziendali, in un framework condiviso list/form/card
`account-summary`	`organizationAccountSummary`	estratto conto per unità, documenti e download allegati
`order-approval`	`organizationOrderApproval`	dashboard e dettaglio delle approvazioni, decisione approva/rifiuta
`unit-order`	`organizationUnitOrder`	storico e dettaglio degli ordini a livello di unità
`user-registration`	`organizationUserRegistration`	form di richiesta account B2B (anche con OTP)

L'entry point principale `@spartacus/organization` riesporta soltanto `administration` (il JSDoc parla ancora di "my-account") — feature-libs/organization/public_api.ts:L7-L10. A margine c'è `core-libs/core/src/cost-center`, che in `@spartacus/core` contiene solo i token dei converter dei cost center e un modulo vuoto — core-libs/core/src/cost-center/connectors/cost-center/converters.ts:L12-L22, core-libs/core/src/cost-center/cost-center.module.ts:L9-L17.

Il pezzo più interessante dal punto di vista architetturale è `administration`. Ha sei entità con UI quasi identiche: una lista paginata, un pannello di dettaglio, un form di creazione/modifica, sotto-liste di relazioni "assegna/rimuovi". Spartacus non scrive sei volte lo stesso codice. Usa **un solo componente CMS generico per la lista** (`ListComponent`) e **due servizi astratti** (`ListService`, `ItemService`) che ogni entità implementa. Il componente CMS li riceve dai `providers` del mapping CMS. La navigazione fra lista, dettaglio e form non usa pagine CMS separate: sono **child route Angular** dichiarate nel mapping CMS (`childRoutes`) e aggiunte a runtime alla configurazione del router. Il risultato è una "split view" a pannelli annidati in cui ogni livello è un `<router-outlet>` — feature-libs/organization/administration/components/budget/budget.config.ts:L29-L81, core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L124.

Questo capitolo segue soprattutto il **budget** da capo a fondo (CMS → route → componenti → servizi → NgRx → OCC). Poi mostra come le altre entità e gli altri sotto-pacchetti riusano lo stesso schema o se ne discostano.

Prerequisiti utili: il caricamento lazy dei feature CMS (cap. 12), le `childRoutes` e le guardie dei componenti CMS (cap. 15, cap. 16), lo store con `StateUtils` (cap. 19), le facade lazy (cap. 20) e OCC (cap. 21).

2. Mappa dei file

PATH	RUOLO
`feature-libs/organization/public_api.ts`	entry `@spartacus/organization`: riesporta `administration`

PATH	RUOLO
feature-libs/organization/administration/administration.module.ts	AdministrationModule (lazy): AdministrationCoreModule.forRoot() + OCC + componenti
feature-libs/organization/administration/root/administration-root.module.ts	AdministrationRootModule (eager): layout, UI config, routing config, CMS feature lazy
feature-libs/organization/administration/root/feature-name.ts	ORGANIZATION_ADMINISTRATION_FEATURE
feature-libs/organization/administration/root/route-params.ts	ROUTE_PARAMS: nomi dei parametri di route
feature-libs/organization/administration/root/config/default-budget-routing.config.ts	route semantiche orgBudget* (analoghi per cost center, permission, units, user, user-group)
feature-libs/organization/administration/root/config/default-organization-layout.config.ts	slot di CompanyPageTemplate
feature-libs/organization/administration/root/config/organization-ui.config.ts	OrganizationUIConfig (listSearch)
feature-libs/organization/administration/components/administration-components.module.ts	aggrega i sei moduli entità
feature-libs/organization/administration/components/shared/item.service.ts	ItemService<T>: base astratta CRUD
feature-libs/organization/administration/components/shared/current-item.service.ts	CurrentItemService<T>: item corrente dai parametri di route
feature-libs/organization/administration/components/shared/list/list.service.ts	ListService<T,P>: paginazione, ghost data, struttura tabella, ricerca
feature-libs/organization/administration/components/shared/list/list.component.ts	ListComponent (cx-org-list): componente CMS generico della lista
feature-libs/organization/administration/components/shared/form/form.service.ts	FormService<T>: form unico riusato
feature-libs/organization/administration/components/shared/form/form.component.ts	FormComponent (cx-org-form): salva e notifica
feature-libs/organization/administration/components/shared/card/card.component.ts	CardComponent (cx-org-card): pannello della split view, MessageService locale
feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts	SubListService: liste di relazioni con assign/unassign
feature-libs/organization/administration/components/shared/sub-list/assign-cell.component.ts	cella assign/unassign
feature-libs/organization/administration/components/shared/table/cell.component.ts	CellComponent: base delle celle della tabella
feature-libs/organization/administration/components/shared/detail/toggle-status-action/toggle-status.component.ts	attiva/disattiva con conferma
feature-libs/organization/administration/components/shared/message/services/message.service.ts	messaggi dentro la card
feature-libs/organization/administration/components/shared/organization.model.ts	OrganizationTableType, BaseItem

PATH	RUOLO
feature-libs/organization/administration/components/budget/budget.config.ts	CMS mapping (component, providers, childRoutes, guards) + table config del budget
feature-libs/organization/administration/components/budget/services/budget-item.service.ts	BudgetItemService
feature-libs/organization/administration/components/budget/services/budget-list.service.ts	BudgetListService
feature-libs/organization/administration/components/budget/services/current-budget.service.ts	CurrentBudgetService
feature-libs/organization/administration/components/budget/form/budget-form.service.ts	costruzione e validatori del form budget
feature-libs/organization/administration/core/administration-core.module.ts	store, guard, page meta, HttpErrorHandler 400/409, connector
feature-libs/organization/administration/core/store/organization-store.module.ts	feature NgRx organization
feature-libs/organization/administration/core/store/reducers/index.ts	reducer map e meta-reducer clearOrganizationState
feature-libs/organization/administration/core/store/effects/budget.effect.ts	effects del budget
feature-libs/organization/administration/core/services/budget.service.ts	BudgetService: servizio-facade sullo store
feature-libs/organization/administration/core/utils/get-item-status.ts	getItemStatus: esito della transizione loading → success/error
feature-libs/organization/administration/core/guards/organization-guards.module.ts	AdminGuard, UserGuard, OrgUnitGuard nel child injector
feature-libs/organization/administration/core/http-interceptors/conflict/conflict.handler.ts	gestione dei 409
feature-libs/organization/administration/occ/administration-occ.module.ts	binding adapter/converter + endpoint
feature-libs/organization/administration/occ/config/default-occ-organization-config.ts	endpoint OCC dell'administration
feature-libs/organization/administration/occ/adapters/occ-budget.adapter.ts	OccBudgetAdapter
feature-libs/organization/account-summary/root/account-summary-root.module.ts	route, componenti lazy, BlobErrorInterceptor
feature-libs/organization/account-summary/core/facade/account-summary.service.ts	implementazione di AccountSummaryFacade
feature-libs/organization/order-approval/root/order-approval-root.module.ts	componenti lazy e route dell'order approval
feature-libs/organization/order-approval/core/services/order-approval.service.ts	OrderApprovalService
feature-libs/organization/order-approval/components/details/order-approval-detail.service.ts	sostituto di OrderDetailsService
feature-libs/organization/unit-order/root/unit-order-root.module.ts	componenti lazy e route degli ordini di unità

PATH	RUOLO
feature-libs/organization/unit-order/root/facade/unit-order.facade.ts	UnitOrderFacade (async: true)
feature-libs/organization/user-registration/root/user-registration-root.module.ts	componenti lazy della registrazione B2B
feature-libs/organization/user-registration/occ/adapters/occ-user-registration.adapter.ts	POST /orgUsers con client token
core-libs/core/src/cost-center/connectors/cost-center/converters.ts	token converter cost center in @spartacus/core
projects/storefrontapp/src/app/spartacus/features/organization/organization-administration-feature.module.ts	wiring nella demo app

File aggiuntivi

PATH	RUOLO
feature-libs/organization/_index.scss	entry SCSS: mixin visible-focus e import degli stili delle cinque sotto-feature
feature-libs/organization/schematics/add-organization/index.ts	rule ng add di @spartacus/organization (addSpartacusOrganization)
feature-libs/organization/schematics/add-organization/index_spec.ts	test Jest dello schematic per le cinque feature (administration, order approval, account summary, user registration, unit order)
feature-libs/organization/vitest.config.ts	configurazione Vitest: molti alias verso helper di test di core/storefront, maxWorkers: 4, soglie coverage

3. Come funziona passo per passo

3.1 Wiring: root eager, feature lazy

L'app importa solo il **root module**. AdministrationRootModule fornisce con provideDefaultConfig la layout config, la UI config, la route semantica organization (path organization), le sei routing config delle entità e una factory che dichiara i CMS component del feature lazy — feature-libs/organization/administration/root/administration-root.module.ts:L44-L64. La factory associa al feature organizationAdministration (costante ORGANIZATION_ADMINISTRATION_FEATURE, feature-libs/organization/administration/root/feature-name.ts:L7-L7) i sei Manage*ListComponent. Quando uno di questi compare in una pagina CMS, il feature viene caricato — feature-libs/organization/administration/root/administration-root.module.ts:L24-L42.

La demo app collega il feature al codice reale. Importa AdministrationRootModule e registra module: () => import('./administration-wrapper.module'), più le traduzioni e i chunk i18n — projects/storefrontapp/src/app/spartacus/features/organization/organization-administration-feature.module.ts:L21-L47. Il wrapper importa AdministrationModule e, se environment.cdc è attivo, anche CdcAdministrationModule — projects/storefrontapp/src/app/spartacus/features/organization/administration-wrapper.module.ts:L12-L20. È questo il punto in cui le integrazioni (CDC) si agganciano al feature lazy.

AdministrationModule porta tre strati — feature-libs/organization/administration/administration.module.ts:L12-L19:

- **core** (AdministrationCoreModule.forRoot()): page meta, store, guard, due HttpErrorHandler multi (OrganizationConflictHandler, OrganizationBadRequestHandler) e, in forRoot(), i sei connector —

feature-libs/organization/administration/core/administration-core.module.ts:L21-L54;

- **OCC** (`AdministrationOccModule`): importa `CostCenterOccModule` di core, fornisce gli endpoint, lega gli adapter astratti alle classi `Occ*` con `useClass` e i converter con `useExisting` + `multi` — feature-libs/organization/administration/occ/administration-occ.module.ts:L58-L80;
- **componenti** (`AdministrationComponentsModule`): i sei moduli entità — feature-libs/organization/administration/components/administration-components.module.ts:L15-L25. Ognuno importa `SharedOrganizationModule` (cioè `ListModule`, `SubListModule`, `FormModule`) e fornisce come default la propria CMS config e la table config — feature-libs/organization/administration/components/budget/budget-components.module.ts:L18-L30, feature-libs/organization/administration/components/shared/shared-organization.module.ts:L12-L15.

Gli altri quattro sotto-pacchetti seguono lo stesso schema root/lazy. Le eccezioni sono annotate nelle sezioni 3.11-3.14.

3.2 Routing semantico e parametri

I nomi dei parametri di route stanno tutti in `ROUTE_PARAMS`. Attenzione all'indirizzo: la chiave è `addressCode` ma il valore è `'addressId'` — feature-libs/organization/administration/root/route-params.ts:L10-L18. Ogni entità ha una routing config che definisce le route semantiche e il `paramsMapping`, cioè quale proprietà del modello riempie il parametro. Per il budget le route sono `orgBudget`, `orgBudgetCreate`, `orgBudgetDetails`, `orgBudgetCostCenters`, `orgBudgetEdit`, con `budgetCode → code` — feature-libs/organization/administration/root/config/default-budget-routing.config.ts:L10-L38.

Le chiavi dei modelli non sono uniformi, e il `paramsMapping` è il punto in cui questa differenza viene assorbita:

ENTITÀ	PARAMETRO	PROPRIETÀ	RIFERIMENTO
Budget	`budgetCode`	`code`	feature-libs/organization/administration/root/config/default-budget-routing.config.ts:L10-L13
Unità	`unitCode` / `addressId` / `userCode`	`uid` / `id` / `customerId`	feature-libs/organization/administration/root/config/default-units-routing.config.ts:L10-L15
Utente	`userCode`	`customerId`	feature-libs/organization/administration/root/config/default-user-routing.config.ts:L10-L13
Gruppo utenti	`userGroupCode`	`uid`	feature-libs/organization/administration/root/config/default-user-group-routing.config.ts:L10-L13
Permesso	`permissionCode`	`code` (path `organization/purchase-limits`)	feature-libs/organization/administration/root/config/default-permission-routing.config.ts:L10-L13

La route della lista unità si chiama `orgUnits` (al plurale). Tutte le altre route delle unità hanno il prefisso `orgUnit` — feature-libs/organization/administration/root/config/default-units-routing.config.ts:L20-L29. Su questo torneremo nelle trappole.

Per la meccanica di `paramsMapping` e della risoluzione `{ cxRoute, params } | cxUrl` vedi il cap. 14.

3.3 Il mapping CMS: un componente generico, provider specifici, child route

Il cuore dell'architettura è nella CMS config di ogni entità. Per il budget, `ManageBudgetsListComponent` è mappato sul generico `ListComponent`. Il mapping sostituisce `ListService` e `ItemService` con `useExisting` verso `BudgetListService` e `BudgetItemService` — feature-libs/organization/administration/components/budget/budget.config.ts:L29-L42. Usa poi tre proprietà di `CmsComponentMapping` (`providers`, `childRoutes`, `guards`) — core-libs/core/src/cms/config/cms-config.ts:L60-L77:

- `childRoutes.parent.data.cxPageMeta`: il breadcrumb `orgBudget.breadcrumbs.list` e il resolver `BudgetRoutePageMetaResolver`;
- `childRoutes.children`: `create` → `BudgetFormComponent`, `:budgetCode` → `BudgetDetailsComponent` con i figli `edit` e `cost-centers` — feature-libs/organization/administration/components/budget/budget.config.ts:L43-L77;
- `guards: [AuthGuard, AdminGuard]` — feature-libs/organization/administration/components/budget/budget.config.ts:L78-L78.

Le sotto-relazioni "assign" seguono un pattern fisso: una route (es. `budgets` sotto un cost center) con due figli, `''` per la lista degli assegnati e `'assign'` per la lista degli assegnabili — feature-libs/organization/administration/components/cost-center/cost-center.config.ts:L70-L83. Le unità sono l'eccezione principale. Usano un `UnitListComponent` specifico (l'albero) e applicano `canActivate: [OrgUnitGuard]` a `create`, `edit` e `children`, e `[UserGuard]` a `users/create` e `users/:userCode/roles` — feature-libs/organization/administration/components/unit/units.config.ts:L53-L138. Gli utenti applicano `UserGuard` a `create`, `edit` e `change-password` — feature-libs/organization/administration/components/user/user.config.ts:L66-L85.

3.4 Dal CMS al router: guardie dei componenti e child route a runtime

Alla prima apertura di `/organization/budgets`, il guard della pagina CMS fa tre cose in ordine — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L99:

1. **Risolve i mapping** dei componenti della pagina (`determineMappings`). È qui che il feature lazy `organizationAdministration` viene caricato, perché la pagina contiene `ManageBudgetsListComponent`.
2. **Esegue le guardie dei componenti**. `CmsGuardsService.cmsPageCanActivate` raccoglie le classi da tutti i mapping (deduplicate con un `Set`), le istanzia attraverso `UnifiedInjector` e le compone con `GuardsComposer` — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48, core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261. Passare dallo `UnifiedInjector` è ciò che permette di trovare `AdminGuard`, che non è `providedIn: 'root'` ma vive nel child injector del modulo lazy (sez. 3.10).
3. **Se l'esito è** `true` e la route non ha già `cxCmsRouteContext`, chiama `CmsRoutesImplService.handleCmsRoutesInGuard`.

`handleCmsRoutesInGuard` controlla se esiste già una route per la label della pagina. Se non esiste, legge le `childRoutes` dei componenti e, quando ci sono figli, chiama `updateRouting`. Poi rinaviga all'URL corrente con `navigateByUrl` e ritorna `false` per annullare la navigazione in corso — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L91. `updateRouting` agisce solo per **content page** con label che inizia con `/` (es. `/organization/budgets`). Crea una route `path: label.substring(1)` con `PageLayoutComponent` e i figli (le guardie avvolte ricorsivamente), unisce `parent.data` con `cxCmsRouteContext` e fa `router.resetConfig([newRoute, ...config])` — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L93-L124. La seconda navigazione trova `cxCmsRouteContext` nei data della route e salta questo passo — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L88-L88.

Da questo momento `organization/budgets/create`, `organization/budgets/:budgetCode`, `.../edit`, `.../cost-centers` sono route Angular "vere". Vengono rese nei `<router-outlet>` annidati della lista e delle card (sez. 3.6).

3.5 Le guardie di accesso

`AdminGuard` attende da `UserAccountFacade.get()` un utente non vuoto, legge `roles` e richiede `B2BUserRole.ADMIN` (`'b2badmingroup'`). Senza il ruolo naviga alla route `organization` e mostra un warning `organization.notification.noSufficientPermissions`. Un commento lo dichiara una soluzione temporanea, in attesa dei

permessi lato backend — feature-libs/organization/administration/core/guards/admin.guard.ts:L27-L49. I valori dei ruoli B2B sono b2badmingroup, b2bcustomergroup, b2bmanagergroup, b2bapprovergroup, e il diritto UNITORDERVIEWER vale unitorderviewergroup — core-libs/core/src/model/org-unit.model.ts:L18-L27.

OrgUnitGuard e UserGuard delegano a OrgUnitService.isUpdatingUnitAllowed() e B2BUserService.isUpdatingUserAllowed(). Se ricevono false mostrano un warning e ritornano un UrlTree verso organization — feature-libs/organization/administration/core/guards/org-unit.guard.ts:L25-L37. Nel core entrambi i metodi ritornano sempre true: sono hook da sovrascrivere, ad esempio nelle integrazioni — feature-libs/organization/administration/core/services/org-unit.service.ts:L501-L503, feature-libs/organization/administration/core/services/b2b-user.service.ts:L414-L416.

3.6 La lista: `ListComponent` + `ListService`

ListComponent (cx-org-list) inietta ListService e ItemService, che il mapping CMS ha risolto nelle implementazioni del budget, e attiva gli stili della feature a11yNavigationChevronContrast — feature-libs/organization/administration/components/shared/list/list.component.ts:L109-L114. Il viewType del servizio diventa la classe host — feature-libs/organization/administration/components/shared/list/list.component.ts:L116-L117.

ListService<T, P> è astratto — feature-libs/organization/administration/components/shared/list/list.service.ts:L42-L51:

- **struttura tabella**: il default è VERTICAL_STACKED, e VERTICAL da lg in su. getStructure() delega a TableService.buildStructure(viewType, defaultTableStructure) — feature-libs/organization/administration/components/shared/list/list.service.ts:L137-L142. Quest'ultimo usa la TableConfig se esiste per quel tipo, altrimenti costruisce una struttura "random" — core-libs/storefront/shared/components/table/table.service.ts:L49-L58;
- **paginazione**: un BehaviorSubject che parte da pageSize: 10. La config è iniettata come OrganizationUIConfig — feature-libs/organization/administration/components/shared/list/list.service.ts:L91-L97;
- **getData()**: unisce pagination$ con l'eventuale options.pagination della struttura, chiama load() ed emette prima i **ghost data** con startWith — feature-libs/organization/administration/components/shared/list/list.service.ts:L117-L129. I ghost data sono un array di 10 elementi vuoti che serve allo skeleton di caricamento — feature-libs/organization/administration/components/shared/list/list.service.ts:L53-L59. hasGhostData li riconosce per identità — feature-libs/organization/administration/components/shared/list/list.service.ts:L147-L170;
- **navigazione**: view() emette una nuova paginazione con currentPage, sort() torna a pagina 0 — feature-libs/organization/administration/components/shared/list/list.service.ts:L147-L170;
- **chiave**: key() vale 'code' di default — feature-libs/organization/administration/components/shared/list/list.service.ts:L103-L105. Gli utenti la ridefiniscono in customerId — feature-libs/organization/administration/components/user/services/user-list.service.ts:L52-L95;
- **viewType/domainType**: tableType (astratto, con il TODO "rename to viewType") è esposto come viewType. domainType vale _domainType se impostato, altrimenti viewType — feature-libs/organization/administration/components/shared/list/list.service.ts:L63-L83. viewType sceglie la table config, domainType è la root i18n delle celle;
- **bottone di creazione**: di default è un link (CreateButtonType.LINK) con label organization.add, e onCreateButtonClick è vuoto — feature-libs/organization/administration/components/shared/list/list.service.ts:L20-L23, feature-libs/organization/administration/components/shared/list/list.service.ts:L184-L198;

- **ricerca**: disabilitata di default. Il minimo di caratteri viene da `organizationUI.listSearch.minCharacters`, con fallback 3. `search()` aggiunge `query` e azzera `currentPage` — feature-libs/organization/administration/components/shared/list/list.service.ts:L204-L242.

Il template della lista sta dentro `cx-split-view [hideMode]="false"`. Il titolo è `viewType + '.header'` e un `<router-outlet>` in fondo ospita le split view annidate delle child route — feature-libs/organization/administration/components/shared/list/list.component.html:L1-L21, feature-libs/organization/administration/components/shared/list/list.component.html:L208-L211. Il link "create" punta a `{ cxRoute: structure.type + 'Create' }` ed è disattivato finché ci sono ghost data — feature-libs/organization/administration/components/shared/list/list.component.html:L128-L140. La tabella evidenzia l'item corrente confrontando `key` con `currentKey$` e usa `domainType` come `i18nRoot` — feature-libs/organization/administration/components/shared/list/list.component.html:L165-L173.

Due dettagli di UX nel componente:

- la ricerca ha `debounceTime(300)`, accetta query lunghe almeno il minimo (o vuote) e scarta le query invariate — feature-libs/organization/administration/components/shared/list/list.component.ts:L84-L98;
- `listData$` tiene in cache i valori dell'ultimo caricamento reale, per mostrare una tabella semitrasparente durante i caricamenti successivi invece dello skeleton — feature-libs/organization/administration/components/shared/list/list.component.ts:L159-L187.

Il click su una riga va a `launchItem`, che delega a `ItemService.launchDetails` — feature-libs/organization/administration/components/shared/list/list.component.ts:L212-L226.

3.7 L'item corrente: `CurrentItemService` + `ItemService`

`CurrentItemService<T>` ricava la **chiave** dell'item dai parametri di route (inclusi quelli delle child route), con la chiave di parametro fornita dalla sottoclasse (`getParamKey()`) e `distinctUntilChanged` — feature-libs/organization/administration/components/shared/current-item.service.ts:L18-L33. `item$` carica l'item con `getItem(code)` solo se la chiave c'è, altrimenti emette `undefined` — feature-libs/organization/administration/components/shared/current-item.service.ts:L40-L42. `b2bUnit$` legge sempre `ROUTE_PARAMS.unitCode`, qualunque sia l'entità — feature-libs/organization/administration/components/shared/current-item.service.ts:L47-L50. Le sottoclassi implementano `getParamKey`, `getItem` e `getError` — feature-libs/organization/administration/components/shared/current-item.service.ts:L56-L69. Per il budget: parametro `budgetCode`, item da `BudgetService.get`, errore da `getErrorState` — feature-libs/organization/administration/components/budget/services/current-budget.service.ts:L20-L42.

`ItemService<T>` ("Provides CRUD operations for all organization entities") riespone `key$`, `current$`, `unit$` ed `error$` da `CurrentItemService` — feature-libs/organization/administration/components/shared/item.service.ts:L17-L45. Aggiunge:

- `save(form, key?)`: con form invalido marca tutto come touched, aggiorna ricorsivamente la validità e ritorna `EMPTY`. Con form valido **salva** `form.value` **prima di** `form.disable()`, perché la disabilitazione in alcuni casi lo svuota, poi chiama `update(key, value)` se c'è una key, altrimenti `create(value)` — feature-libs/organization/administration/components/shared/item.service.ts:L47-L65;
- i metodi astratti `load`, `create` (protected), `update` e `getDetailsRoute` (protected). `delete` è opzionale — feature-libs/organization/administration/components/shared/item.service.ts:L70-L92;
- `launchDetails(item)`: naviga con `RoutingService.go({ cxRoute, params })` solo se la route esiste e l'item ha almeno una proprietà. I `params` sono di default l'item stesso (`buildRouteParams`) — feature-libs/organization/administration/components/shared/item.service.ts:L101-L120;
- `setEditMode()`, che alimenta il `BehaviorSubject` `isInEditMode$` — feature-libs/organization/administration/components/shared/item.service.ts:L126-L131.

`BudgetItemService.load` forza un nuovo caricamento a ogni chiamata ("to ensure accurate data") e poi ritorna `get(code)` — feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L32-L41. `update`/`create` fanno dispatch e ritornano `getLoadingStatus(value.code ?? '')`, cioè lo stato dell'entità identificata dal codice **nel valore inviato** e non dalla key di route — feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L43-L54. La route di dettaglio è `orgBudgetDetails`, anche se il JSDoc dice ancora `'budgetDetails'` — feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L56-L62.

Il dettaglio (`BudgetDetailsComponent`) costruisce `model$` in `ngOnInit` da `key$ → itemService.load(code)` con `startWith({})` — feature-libs/organization/administration/components/budget/details/budget-details.component.ts:L48-L59. Il template disabilita "edit" se il budget non è attivo o se si è già in edit mode. Include `cx-org-toggle-status` e `cx-org-disable-info` e linka la sotto-lista `orgBudgetCostCenters` — feature-libs/organization/administration/components/budget/details/budget-details.component.html:L6-L17, feature-libs/organization/administration/components/budget/details/budget-details.component.html:L81-L89.

3.8 Il form: `FormService` + `FormComponent`

`FormService<T>` costruisce il form **una sola volta** (`build`) e poi lo riusa. Se il form esiste e c'è un item, fa solo `patchData`. Altrimenti esegue `reset()`, `enable()` e patch, perché "while we should be able to reset with initial value, this doesn't always work" — feature-libs/organization/administration/components/shared/form/form.service.ts:L20-L37. `patchData` disabilita il form se `active === false`, lo riabilita se `active === true` e applica `{...defaultValue, ...item}` — feature-libs/organization/administration/components/shared/form/form.service.ts:L39-L51.

`BudgetFormService.build` crea questi controlli — feature-libs/organization/administration/components/budget/form/budget-form.service.ts:L26-L80:

- `code`: obbligatorio, senza caratteri speciali;
- `name`;
- `startDate` / `endDate`, nel formato del date picker;
- `budget`, che deve essere positivo;
- i sotto-gruppi `currency.isocode` e `orgUnit.uid`;
- un validatore `dateRange` sull'intero gruppo.

Il nome genera il codice al `blur` tramite `createCodeWithName` — feature-libs/organization/administration/components/budget/form/budget-form.component.html:L9-L15. La generazione avviene solo se il codice è vuoto: spazi → `-`, tutto in minuscolo — feature-libs/organization/administration/components/shared/utility/entity-code.ts:L9-L17. `BudgetFormComponent` fornisce localmente `ItemService → BudgetItemService` e `CurrentItemService → CurrentBudgetService`, preseleziona unità e valuta se ce n'è una sola e in `ngOnInit` chiama `unitService.loadList()` — feature-libs/organization/administration/components/budget/form/budget-form.component.ts:L42-L95.

`FormComponent` (`cx-org-form`) deriva `form$` da `itemService.current$`. Senza item passa `{}` a `getForm` ("we trick the form builder...") — feature-libs/organization/administration/components/shared/form/form.component.ts:L65-L75. `disabled$` osserva `statusChanges` e vale `true` sullo stato `'DISABLED'` — feature-libs/organization/administration/components/shared/form/form.component.ts:L28-L83. La chiave i18n è `i18nRoot + '.edit'` oppure `'.create'`. `ngOnInit`/`ngOnDestroy` accendono e spengono l'edit mode — feature-libs/organization/administration/components/shared/form/form.component.ts:L125-L142. `save()` prende la key corrente, chiama `itemService.save` e su `LoadStatus.SUCCESS` va al dettaglio e notifica `${i18nRoot}.messages.create|update`. **In ogni caso** riabilita il form — feature-libs/organization/administration/components/shared/form/form.component.ts:L90-L123. Il template avvolge una `cx-org-card` con

`cxOrgItemActive`. Il bottone Save è disabilitato da `form.disabled || disabled$`, e il Cancel sfrutta il soft-close della split view — `feature-libs/organization/administration/components/shared/form/form.component.html:L1-L30`.

3.9 Card, messaggi, toggle-status, sotto-liste

Card. `CardComponent` (`cx-org-card`) fornisce **a livello di componente** il proprio `MessageService` e legge l'item da `itemService.current$` — `feature-libs/organization/administration/components/shared/card/card.component.ts:L29-L59`. Quando cambia l'item svuota i messaggi, confrontando `code`, `uid` o `customerId` — `feature-libs/organization/administration/components/shared/card/card.component.ts:L88-L98`. `closeView` chiude a mano la view (`view.toggle(true)`) e dopo 500 ms simula il click sul parent. Senza questo accorgimento la chiusura del router outlet sarebbe immediata e taglierebbe l'animazione — `feature-libs/organization/administration/components/shared/card/card.component.ts:L68-L82`. Il template ha tre slot di proiezione (`[actions]`, `[info]`, `[main]`), un `cx-org-message` con `aria-live="assertive"` e un `<router-outlet>` dopo la view per il livello successivo della split view — `feature-libs/organization/administration/components/shared/card/card.component.html:L36-L63`.

Poiché il `MessageService` è nei `providers` (non nei `viewProviders`), è visibile anche al contenuto proiettato. In Angular i `providers` di un componente sono accessibili al template **e** al contenuto `ng-content`, mentre i `viewProviders` arrivano solo al template — `angular:adev/src/content/guide/di/debugging-and-troubleshooting-di.md:L279-L287`, `angular:packages/core/src/render3/di.ts:L685-L704`. Per questo `cx-org-toggle-status`, proiettato in `[actions]` dal template di dettaglio, scrive nella stessa area messaggi della card.

Messaggi. `MessageService` è `@Injectable()` senza `providedIn`. `add()` unisce i default, crea un `Subject` di eventi e lo restituisce. I messaggi info senza timeout ricevono 3000 ms — `feature-libs/organization/administration/components/shared/message/services/message.service.ts:L12-L51`. `MessageComponent` crea il componente del messaggio in un `ViewContainerRef` figlio, con un injector che fornisce `MessageData` — `feature-libs/organization/administration/components/shared/message/message.component.ts:L30-L58`. La chiusura avviene in due fasi: prima la classe `terminated` per l'animazione CSS, poi `destroy()` dopo 500 ms — `feature-libs/organization/administration/components/shared/message/message.component.ts:L68-L79`. L'auto-hide gira solo nel browser — `feature-libs/organization/administration/components/shared/message/base-message.component.ts:L68-L75`.

Toggle status. La disattivazione chiede conferma con un `ConfirmationMessageComponent` nel `MessageService` della card, l'attivazione no — `feature-libs/organization/administration/components/shared/detail/toggle-status-action/toggle-status.component.ts:L70-L105`. `confirm()` emette `{ confirm: true }` sugli eventi del messaggio — `feature-libs/organization/administration/components/shared/message/confirmation/confirmation-message.component.ts:L55-L63`. L'aggiornamento invia a `itemService.update` solo un patch `{ [key]: valore, active: !active }` — `feature-libs/organization/administration/components/shared/detail/toggle-status-action/toggle-status.component.ts:L118-L138`. Se l'input `disabled` è valorizzato, è lui a decidere (anche con `false`). Altrimenti il toggle è bloccato quando il parent è disabilitato o l'item è la root unit — `feature-libs/organization/administration/components/shared/detail/toggle-status-action/toggle-status.component.ts:L110-L116`. La root unit si riconosce per esclusione: ha `uid` e `name`, non ha `orgUnit`/`unit`, e non ha `parentOrgUnit` oppure ce l'ha uguale al proprio `uid` — `feature-libs/organization/administration/components/shared/detail/disable-info/disable-info.service.ts:L19-L34`. Quando l'item corrente passa a `active === false`, `ItemActiveDirective` mostra `organization.notification.disabled` — `feature-libs/organization/administration/components/shared/item-active.directive.ts:L26-L47`. `ItemExistsDirective` mostra `organization.notification.notExist` quando `error$` emette `true` — `feature-libs/organization/administration/components/shared/item-exists.directive.ts:L24-L37`.

Solo `UserGroupItemService` e `UnitAddressItemService` implementano `delete`. Le altre entità si "eliminano" disattivandole — `feature-libs/organization/administration/components/user-group/services/user-group-`

item.service.ts:L45-L47, feature-libs/organization/administration/components/unit/links/addresses/services/unit-address-item.service.ts:L74-L74. DeleteItemComponent chiede conferma e chiama l'opzionale itemService.delete?.(...) — feature-libs/organization/administration/components/shared/detail/delete-item-action/delete-item.component.ts:L97-L117.

Sotto-liste. SubListService estende ListService. Usa il solo layout VERTICAL, ghost data di 3 righe, e assign/unassign opzionali che di default ritornano EMPTY (con un TODO "abstract") — feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts:L18-L44. filterSelected filtra lato client perché "we can't filter with the backend API" — feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts:L46-L63. SubListComponent estende ListComponent e ricarica con service.getData(key) a ogni cambio di key — feature-libs/organization/administration/components/shared/sub-list/sub-list.component.ts:L43-L69. Il suo template rende una cx-org-card con i18nRoot = viewType e mostra la paginazione solo con totalPages > 1 — feature-libs/organization/administration/components/shared/sub-list/sub-list.component.html:L1-L5, feature-libs/organization/administration/components/shared/sub-list/sub-list.component.html:L23-L35. AssignCellComponent.toggleAssign chiama assign/unassign(key, link) e notifica ${viewType}.assigned|unassigned. Il link è code ?? customerId ?? uid del contesto della riga, perché "the object do not have a normalized shape" — feature-libs/organization/administration/components/shared/sub-list/assign-cell.component.ts:L50-L109.

Celle. CellComponent legge il contesto dall'outlet. Con il toggle a11yCardNotificationMessage si sottoscrive a outlet.context$ e chiama markForCheck — feature-libs/organization/administration/components/shared/table/cell.component.ts:L31-L47. Una cella è linkabile solo se la proprietà è definita e cellOptions.linkable è true. La route generata è _type + 'Details' — feature-libs/organization/administration/components/shared/table/cell.component.ts:L69-L87. Le sottoclassi sono piccole: ActiveLinkCellComponent cambia solo tabIndex — feature-libs/organization/administration/components/shared/table/active-link/active-link-cell.component.ts:L19-L23; AmountCellComponent mostra budget + ' ' + currency — feature-libs/organization/administration/components/shared/table/amount/amount-cell.component.ts:L19-L33.

3.10 Core: NgRx, servizi-facade, esito delle operazioni

Store. OrganizationStoreModule registra la feature NgRx organization con reducer da token, metaReducers ed effects — feature-libs/organization/administration/core/store/organization-store.module.ts:L14-L23. Lo stato ha sei slice (budget, orgUnit, userGroup, permission, costCenter, b2bUser) — feature-libs/organization/administration/core/store/organization-state.ts:L103-L114. Ognuna è un combineReducers di entityLoaderReducer (entities, list e le liste di relazione) — feature-libs/organization/administration/core/store/reducers/index.ts:L94-L105. Il meta-reducer clearOrganizationState azzera **l'intero** stato organization su CLEAR_ORGANIZATION_DATA, AuthActions.LOGOUT e SiteContextActions.LANGUAGE_CHANGE — feature-libs/organization/administration/core/store/reducers/index.ts:L209-L227.

Azioni ed effects. Le azioni budget estendono le StateUtils.Entity*Action. LoadBudgets usa come id dell'entità lista la serializeSearchConfig(params) — feature-libs/organization/administration/core/store/actions/budget.action.ts:L64-L75. LoadBudgetSuccess usa i code come id — feature-libs/organization/administration/core/store/actions/budget.action.ts:L51-L62. UpdateBudget marca in loading payload.budget.code, non payload.budgetCode — feature-libs/organization/administration/core/store/actions/budget.action.ts:L132-L140. loadBudgets$ normalizza la pagina con StateUtils.normalizeListPage(budgets, 'code') ed emette sia le entità sia la pagina — feature-libs/organization/administration/core/store/effects/budget.effect.ts:L58-L72. createBudget$ e updateBudget$ emettono OrganizationClearData sia in successo sia in errore — feature-libs/organization/administration/core/store/effects/budget.effect.ts:L94-L109, feature-libs/organization/administration/core/store/effects/budget.effect.ts:L122-L139. Gli errori passano da tryNormalizeHttpError — feature-libs/organization/administration/core/store/effects/budget.effect.ts:L37-L44.

Servizi. `BudgetService` è `providedIn: 'root'`. I comandi ottengono lo userId con `takeUserId(true)` e fanno dispatch. Il ramo d'errore è vuoto, con un TODO — feature-libs/organization/administration/core/services/budget.service.ts:L28-L45. Le query usano due schemi diversi:

- `get(code)` usa `using()`: una sottoscrizione collaterale a `loading$` (con `auditTime(0)`) innesca `loadBudget` se lo stato non è né loading né success né error. Il valore emesso è `getBudgetValue` filtrato sui valori truthy — feature-libs/organization/administration/core/services/budget.service.ts:L63-L89;
- `getList(params)` usa `observeOn(queueScheduler)`, innesca il load se serve ed emette solo a stato success o error — feature-libs/organization/administration/core/services/budget.service.ts:L91-L104.

`getCostCenters(code)` non fa chiamate dedicate: legge i `costCenters` annidati nel budget — feature-libs/organization/administration/core/services/budget.service.ts:L106-L115.

Esito. `getItemStatus` osserva lo stato dell'entità con `pairwise()` e produce un risultato solo per la transizione **da** `loading` **a uno stato successivo**: `SUCCESS`, `ERROR` o `null` — feature-libs/organization/administration/core/utils/get-item-status.ts:L15-L31. `LoadStatus` è un enum numerico (`SUCCESS = 0`, `ERROR = 1`) — feature-libs/organization/administration/core/model/organization-item-status.ts:L7-L15. È questo meccanismo che collega un dispatch "fire-and-forget" alla UI che aspetta un esito.

Nel create l'ordine delle azioni è `CreateBudgetSuccess` seguito da `OrganizationClearData` — feature-libs/organization/administration/core/store/effects/budget.effect.ts:L96-L99. `getItemStatus` vede quindi prima la coppia (loading, success) ed emette `SUCCESS`. Il reset successivo produce una coppia con stato precedente non-loading, che viene filtrata. `FormComponent` comunque prende solo il primo esito (`take(1)`) — feature-libs/organization/administration/components/shared/form/form.component.ts:L95-L96.

Selettori, connector, guardie. I selector partono da `createFeatureSelector('organization')`. La lista viene denormalizzata con `StateUtils.denormalizeSearch` — feature-libs/organization/administration/core/store/selectors/budget.selector.ts:L51-L59, feature-libs/organization/administration/core/store/selectors/feature.selector.ts:L14-L17. `BudgetConnector` delega 1:1 all'astratto `BudgetAdapter` — feature-libs/organization/administration/core/connectors/budget/budget.connector.ts:L13-L41, feature-libs/organization/administration/core/connectors/budget/budget.adapter.ts:L11-L34. Le guardie sono fornite da `OrganizationsGuardsModule` nel child injector del modulo lazy, apposta per permetterne l'override da altre librerie o dall'app — feature-libs/organization/administration/core/guards/organization-guards.module.ts:L12-L21. Il core aumenta inoltre via module augmentation i tipi `B2BUnit`, `B2BUser` e `CostCenter` di `@spartacus/core` — feature-libs/organization/administration/core/model/augmented-core.model.ts:L9-L33.

Errori HTTP. `OrganizationConflictHandler` gestisce i 409 con errori `AlreadyExistsError`: confronta il messaggio con regex per budget, user, userGroup, unit e cost center e mostra `organization.httpHandlers.conflict.<key>` — feature-libs/organization/administration/core/http-interceptors/conflict/conflict.handler.ts:L20-L92. Il suo `hasMatch` richiede sia lo status sia una regex corrispondente, così le altre risposte 409 non vengono intercettate — feature-libs/organization/administration/core/http-interceptors/conflict/conflict.handler.ts:L29-L31. `OrganizationBadRequestHandler` fa lo stesso con i 400 `ModelSavingError`/`DuplicateUidError` — feature-libs/organization/administration/core/http-interceptors/bad-request/bad-request.handler.ts:L20-L29, feature-libs/organization/administration/core/http-interceptors/bad-request/bad-request.handler.ts:L84-L89. Per la catena `HttpErrorHandler` vedi il cap. 41.

Page meta. `OrganizationPageMetaResolver` si applica a `CompanyPageTemplate` (content page) — feature-libs/organization/administration/core/services/organization-page-meta.module.ts:L11-L20, feature-libs/organization/administration/core/services/organization-page-meta.resolver.ts:L41-L50. Inserisce il breadcrumb "Organization" dopo "home", tranne quando la route corrente è proprio `organization` — feature-libs/organization/administration/core/services/organization-page-meta.resolver.ts:L99-L130.

Il layout di `CompanyPageTemplate` ha il solo slot `BodyContent` — `feature-libs/organization/administration/root/config/default-organization-layout.config.ts:L9-L15`.

3.11 OCC

Gli endpoint budget sono `/users/${userId}/budgets` e `/users/${userId}/budgets/${budgetCode}`. Quelli dei cost center non contengono lo `userId` (`/costcenters`, `/costcentersall`) — `feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L13-L53`. `OccBudgetAdapter` usa questi metodi HTTP — `feature-libs/organization/administration/occ/adapters/occ-budget.adapter.ts:L33-L73`:

- GET per il singolo budget e per la lista;
- POST su `budgets` per create;
- PATCH su `budget` per update;
- il serializer `BUDGET_SERIALIZER` e il normalizer `BUDGET_NORMALIZER` per le conversioni.

La `SearchConfig`, compresa l'eventuale `query` della ricerca, finisce nei `queryParams` di `buildUrl` — `feature-libs/organization/administration/occ/adapters/occ-budget.adapter.ts:L81-L86`. Il normalizer converte `startDate`/`endDate` da datetime a date. Il serializer fa l'inverso e porta `endDate` a fine giornata — `feature-libs/organization/administration/occ/converters/occ-budget-normalizer.ts:L20-L26`, `feature-libs/organization/administration/occ/converters/occ-budget-serializer.ts:L22-L28`. `OccCostCenterAdapter.loadList` usa `costCentersAll`, mentre `create` usa `costCenters`. `assignBudget` fa POST con `budgetCode` come query param e body `null` — `feature-libs/organization/administration/occ/adapters/occ-cost-center.adapter.ts:L41-L92`.

Il core di Spartacus ha già i converter OCC dei cost center in `CostCenterOccModule`, che aggiunge anche l'endpoint `getActiveCostCenters`. `AdministrationOccModule` lo importa — `core-libs/core/src/occ/adapters/cost-center/cost-center-occ.module.ts:L20-L40`, `core-libs/core/src/occ/adapters/cost-center/default-occ-cost-centers-config.ts:L9-L18`.

3.12 Le altre entità: dove divergono dal budget

- **Unità (albero).** `UnitListService` costruisce la lista da `OrgUnitService.getTree()`, inizializza `UnitTreeService` con la key corrente e appiattisce l'albero nei nodi espansi. La ricerca è client-side su `id`/`name` (toggle `enableB2BUnitSearch`) — `feature-libs/organization/administration/components/unit/services/unit-list.service.ts:L50-L95`. `filterTree` tiene il nodo che corrisponde con tutti i figli. Per gli antenati di un match tiene solo i rami corrispondenti e li forza espansi — `feature-libs/organization/administration/components/unit/services/unit-list.service.ts:L97-L142`. I nodi ricevono `count`, `expanded`, `depthLevel` e `uid = id` (commento "tmp, should be normalized") — `feature-libs/organization/administration/components/unit/services/unit-list.service.ts:L155-L189`. L'espansione combina il toggle globale, quello del singolo nodo e `minimalExpanded = 1` — `feature-libs/organization/administration/components/unit/services/unit-tree.service.ts:L26-L91`.
- **Utenti.** La key è `customerId`. I ruoli vengono filtrati su quelli noti e la ricerca dipende da `enableB2BCustomerSearch` — `feature-libs/organization/administration/components/user/services/user-list.service.ts:L52-L95`. I ruoli disponibili sono una lista fissa di quattro, "not driven by the backend (lack of API)" — `feature-libs/organization/administration/core/services/b2b-user.service.ts:L349-L364`. Alla creazione di un utente il customerId non è noto. L'effect naviga quindi direttamente a `orgUserDetails` (workaround dichiarato), tranne quando il contesto di route è `/organization/units`. Emette poi due `CreateB2BUserSuccess`, uno con i dati reali e uno con l'id temporaneo — `feature-libs/organization/administration/core/store/effects/b2b-user.effect.ts:L85-L113`, `feature-libs/organization/administration/core/store/effects/b2b-user.effect.ts:L641-L648`. Creando un utente da un'unità, `UnitUserItemService.create` imposta `customerId = 'new'` come id temporaneo da tracciare —

feature-libs/organization/administration/components/unit/links/users/create/unit-user-item.service.ts:L49-L60.

- **Cost center.** La ricerca dipende da `enableB2BCostCenterSearch` — feature-libs/organization/administration/components/cost-center/services/cost-center-list.service.ts:L55-L57. L'assegnazione dei budget osserva lo stato di loading del **budget**, perché le azioni `AssignBudget` marcano l'entità `BUDGET_ENTITIES` — feature-libs/organization/administration/components/cost-center/budgets/cost-center-budget-list.service.ts:L46-L64, feature-libs/organization/administration/core/store/actions/cost-center.action.ts:L242-L254.

I tre toggle di ricerca sono `true` nei `defaultFeatureToggles` — core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L252-L265, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1044-L1046.

3.13 Account summary

`AccountSummaryRootModule` registra quattro cose — feature-libs/organization/account-summary/root/account-summary-root.module.ts:L37-L59:

- una route `path: null` (con `@ts-ignore`) per `cxRoute: 'orgAccountSummaryDetails'`, protetta da `CmsPageGuard`;
- la routing config;
- i CMS component lazy;
- un `BlobErrorInterceptor` in `HTTP_INTERCEPTORS`.

Le route sono `organization/account-summary` e `organization/account-summary/details/:orgUnit` — feature-libs/organization/account-summary/root/config/default-account-summary-routing.config.ts:L9-L23. `AccountSummaryFacade` è una facade lazy con `getAccountSummary`, `getDocumentList` e `getDocumentAttachment` — feature-libs/organization/account-summary/root/facade/account-summary.facade.ts:L17-L30. `AccountSummaryService` memorizza `userId` e `orgUnitId` in campi aggiornati da sottoscrizioni nel costruttore, e restituisce le chiamate al connector con `shareReplay(1)` — feature-libs/organization/account-summary/core/facade/account-summary.service.ts:L27-L82.

La lista **riusa il framework administration**. `AccountSummaryModule` importa `AdministrationModule` — feature-libs/organization/account-summary/account-summary.module.ts:L13-L21. La CMS config sostituisce `ListService`/`ItemService` con le versioni account summary e dichiara la child route `:unitCode` — feature-libs/organization/account-summary/components/list/account-summary-list.config.ts:L47-L84. `OrganizationTableType` viene esteso a runtime con `ACCOUNT_SUMMARY_UNIT` — feature-libs/organization/account-summary/components/model/augmented.model.ts:L10-L16. `AccountSummaryUnitListService` estende `UnitListService` cambiando solo `tableType` — feature-libs/organization/account-summary/components/services/account-summary-unit-list.service.ts:L15-L17.

Il download degli allegati usa `responseType: 'blob'`. Per questo `BlobErrorInterceptor` riconverte in JSON i corpi d'errore `Blob` (solo nel browser) — feature-libs/organization/account-summary/occ/adapters/occ-account-summary.adapter.ts:L71-L90, feature-libs/organization/account-summary/root/http-interceptors/blob-error.interceptor.ts:L31-L55. Il componente scarica con `URL.createObjectURL` + `FileDownloadService.download` — feature-libs/organization/account-summary/components/details/document/account-summary-document.component.ts:L133-L142.

3.14 Order approval, unit order, user registration

Order approval. Il root dichiara 7 CMS component lazy con dipendenza `ORDER_FEATURE`, una route `path: null` per `orderApprovalDetails` con `AuthGuard, CmsPageGuard`, e le route `my-account/approval-dashboard` e `my-`

account/approval/:approvalCode — feature-libs/organization/order-approval/root/order-approval-root.module.ts:L21-L72. Non ci sono entry point /core e /occ: il public API principale esporta tutto — feature-libs/organization/order-approval/public_api.ts:L7-L16. OrderApprovalService non ha una facade astratta. Usa lo store NgRx e il process state orderApproval.makeDecision — feature-libs/organization/order-approval/core/services/order-approval.service.ts:L116-L143, feature-libs/organization/order-approval/core/store/order-approval-state.ts:L10-L14. L'effect makeDecision$ ignora gli utenti anonimi e, su successo, ricarica l'approvazione — feature-libs/organization/order-approval/core/store/effects/order-approval.effect.ts:L97-L114. Il dettaglio riusa i componenti di @spartacus/order sostituendo OrderDetailsService con OrderApprovalDetailService — feature-libs/organization/order-approval/components/details/order-approval-details.module.ts:L45-L94. Il servizio legge approvalCode dallo stato del router e chiama loadOrderApproval a ogni emissione — feature-libs/organization/order-approval/components/details/order-approval-detail.service.ts:L19-L32. ApproverGuard ammette approver e admin — feature-libs/organization/order-approval/core/guards/approver.guard.ts:L29-L52.

Unit order. Il root dichiara 4 CMS component lazy e due route path: null. Quella di dettaglio ha un cxContext ORDER_ENTRIES_CONTEXT — feature-libs/organization/unit-order/root/unit-order-root.module.ts:L23-L82. UnitOrderFacade usa facadeFactory con async: true — feature-libs/organization/unit-order/root/facade/unit-order.facade.ts:L13-L33. Con async il resolver aggiunge un delay(0), "required to make lazy NgRx store feature ready" — core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L36, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57. Lo stato è la feature NgRx 'unit order' (con uno spazio nel nome) — feature-libs/organization/unit-order/core/store/unit-order-state.ts:L10-L21. UnitLevelOrdersViewerGuard ammette unitorderviewergroup o b2badmingroup — feature-libs/organization/unit-order/core/guards/unit-level-orders-viewer.guard.ts:L30-L54.

User registration. Il root dichiara solo 3 CMS component lazy, senza routing config — feature-libs/organization/user-registration/root/user-registration-root.module.ts:L11-L34. UserRegistrationService implementa la facade con un Command — feature-libs/organization/user-registration/core/facade/user-registration.service.ts:L16-L41. L'adapter fa POST su /orgUsers con l'header USE_CLIENT_TOKEN — feature-libs/organization/user-registration/occ/adapters/occ-user-registration.adapter.ts:L37-L57. Per il client token vedi il cap. 22. I dati aziendali non vengono inviati come campi strutturati: finiscono nel message tramite la chiave i18n userRegistrationForm.messageToApproverTemplate — feature-libs/organization/user-registration/components/form/user-registration-form.service.ts:L144-L200. Dopo l'invio si reindirizza al login solo con ResourceOwnerPasswordFlow o con il toggle authorizationCodeFlowByDefault — feature-libs/organization/user-registration/components/form/user-registration-form.service.ts:L177-L206. I componenti sono protetti da NotAuthGuard — feature-libs/organization/user-registration/components/verification-token-form/verification-token-form.module.ts:L51-L70. Un HttpErrorHandler dedicato gestisce i 409 — feature-libs/organization/user-registration/core/user-registration-core.module.ts:L14-L28.

3.15 Stili, schematic e test

feature-libs/organization/_index.scss definisce il mixin visible-focus e importa gli stili di administration, order-approval, user-registration, account-summary e unit-order (feature-libs/organization/_index.scss:L3-L15). Lo schematic addSpartacusOrganization segue la sequenza standard analyzeApplication → addFeatures → addPackageJsonDependenciesForLibrary → finalizeInstallation (feature-libs/organization/schematics/add-organization/index.ts:L25-L45); lo spec ha un blocco per ciascuna delle cinque feature (feature-libs/organization/schematics/add-organization/index_spec.ts:L126-L459) e per administration verifica con snapshot anche il modulo di configurazione B2B generato (feature-libs/organization/schematics/add-organization/index_spec.ts:L162-L170). feature-libs/organization/vitest.config.ts mappa numerosi path core-libs/... di helper di test (pagination, split view, url pipe, mock transla-

tion) sui sorgenti (feature-libs/organization/vitest.config.ts:L16-L34) e limita il pool a maxWorkers: 4 (feature-libs/organization/vitest.config.ts:L37-L37).

---

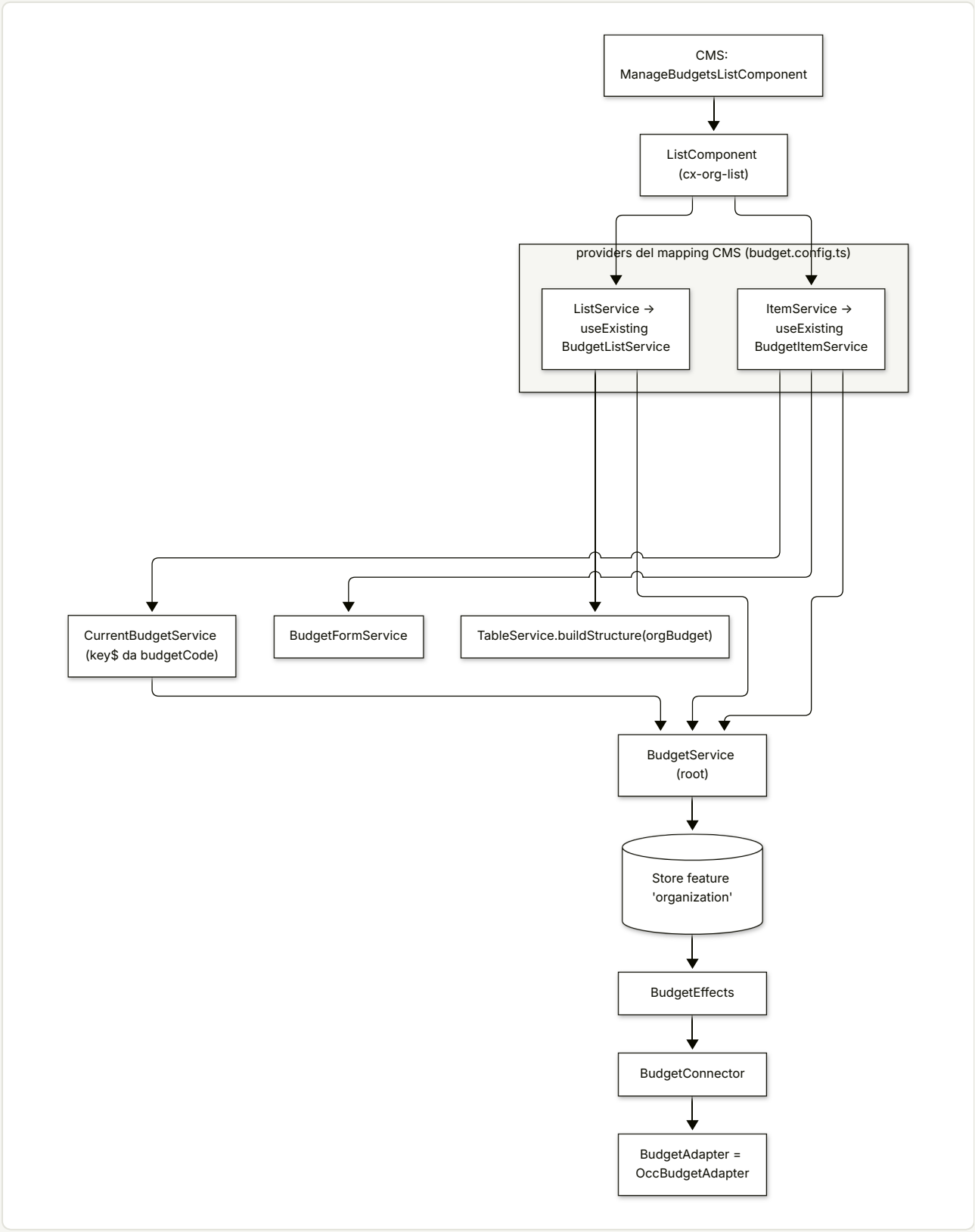
4. Diagrammi

4.1 Prima apertura di /organization/budgets: lazy loading, guardie, child route

■ TAVOLA 28.1 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: sez. 3.4 (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L70, core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L99, core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48, core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L124) e sez. 3.5.

4.2 Composizione DI: un `ListComponent`, sei entità



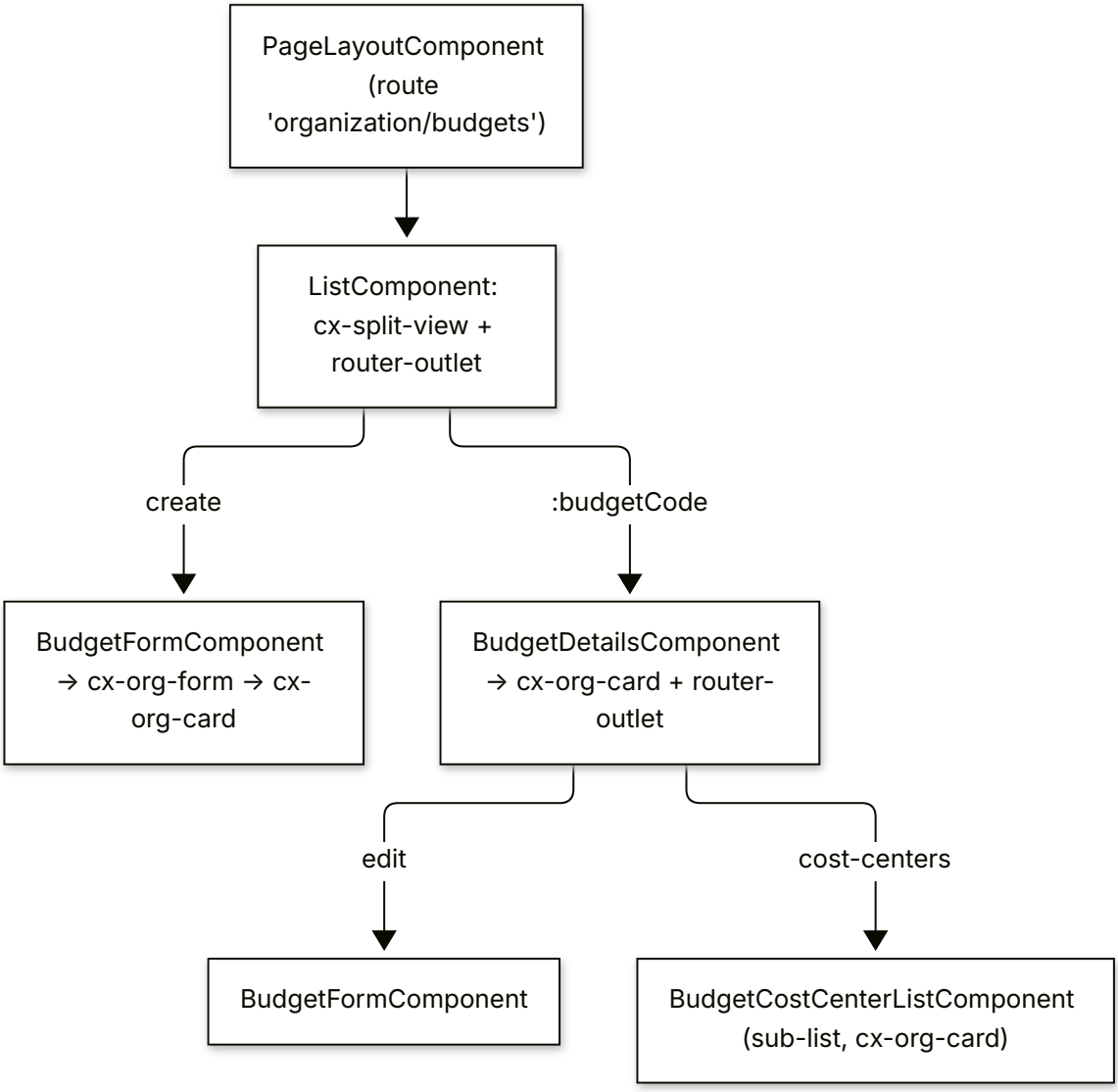
Riferimenti: `feature-libs/organization/administration/components/budget/budget.config.ts:L29-L42`, `feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L22-L30`, `feature-libs/organization/administration/components/budget/services/budget-list.service.ts:L30-L38`, `feature-libs/organization/administration/occ/administration-occ.module.ts:L58-L80`.

4.3 Salvataggio del form budget (create)

■ TAVOLA 28.2 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: sez. 3.7, 3.8, 3.10 (feature-libs/organization/administration/components/shared/item.service.ts:L47-L65, feature-libs/organization/administration/core/store/effects/budget.effect.ts:L86-L112, feature-libs/organization/administration/core/utils/get-item-status.ts:L15-L31). Nel ramo `EMPTY` la sottoscrizione di `FormComponent.save` non riceve mai un valore, quindi `form.enable()` non viene chiamato. Non serve, perché il form invalido non era stato disabilitato.

4.4 Split view: livelli di router-outlet



Riferimenti: feature-libs/organization/administration/components/budget/budget.config.ts:L52-L76, feature-libs/organization/administration/components/shared/list/list.component.html:L208-L211, feature-libs/organization/administration/components/shared/card/card.component.html:L36-L63.

5. Estratti di codice reale

5.1 Il mapping CMS del budget

feature-libs/organization/administration/components/budget/budget.config.ts:L29-L81

```ts
export const budgetCmsConfig: CmsConfig = {
  cmsComponents: {
    ManageBudgetsListComponent: {
      component: ListComponent,
      providers: [
        {
          provide: ListService,
          useExisting: BudgetListService,
        },
        {
          provide: ItemService,
          useExisting: BudgetItemService,
        },
      ],
      childRoutes: {
        parent: {
          data: {
            cxPageMeta: {
              breadcrumb: 'orgBudget.breadcrumbs.list',
              resolver: BudgetRoutePageMetaResolver,
            },
          },
        },
        children: [
          {
            path: 'create',
            component: BudgetFormComponent,
          },
          {
            path: `:${ROUTE_PARAMS.budgetCode}`,
            component: BudgetDetailsComponent,
            // ...
            children: [
              {
                path: `edit`,
                component: BudgetFormComponent,
              },
              {
                path: 'cost-centers',
                component: BudgetCostCenterListComponent,
              },
            ],
```

```
          },
        ],
      },
      guards: [AuthGuard, AdminGuard],
    },
  },
};
```

5.2 Child route aggiunte a runtime

core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L93-L124

```ts
  private updateRouting(
    pageContext: PageContext,
    pageLabel: string,
    childRoutesConfig: CmsComponentChildRoutesConfig
  ): boolean {
    if (
      pageContext.type === PageType.CONTENT_PAGE &&
      pageLabel.startsWith('/') &&
      pageLabel.length > 1
    ) {
      const children = this.wrapCmsGuardsRecursively(
        childRoutesConfig.children ?? []
      );

      const newRoute: CmsRoute = {
        path: pageLabel.substring(1),
        component: PageLayoutComponent,
        children: children,
        data: deepMerge({}, childRoutesConfig?.parent?.data ?? {}, {
          cxCmsRouteContext: {
            type: pageContext.type,
            id: pageLabel,
          },
        }),
      };

      this.router.resetConfig([newRoute, ...this.router.config]);
      return true;
    }

    return false;
  }
```

5.3 `CurrentItemService`: la chiave viene dalla route

feature-libs/organization/administration/components/shared/current-item.service.ts:L30-L42

```ts
readonly key$: Observable<string> = this.routingService.getParams().pipe(
  map((params) => params[this.getParamKey()]),
  distinctUntilChanged()
);

/**
 * Observes the active item.
 *
 * The active item is loaded by the active `key$`.
 */
readonly item$: Observable<T | undefined> = this.key$.pipe(
  switchMap((code: string) => (code ? this.getItem(code) : of(undefined)))
);
```

5.4 ItemService.save

feature-libs/organization/administration/components/shared/item.service.ts:L47-L65

```ts
save(
  form: UntypedFormGroup,
  key?: string
): Observable<OrganizationItemStatus<T>> {
  if (form.invalid) {
    form.markAllAsTouched();
    FormUtils.deepUpdateValueAndValidity(form);
    return EMPTY;
  } else {
    /**
     * This assignment is needed to re-use form value after `form.disable()` call
     * In some cases value was converted by `form.disable()` into empty object
     */
    const formValue = form.value;
    form.disable();

    return key ? this.update(key, formValue) : this.create(formValue);
  }
}
```

5.5 FormComponent.save

feature-libs/organization/administration/components/shared/form/form.component.ts:L90-L112

```ts
save(form: UntypedFormGroup): void {
  this.itemService.key$
    .pipe(
      first(),
      switchMap((key) =>
```

```
        this.itemService.save(form, key).pipe(
          take(1),
          map((data) => ({
            item: data.item,
            status: data.status,
            action: key ? 'update' : 'create',
          }))
        )
      )
    )
    .subscribe(({ item, action, status }) => {
      if (status === LoadStatus.SUCCESS) {
        this.itemService.launchDetails(item);
        this.notify(item, action);
      }
      form.enable();
    });
}
```

5.6 `ListService.getData`: ghost data prima di ogni load

feature-libs/organization/administration/components/shared/list/list.service.ts:L117-L129

TS

```
getData(...args: any): Observable<EntitiesModel<T> | undefined> {
  return this.pagination$.pipe(
    // we merge any configured pagination from the table structure
    switchMap((pagination) =>
      this.getStructure().pipe(
        map((config) => ({ ...pagination, ...config.options?.pagination }))
      )
    ),
    switchMap((pagination) =>
      this.load(pagination, ...args).pipe(startWith(this.ghostData))
    )
  );
}
```

5.7 `BudgetService.get` e `getList`: load implicito

feature-libs/organization/administration/core/services/budget.service.ts:L75-L104

TS

```
get(budgetCode: string): Observable<Budget> {
  const loading$ = this.getBudgetState(budgetCode).pipe(
    auditTime(0),
    tap((state) => {
      if (!(state.loading || state.success || state.error)) {
        this.loadBudget(budgetCode);
```

```
        }
      })
    );

    return using(
      () => loading$.subscribe(),
      () => this.getBudgetValue(budgetCode)
    );
  }

  getList(params: SearchConfig): Observable<EntitiesModel<Budget> | undefined> {
    return this.getBudgetList(params).pipe(
      observeOn(queueScheduler),
      tap((process: StateUtils.LoaderState<EntitiesModel<Budget>>) => {
        if (!(process.loading || process.success || process.error)) {
          this.loadBudgets(params);
        }
      }),
      filter((process: StateUtils.LoaderState<EntitiesModel<Budget>>) =>
        Boolean(process.success || process.error)
      ),
      map((result) => result.value)
    );
  }
```

5.8 getItemStatus

feature-libs/organization/administration/core/utils/get-item-status.ts:L15-L31

TS

```
export function getItemStatus<T>(
  itemState: Observable<StateUtils.LoaderState<T>>
): Observable<OrganizationItemStatus<T>> {
  return itemState.pipe(
    observeOn(queueScheduler),
    pairwise(),
    filter(([previousState]) => previousState.loading ?? false),
    map(([_previousState, currentState]) => ({
      status: currentState.success
        ? LoadStatus.SUCCESS
        : currentState.error
          ? LoadStatus.ERROR
          : null,
      item: currentState.value,
    }))
  );
}
```

5.9 Create effect e meta-reducer

feature-libs/organization/administration/core/store/effects/budget.effect.ts:L94-L109

```ts
      switchMap((payload) =>
        this.budgetConnector.create(payload.userId, payload.budget).pipe(
          switchMap((data) => [
            new BudgetActions.CreateBudgetSuccess(data),
            new OrganizationActions.OrganizationClearData(),
          ]),
          catchError((error: HttpErrorResponse) =>
            from([
              new BudgetActions.CreateBudgetFail({
                budgetCode: payload.budget.code ?? '',
                error: tryNormalizeHttpError(error, this.logger),
              }),
              new OrganizationActions.OrganizationClearData(),
            ])
          )
        )
```

feature-libs/organization/administration/core/store/reducers/index.ts:L209-L227

```ts
export function clearOrganizationState(
  reducer: ActionReducer<OrganizationState, Action>
): ActionReducer<OrganizationState, Action> {
  return function (state, action) {
    if (action.type === OrganizationActions.CLEAR_ORGANIZATION_DATA) {
      state = undefined;
    }
    if (action.type === AuthActions.LOGOUT) {
      state = undefined;
    }
    if (action.type === SiteContextActions.LANGUAGE_CHANGE) {
      state = undefined;
    }

    return reducer(state, action);
  };
}

export const metaReducers: MetaReducer<any>[] = [clearOrganizationState];
```

5.10 Guardie nel child injector

feature-libs/organization/administration/core/guards/organization-guards.module.ts:L12-L21

```ts
@NgModule({
  // To ensure effective over-providing from within the child injector of a lazy-loaded module
  // by other extension libraries or end-user customizations, it's essential to provide guards manually
  // in the child injector and not in the root injector (providedIn: 'root').
  //
  // If guards were provided in the root injector, the only place to over-provide them or their dependencies
  // would be the root injector (root module), which would break lazy loading for those guards and their dependencies.
  providers: [AdminGuard, UserGuard, OrgUnitGuard],
})
export class OrganizationsGuardsModule {}
```

5.11 Toggle status: conferma solo in disattivazione, patch minimo

feature-libs/organization/administration/components/shared/detail/toggle-status-action/toggle-status.component.ts:L118-L138

```ts
  protected update(item: T): void {
    this.itemService
      .update(
        item[this.key as keyof BaseItem] as string,
        this.getPatchedItem(item)
      )
      .pipe(
        take(1),
        filter((data) => data.status === LoadStatus.SUCCESS)
      )
      .subscribe((data) => this.notify({ ...item, ...data.item }));
  }

  protected getPatchedItem(item: T): T {
    const patch: BaseItem = {};

    Object.assign(patch, { [this.key]: item[this.key as keyof T] });

    patch.active = !item.active;
    return patch as T;
  }
```

5.12 Estendere un enum a runtime (account summary)

feature-libs/organization/account-summary/components/model/augmented.model.ts:L10-L16

```ts
declare module '@spartacus/organization/administration/components' {
  enum OrganizationTableType {
    ACCOUNT_SUMMARY_UNIT = 'orgAccountSummary',
```

```
  }
}

(OrganizationTableType as any)['ACCOUNT_SUMMARY_UNIT'] = 'orgAccountSummary';
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`organizationUI.listSearch.minCharacters`	number	3	`feature-libs/organization/administration/root/config/default-organization-ui.config.ts:L9-L15` (tipo `feature-libs/organization/administration/root/config/organization-ui.config.ts:L17-L30`)	`feature-libs/organization/administration/components/shared/list/list.service.ts:L221-L223`
`routing.routes.organization`	RouteConfig	`paths: ['organization']`	`feature-libs/organization/administration/root/administration-root.module.ts:L49-L51`	`feature-libs/organization/administration/core/guards/admin.guard.ts:L38-L38`, `feature-libs/organization/administration/core/guards/org-unit.guard.ts:L32-L34`
`routing.routes.orgBudget*` (5 route)	RoutingConfig	`organization/budgets[...]`, `budgetCode → code`	`feature-libs/organization/administration/root/config/default-budget-routing.config.ts:L15-L38`	`feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L60-L62`, `feature-libs/organization/administration/components/shared/list/list.component.html:L135-L135`, `feature-libs/organization/administration/components/shared/table/cell.component.ts:L85-L87`
`routing.routes.orgCostCenter*`, `orgPurchaseLimit*`, `orgUnit*`, `orgUser*`, `orgUserGroup*`	RoutingConfig	vedi file	`feature-libs/organization/administration/root/config/default-units-routing.config.ts:L10-L29` e analoghi	gli `ItemService` delle entità
`routing.routes.orgAccountSummary`, `orgAccountSummaryDetails`	RoutingConfig	`organization/account-summary[/details/:orgUnit]`	`feature-libs/organization/account-summary/root/config/default-account-summary-routing.config.ts:L11-L23`	`feature-libs/organization/account-summary/root/account-summary-root.module.ts:L45-L45`
`routing.routes.orderApprovals`, `orderApprovalDetails`	RoutingConfig	`my-account/approval-dashboard`, `my-account/approval/:approvalCode`	`feature-libs/organization/order-approval/root/order-approval-root.module.ts:L58-L69`	route `path: null` `feature-libs/organization/order-approval/root/order-approval-root.module.ts:L44-L51`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`routing.routes.unitLevelOrders`, `unitLevelOrderDetail`	RoutingConfig	`my-account/unitLevelOrders`, `my-account/unitLevelOrderDetails/:orderCode`	`feature-libs/organization/unit-order/root/unit-order-root.module.ts:L67-L79`	route `path: null` `feature-libs/organization/unit-order/root/unit-order-root.module.ts:L43-L62`
`layoutSlots.CompanyPageTemplate.slots`	string[]	`['BodyContent']`	`feature-libs/organization/administration/root/config/default-organization-layout.config.ts:L9-L15`	layout storefront (cap. 16)
`featureModules.organizationAdministration.cmsComponents`	string[]	6 `Manage*ListComponent`	`feature-libs/organization/administration/root/administration-root.module.ts:L25-L42`	caricamento feature CMS (cap. 12)
`featureModules.organizationAccountSummary.cmsComponents`	string[]	3 componenti	`feature-libs/organization/account-summary/root/account-summary-root.module.ts:L21-L35`	idem
`featureModules.organizationOrderApproval`	cmsComponents + `dependencies: [ORDER_FEATURE]`	7 componenti	`feature-libs/organization/order-approval/root/order-approval-root.module.ts:L21-L40`	idem
`featureModules.organizationUnitOrder`	cmsComponents + `dependencies: [ORDER_FEATURE]`	4 componenti	`feature-libs/organization/unit-order/root/unit-order-root.module.ts:L23-L39`	idem
`featureModules.organizationUserRegistration.cmsComponents`	string[]	3 componenti	`feature-libs/organization/user-registration/root/user-registration-root.module.ts:L11-L25`	idem
`cmsComponents.Manage*ListComponent` (`component`, `providers`, `childRoutes`, `guards`)	CmsComponentMapping	per entità	`feature-libs/organization/administration/components/budget/budget.config.ts:L29-L81` e analoghi	`core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L124`, `core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48`
`table.<OrganizationTableType>`	TableConfig	budget: celle `name`, `active`, `amount`, `dateRange`, `unit`	`feature-libs/organization/administration/components/budget/budget.config.ts:L87-L127`	`feature-libs/organization/administration/components/shared/list/list.service.ts:L137-L142`
`table.<type>.options.pagination.pageSize`	number	10 (ListService); `MAX_OCC_INTEGER_VALUE` per le sotto-liste "assigned"	`feature-libs/organization/administration/components/shared/list/list.service.ts:L91-L93`, `feature-libs/organization/administration/components/budget/budget.config.ts:L121-L123`	`feature-libs/organization/administration/components/shared/list/list.service.ts:L120-L124`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.endpoints.budgets, budget, costCenters*, orgUnits*, userGroups*, permissions*, b2bUsers*	OccEndpoint	vedi file	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L9-L75	adapter in feature-libs/organization/administration/occ/adapters/occ-budget.adapter.ts:L81-L86 e analoghi
backend.occ.endpoints.getActiveCostCenters	string	/costcenters?fields=DEFAULT,unit(BASIC,addresses(DEFAULT))	core-libs/core/src/occ/adapters/cost-center/default-occ-cost-centers-config.ts:L9-L18	core-libs/core/src/occ/adapters/user/occ-user-cost-centers.adapter.ts:L51-L54, consumato da core-libs/core/src/user/facade/user-cost-center.service.ts:L49-L49
backend.occ.endpoints.accountSummary*	string	users/${userId}/orgUnits/${orgUnitId}/...	feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L10-L15	feature-libs/organization/account-summary/occ/adapters/occ-account-summary.adapter.ts:L71-L90
backend.occ.endpoints.orderApproval*	string	/users/${userId}/orderapprovals[...]	feature-libs/organization/order-approval/occ/config/default-occ-organization-config.ts:L9-L21	adapter order approval
backend.occ.endpoints.unitLevelOrder*	string	/orgUsers/${userId}/orgUnits/orders[...]	feature-libs/organization/unit-order/occ/config/default-occ-organization-config.ts:L9-L18	adapter unit order
backend.occ.endpoints.organizationUserRegistration	string	/orgUsers	feature-libs/organization/user-registration/occ/config/default-occ-organization-config.ts:L9-L17	feature-libs/organization/user-registration/occ/adapters/occ-user-registration.adapter.ts:L59-L61
toggle enableB2BUnitSearch, enableB2BCostCenterSearch, enableB2BCustomerSearch	boolean	true	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1044-L1046	feature-libs/organization/administration/components/unit/services/unit-list.service.ts:L50-L52, feature-libs/organization/administration/components/cost-center/services/cost-center-list.service.ts:L55-L57, feature-libs/organization/administration/components/user/services/user-list.service.ts:L93-L95
toggle a11yCardNotificationMessage	boolean	true	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1040-L1040	feature-libs/organization/administration/components/shared/table/cell.component.ts:L39-L46
toggle a11yNavigationChevronContrast	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1109-L1109	feature-libs/organization/administration/components/shared/list/list.component.ts:L113-L113

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `authorizationCodeFlowByDefault`	boolean	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029`	`feature-libs/organization/user-registration/components/form/user-registration-form.service.ts:L181-L181`
chunk i18n `organization`	TranslationChunksConfig	29 namespace (`organization`, `orgBudget`, `orgUnit*`, …)	`feature-libs/organization/administration/assets/translations/translations.ts:L9-L41`	demo `projects/storefrontapp/src/app/spartacus/features/organization/organization-administration-feature.module.ts:L35-L45`

Tutti i valori di `OrganizationTableType`, che fanno da root i18n di liste e card, compaiono come namespace del chunk `organization`. Il chunk ne contiene anche altri (`organization`, `orgUnitAssignedRoles`, `orgUnitUserRoles`, `orgUnitAssignedUsers`) — `feature-libs/organization/administration/assets/translations/translations.ts:L9-L41`, `feature-libs/organization/administration/components/shared/organization.model.ts:L9-L50`. Per il caricamento dei chunk vedi il cap. 37.

---

7. Commenti e note notevoli dal sorgente

- "Provides CRUD operations for all organization entities. This base class simplifies the various entity implementation, and ensures a consistent component implementation." — `feature-libs/organization/administration/components/shared/item.service.ts:L17-L22`. È la dichiarazione d'intenti dell'intero framework.
- "This assignment is needed to re-use form value after `form.disable()` call..." — `feature-libs/organization/administration/components/shared/item.service.ts:L56-L59`. Spiega perché l'ordine "leggi valore, poi disabilita" non va invertito.
- "while we should be able to reset with initial value, this doesn't always work hence, we're patching afterwards." — `feature-libs/organization/administration/components/shared/form/form.service.ts:L30-L31`.
- "TODO: rename to `viewType`" — `feature-libs/organization/administration/components/shared/list/list.service.ts:L66-L66`. Nelle sottoclassi la proprietà resta `tableType`.
- "// TODO: abstract" su `assign` — `feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts:L34-L34`.
- "As we can't filter with the backend API, we do this client side." — `feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts:L46-L48`.
- "// TODO: this doesn't work with lazy loaded feature" sulla routing config dei gruppi utenti — `feature-libs/organization/administration/root/config/default-user-group-routing.config.ts:L15-L15`.
- "// TODO: Inline this factory when we start releasing Ivy compiled libraries" — `feature-libs/organization/administration/root/administration-root.module.ts:L24-L24`. Lo stesso commento compare nei root module di account-summary, order-approval e unit-order (non in user-registration) — `feature-libs/organization/account-summary/root/account-summary-root.module.ts:L20-L20`, `feature-libs/organization/order-approval/root/order-approval-root.module.ts:L20-L20`, `feature-libs/organization/unit-order/root/unit-order-root.module.ts:L22-L22`, `feature-libs/organization/user-registration/root/user-registration-root.module.ts:L10-L11`.
- "routing as temporary solution until /organization won't have set up proper permission on backend" — `feature-libs/organization/administration/core/guards/admin.guard.ts:L36-L37`.

- Il commento che motiva la registrazione delle guardie nel child injector — feature-libs/organization/administration/core/guards/organization-guards.module.ts:L13-L18.
- "// TODO: for future releases, refactor this part to thrown errors" (gli errori di takeUserId vengono ignorati) — feature-libs/organization/administration/core/services/budget.service.ts:L42-L42.
- "// TODO Workaround for not known customerId while user creation (redireciton)" — feature-libs/organization/administration/core/store/effects/b2b-user.effect.ts:L87-L87.
- "Note: No id or code is provided when creating a new user so we cannot store a value in the ngrx state..." — feature-libs/organization/administration/components/unit/links/users/create/unit-user-item.service.ts:L52-L56.
- @deprecated Use budgetTableConfig instead. su newBudgetTableConfig — feature-libs/organization/administration/components/budget/budget.config.ts:L129-L132.
- "This is not for the public API" / "TODO:#my-account-architecture - Number.MAX_VALUE?" su MAX_OCC_INTEGER_VALUE = 2147483647 — feature-libs/organization/administration/components/constants.ts:L7-L9.
- "The views are router based, which means if we close a view, the router outlet is cleaned immediately..." — feature-libs/organization/administration/components/shared/card/card.component.ts:L68-L72.
- "We use a child view container ref, as creating components will become siblings." — feature-libs/organization/administration/components/shared/message/message.component.ts:L30-L31.
- "// we don't want to run this logic when doing SSR" — feature-libs/organization/administration/components/shared/message/base-message.component.ts:L70-L70.
- Il blocco AUGMENTABLE_TYPES_START … AUGMENTABLE_TYPES_END attorno all'export di OrganizationTableType — feature-libs/organization/administration/components/public_api.ts:L16-L18.
- Il JSDoc Returns 'budgetDetails' contraddice il codice, che ritorna 'orgBudgetDetails' — feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L56-L62.
- "When enabled, the organization's table component will stop re-rendering its rows each data update..." (toggle a11yCardNotificationMessage) — core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L238-L242.

8. Extension point

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
ItemService<T>, CurrentItemService<T>	sottoclasse e sostituzione nei providers del mapping CMS (o nei providers del componente form)	feature-libs/organization/administration/components/shared/item.service.ts:L23-L132, feature-libs/organization/administration/components/shared/current-item.service.ts:L18-L70	Ricette CMS, outlet e layout
ListService<T,P>, SubListService<T>	sottoclasse (tableType, load, key, isSearchEnabled, getCreateButtonType, assign/unassign)	feature-libs/organization/administration/components/shared/list/list.service.ts:L20-L243, feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts:L18-L64	Ricette CMS, outlet e layout

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`FormService<T>`	sottoclasse (`build`, `defaultValue`)	`feature-libs/organization/administration/components/shared/form/form.service.ts:L11-L58`	Ricette CMS, outlet e layout
`CmsConfig.cmsComponents.Manage*ListComponent`	`provideConfig` con `component`, `providers`, `childRoutes`, `guards` sostitutivi	`feature-libs/organization/administration/components/budget/budget.config.ts:L29-L81`	Ricette CMS, outlet e layout
`TableConfig` per `OrganizationTableType`	`provideConfig({ table: { orgBudget: … } })` (celle, `dataComponent`, paginazione)	`feature-libs/organization/administration/components/budget/budget.config.ts:L87-L127`	Ricette CMS, outlet e layout
`CellComponent` e celle derivate	componente custom referenziato da `dataComponent`	`feature-libs/organization/administration/components/shared/table/cell.component.ts:L31-L111`	Ricette CMS, outlet e layout
`MessageService` / `MessageData.component`	messaggio custom che estende `BaseMessageComponent`	`feature-libs/organization/administration/components/shared/message/services/message.service.ts:L14-L56`, `feature-libs/organization/administration/components/shared/message/message.model.ts:L42-L47`	Ricette CMS, outlet e layout
`OrganizationTableType`	module augmentation + assegnazione runtime (vedi 5.12)	`feature-libs/organization/administration/components/shared/organization.model.ts:L9-L35`, `feature-libs/organization/administration/components/public_api.ts:L16-L18`	Upgrade-safe
Routing config delle entità, `ROUTE_PARAMS`	`provideConfig({ routing: { routes: { orgBudget: { paths: [...] } } } })`	`feature-libs/organization/administration/root/config/default-budget-routing.config.ts:L10-L38`, `feature-libs/organization/administration/root/route-params.ts:L10-L18`	Ricette config e DI
`OrganizationUIConfig`	`provideConfig({ organizationUI: { listSearch: { minCharacters } } })`	`feature-libs/organization/administration/root/config/organization-ui.config.ts:L13-L34`	Ricette config e DI
`AdminGuard`, `OrgUnitGuard`, `UserGuard`	override nel child injector del feature lazy (es. modulo importato nel wrapper accanto ad `AdministrationModule`)	`feature-libs/organization/administration/core/guards/organization-guards.module.ts:L12-L21`	Ricette config e DI
`OrgUnitService.isUpdatingUnitAllowed`, `B2BUserService.isUpdatingUserAllowed`	override del servizio per abilitare i guard	`feature-libs/organization/administration/core/services/org-unit.service.ts:L501-L503`, `feature-libs/organization/administration/core/services/b2b-user.service.ts:L414-L416`	Ricette config e DI

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`OrganizationConflictHandler`, `OrganizationBadRequestHandler`	nuovo `HttpErrorHandler` multi o sottoclasse con regex aggiuntive	feature-libs/organization/administration/core/http-interceptors/conflict/conflict.handler.ts:L20-L92	Ricette config e DI
`BudgetAdapter` & co., token `BUDGET_NORMALIZER/BUDGET_SERIALIZER`, `COST_CENTER_NORMALIZER`	adapter custom (`useClass`) o converter aggiuntivi (`multi`)	feature-libs/organization/administration/core/connectors/budget/budget.adapter.ts:L11-L34, feature-libs/organization/administration/core/connectors/budget/converters.ts:L11-L20, core-libs/core/src/cost-center/connectors/cost-center/converters.ts:L12-L22	Ricette OCC e dati
Endpoint OCC (`backend.occ.endpoints.*`)	`provideConfig` con endpoint sostitutivi	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L9-L75	Ricette OCC e dati
`OrganizationPageMetaResolver`	sottoclasse con `getScore` più alto (come fa `AccountSummaryPageMetaResolver`)	feature-libs/organization/administration/core/services/organization-page-meta.resolver.ts:L38-L131, feature-libs/organization/account-summary/core/account-summary-page-meta.resolver.ts:L31-L103	Ricette OCC e dati
`AccountSummaryFacade`, `UnitOrderFacade`, `UserRegistrationFacade`	facade lazy: override del servizio di implementazione nel feature	feature-libs/organization/account-summary/root/facade/account-summary.facade.ts:L17-L30, feature-libs/organization/unit-order/root/facade/unit-order.facade.ts:L13-L33, feature-libs/organization/user-registration/root/facade/user-registration.facade.ts:L13-L31	Ricette feature lazy
`OrderApprovalDetailService`	servizio sostitutivo di `OrderDetailsService` nei componenti order	feature-libs/organization/order-approval/components/details/order-approval-detail.service.ts:L15-L66	Ricette CMS, outlet e layout
Feature module lazy (wrapper)	aggiungere estensioni accanto ad `AdministrationModule`, come fa la demo con CDC	projects/storefrontapp/src/app/spartacus/features/organization/administration-wrapper.module.ts:L12-L20	Ricette feature lazy

Due note pratiche.

Override dei guard. Non serve ridichiarare il mapping CMS. Basta fornire la propria classe con `{ provide: AdminGuard, useClass: … }` in un modulo caricato **dentro** il feature lazy. Il guard viene istanziato tramite `UnifiedInjector` — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L41-L45 — e il commento in feature-libs/organization/administration/core/guards/organization-guards.module.ts:L13-L18 spiega che un override nel root injector romperebbe il lazy loading.

Nuova entità o vista. Il percorso più breve è quello dell'account summary:

- estendere `OrganizationTableType`;
- sottoclassare un `ListService` esistente cambiando `tableType`;
- dichiarare un CMS mapping con il componente lista adatto e i propri provider `ListService`/`ItemService`. L'account summary usa `AccountSummaryListComponent extends UnitListComponent`; per una lista piatta si usa `ListComponent` — feature-libs/organization/account-summary/components/list/account-summary-list.config.ts:L49-L51, feature-libs/organization/account-summary/components/list/account-summary-list.component.ts:L18-L18.

Riferimenti: feature-libs/organization/account-summary/components/services/account-summary-unit-list.service.ts:L15-L17, feature-libs/organization/account-summary/components/list/account-summary-list.config.ts:L47-L84.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `LoadStatus.SUCCESS` **vale** `0`**, cioè falsy.** `if (data.status)` non funziona. Il codice confronta sempre con `=== LoadStatus.SUCCESS` — feature-libs/organization/administration/core/model/organization-item-status.ts:L7-L10, feature-libs/organization/administration/components/shared/form/form.component.ts:L106-L106.
2. **Ogni create/update di budget azzera tutto lo stato** `organization`. Succede anche quando l'operazione fallisce, perché viene emesso `OrganizationClearData`. Liste, alberi e relazioni vengono ricaricati da zero — feature-libs/organization/administration/core/store/effects/budget.effect.ts:L96-L107, feature-libs/organization/administration/core/store/reducers/index.ts:L209-L227. Lo stesso reset avviene al cambio lingua — feature-libs/organization/administration/core/store/reducers/index.ts:L219-L221.
3. `getItemStatus` **vede solo la transizione.** Se ci si sottoscrive dopo che il loading è già finito, non arriva alcun esito. L'id osservato deve inoltre coincidere con l'id marcato dall'azione — feature-libs/organization/administration/core/utils/get-item-status.ts:L19-L21.
4. **Update con codice cambiato.** `BudgetItemService.update(code, value)` osserva `getLoadingStatus(value.code)`, e `UpdateBudget` marca `budget.code`. Se il codice nel form diverge dalla key di route, il loading viene tracciato sul nuovo codice — feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L43-L49, feature-libs/organization/administration/core/store/actions/budget.action.ts:L132-L140.
5. **Create utente.** `UserItemService.create` osserva `getLoadingStatus(value.uid ?? '')`, ma il form utente non ha un controllo `uid` (ha `customerId`, vuoto in creazione), quindi l'id osservato è `''` — feature-libs/organization/administration/components/user/services/user-item.service.ts:L45-L50, feature-libs/organization/administration/components/user/form/user-form.service.ts:L22-L24. `CreateB2BUser` marca come loading `customerId ?? null`, e `entityReducer` converte un id `null` in `''` — feature-libs/organization/administration/core/store/actions/b2b-user.action.ts:L128-L134, core-libs/core/src/state/utils/entity/entity.reducer.ts:L28-L33. Entrambi gli id collassano quindi sull'entità `''`. Il secondo `CreateB2BUserSuccess`, quello con il `customerId` temporaneo, chiude la transizione, e `FormComponent` riceve `SUCCESS` — feature-libs/organization/administration/core/store/effects/b2b-user.effect.ts:L92-L97. La navigazione principale al dettaglio la fa comunque l'effect (`redirectToDetails`) — feature-libs/organization/administration/core/store/effects/b2b-user.effect.ts:L88-L90. Lo skip del redirect dipende dalla stringa hardcoded `'/organization/units'` — feature-libs/organization/administration/core/store/effects/b2b-user.effect.ts:L641-L648.

6. **Un solo form per entità.** Il `FormService` sono `providedIn: 'root'`, quindi lo stato del form sopravvive alle navigazioni e viene resettato o patchato a ogni `getForm` — feature-libs/organization/administration/components/shared/form/form.service.ts:L20-L37, feature-libs/organization/administration/components/budget/form/budget-form.service.ts:L18-L21. `BudgetFormComponent` chiama `getForm()` senza item in un field initializer, `FormComponent.form$` lo richiama con l'item: le due chiamate condividono lo stesso `FormGroup` — feature-libs/organization/administration/components/budget/form/budget-form.component.ts:L67-L67, feature-libs/organization/administration/components/shared/form/form.component.ts:L65-L75.
7. **Entità disattivata = form disabilitato.** `patchData` disabilita il form se `active === false` — feature-libs/organization/administration/components/shared/form/form.service.ts:L39-L51. Visto che il form è unico (punto 6), aprire un'entità disattivata e poi una attiva riabilita il form solo perché `active === true` scatena il ramo `enable`.
8. `orgUnits` **vs** `orgUnit`**.** La route lista delle unità è `orgUnits`, ma `OrganizationTableType.UNIT` vale `'orgUnit'`. I link "create" (`structure.type + 'Create'` → `orgUnitCreate`) e le celle (`_type + 'Details'` → `orgUnitDetails`) funzionano perché producono nomi che esistono. Una route costruita come `type` senza suffisso non esisterebbe — feature-libs/organization/administration/root/config/default-units-routing.config.ts:L20-L29, feature-libs/organization/administration/components/shared/organization.model.ts:L15-L15, feature-libs/organization/administration/components/shared/list/list.component.html:L135-L135.
9. **Le child route esistono solo dopo la prima visita.** Vengono create da `handleCmsRoutesInGuard` solo per content page con label che inizia con `/` — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L98-L102. Il primo ingresso costa una navigazione annullata e una ripetuta (`navigateByUrl` + `return false`) — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L83-L88.
10. `AdminGuard` **può restare sospeso.** Filtra finché l'utente non è caricato (`Object.keys(user).length > 0`). Se `UserAccountFacade.get()` non emette mai un utente non vuoto, la navigazione non si risolve — feature-libs/organization/administration/core/guards/admin.guard.ts:L28-L29.
11. `OrgUnitGuard` **e** `UserGuard` **sono no-op** finché qualcuno non sovrascrive `isUpdating*Allowed` — feature-libs/organization/administration/core/services/org-unit.service.ts:L501-L503, feature-libs/organization/administration/core/services/b2b-user.service.ts:L414-L416.
12. **La sotto-lista "cost center del budget" filtra per** `active`, non per `selected` come il `filterSelected` di base. Mostra solo i cost center attivi annidati nel budget — feature-libs/organization/administration/components/budget/cost-centers/budget-cost-center-list.service.ts:L46-L56, feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts:L49-L63. La table config imposta `pageSize: MAX_OCC_INTEGER_VALUE` (2147483647), ma `load` ignora la paginazione: i cost center vengono dai dati annidati del budget (`getCostCenters`), senza una chiamata dedicata — feature-libs/organization/administration/components/budget/budget.config.ts:L121-L123, feature-libs/organization/administration/components/budget/cost-centers/budget-cost-center-list.service.ts:L33-L40.
13. `cachedValues` **si aggiorna anche con lista vuota.** La condizione `length >= 0` è sempre vera — feature-libs/organization/administration/components/shared/list/list.component.ts:L178-L181.
14. **Il** `MessageService` **della card è condiviso.** Sta nei `providers`, quindi i componenti proiettati (toggle-status, delete-item) e `cxOrgItemActive` sull'host usano la stessa istanza — feature-libs/organization/administration/components/shared/card/card.component.ts:L34-L34, angular:adev/src/content/guide/di/debugging-and-troubleshooting-di.md:L279-L287. Spostarlo in `viewProviders` in una sottoclasse romperebbe i messaggi delle azioni proiettate.
15. `@ViewChild(MessageService)` **in** `SubListComponent` **è lecito.** Sembra strano, ma `@ViewChild` accetta un provider definito nell'albero dei componenti figli. Qui trova quello della `cx-org-card` nel template — feature-

libs/organization/administration/components/shared/sub-list/sub-list.component.ts:L46-L47, angular:packages/core/src/metadata/di.ts:L431-L436.

16. **Testi e nomi da sapere.** "assign"/"unassign" sono hardcoded nel template della cella e non tradotti — feature-libs/organization/administration/components/shared/sub-list/assign-cell.component.ts:L28-L32. La table config dei permessi si chiama `permisionTableConfig` (con il refuso) — feature-libs/organization/administration/components/permission/permission.config.ts:L79-L79. `CostCenterService.get` è tipato `Observable<Budget>` — feature-libs/organization/administration/core/services/cost-center.service.ts:L87-L101.
17. **API Angular deprecate.** `MessageRenderService` usa `ComponentFactoryResolver`/`ComponentFactory` — feature-libs/organization/administration/components/shared/message/services/message-render.service.ts:L7-L26.
18. **Endpoint senza slash iniziale.** `accountSummary*` e `unitLevelOrderDetail` non hanno lo slash iniziale, gli altri sì — feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L10-L15, feature-libs/organization/unit-order/occ/config/default-occ-organization-config.ts:L13-L14. È innocuo: `urlPathJoin` rimuove gli slash ai bordi di ogni parte e conserva solo quello iniziale della prima, il base URL — core-libs/core/src/occ/utils/occ-url-util.ts:L12-L33, core-libs/core/src/occ/services/occ-endpoints.service.ts:L233-L238.
19. **Campi asincroni in** `AccountSummaryService`**.** `userId` e `orgUnitId` vengono letti da campi impostati in modo asincrono nel costruttore. Se un metodo viene chiamato prima delle emissioni, i valori sono `undefined`. `getAccountSummary` usa `??`, gli altri due metodi usano `||` — feature-libs/organization/account-summary/core/facade/account-summary.service.ts:L32-L73. Il download crea un object URL senza `revokeObjectURL` nel codice letto — feature-libs/organization/account-summary/components/details/document/account-summary-document.component.ts:L133-L142. Il resolver usa l'etichetta hardcoded `'Account Summaries'` — feature-libs/organization/account-summary/core/account-summary-page-meta.resolver.ts:L42-L42.
20. **Load ripetuti nell'order approval.** `approvalCode$` deriva da `getRouterState()` senza `distinctUntilChanged`, quindi ogni emissione dello stato del router con un codice valorizzato rilancia `loadOrderApproval`, anche se il codice non cambia. Il fallback di `get()` (load se lo stato è vuoto) di solito non scatta, perché il dispatch del `tap` avviene prima della sottoscrizione a `get()` — feature-libs/organization/order-approval/components/details/order-approval-detail.service.ts:L19-L30, feature-libs/organization/order-approval/core/services/order-approval.service.ts:L74-L85.
21. `CostCenterModule.forRoot()` **di core è vuoto** e non risulta importato nelle librerie — core-libs/core/src/cost-center/cost-center.module.ts:L9-L17. I converter OCC dei cost center sono altrove, in `CostCenterOccModule` — core-libs/core/src/occ/adapters/cost-center/cost-center-occ.module.ts:L20-L40.

---

10. Come fare debug

- **"La lista non compare / redirect a organization".** Verifica i ruoli dell'utente: `AdminGuard` richiede `b2badmingroup` e reindirizza con il warning `organization.notification.noSufficientPermissions` — feature-libs/organization/administration/core/guards/admin.guard.ts:L31-L46. Se la pagina resta bianca senza redirect, controlla che `UserAccountFacade.get()` emetta un utente non vuoto (trappola 10).
- **Child route non attive (404 su** `/organization/budgets/xyz`**).** Ispeziona `router.config` dopo la prima visita: deve comparire una route `organization/budgets` con `data.cxCmsRouteContext` — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L107-L119. Se manca, la pagina CMS non è una

content page con label `/organization/budgets`, oppure il mapping del componente non ha `childRoutes` (sez. 3.4).

- **Store.** In Redux DevTools apri lo slice `organization.budget`: ci sono `entities` (chiave `code`) e `list` (chiave = `serializeSearchConfig(params)`) — feature-libs/organization/administration/core/store/reducers/index.ts:L94-L105, feature-libs/organization/administration/core/store/actions/budget.action.ts:L64-L75. Dopo un salvataggio deve comparire l'azione `'[Organization] Clear Data'` (`CLEAR_ORGANIZATION_DATA`) e lo stato tornare vuoto — feature-libs/organization/administration/core/store/actions/organization.action.ts:L9-L9, feature-libs/organization/administration/core/store/reducers/index.ts:L213-L215.
- **Il form non naviga dopo Save.** Controlla in DevTools che l'entità con id = `value.code` passi da `loading: true` a `success: true`. `getItemStatus` reagisce solo a quella transizione — feature-libs/organization/administration/core/utils/get-item-status.ts:L19-L28. Se l'azione marcasse un id diverso da quello osservato, non arriverebbe alcun `SUCCESS`.
- **Il form resta disabilitato.** `save()` disabilita il form prima della chiamata — feature-libs/organization/administration/components/shared/item.service.ts:L60-L61. `FormComponent` lo riabilita solo quando arriva un esito — feature-libs/organization/administration/components/shared/form/form.component.ts:L105-L110. Se `getItemStatus` non emette mai, il form resta disabilitato. Un'altra causa è un item con `active === false` (trappola 7).
- **Errore 409/400 senza messaggio specifico.** Gli handler intervengono solo se il messaggio corrisponde alle loro regex — feature-libs/organization/administration/core/http-interceptors/conflict/conflict.handler.ts:L29-L31. Altrimenti decide la catena generica (cap. 41).
- **Tabella "random" o colonne sbagliate.** `TableService.buildStructure` ricade sulla struttura random quando manca una `table.<viewType>` — core-libs/storefront/shared/components/table/table.service.ts:L49-L58. Verifica che `tableType` del tuo `ListService` coincida con la chiave della table config.
- **Breakpoint utili:**
  - `ListService.getData` (feature-libs/organization/administration/components/shared/list/list.service.ts:L117-L129);
  - `ItemService.save` (feature-libs/organization/administration/components/shared/item.service.ts:L47-L65);
  - `CmsRoutesImplService.handleCmsRoutesInGuard` (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L69-L91);
  - `CmsGuardsService.cmsPageCanActivate` (core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48).
- **Rete.** Una lista budget produce un GET su `/users/{userId}/budgets` con `pageSize`, `currentPage`, `sort` ed eventuale `query` come query string — feature-libs/organization/administration/occ/adapters/occ-budget.adapter.ts:L81-L86. Le date in uscita sono datetime con `endDate` a fine giornata — feature-libs/organization/administration/occ/converters/occ-budget-serializer.ts:L22-L28.

---

11. Domande di autoverifica

1. **Perché esiste un solo `ListComponent` per sei entità, e come sa quali dati caricare?** Il mapping CMS di ogni `Manage*ListComponent` fornisce `ListService` e `ItemService` con `useExisting` verso le implementazioni dell'entità. Il componente inietta le astrazioni — feature-libs/organization/administration/components/budget/budget.config.ts:L31-L42, feature-libs/organization/administration/components/shared/list/list.component.ts:L109-L114.

2. **In che ordine il guard CMS carica il feature, esegue** `AdminGuard` **e crea le child route?** Prima `determineMappings` (lazy loading), poi `cmsPageCanActivate` (guardie). Solo se l'esito è `true` e manca `cxCmsRouteContext` viene chiamato `handleCmsRoutesInGuard` — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L99.
3. **Perché** `AdminGuard` **non è** `providedIn: 'root'`**, e come viene trovato?** È fornito nel child injector del modulo lazy per permetterne l'override senza rompere il lazy loading — feature-libs/organization/administration/core/guards/organization-guards.module.ts:L13-L19. Viene risolto tramite `UnifiedInjector` — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L41-L45.
4. **Da dove viene la "chiave" dell'item corrente in un pannello di dettaglio?** Dai parametri di route, con la chiave restituita da `getParamKey()` (es. `budgetCode`) — feature-libs/organization/administration/components/shared/current-item.service.ts:L30-L33, feature-libs/organization/administration/components/budget/services/current-budget.service.ts:L32-L33.
5. **Come fa la UI a sapere che un create è riuscito, se** `BudgetService.create` **ritorna** `void`**?** `getLoadingStatus` usa `getItemStatus`, che con `pairwise()` intercetta la transizione da `loading` a `success`/`error` dell'entità — feature-libs/organization/administration/core/services/budget.service.ts:L145-L149, feature-libs/organization/administration/core/utils/get-item-status.ts:L15-L31.
6. **Perché** `if (status)` **sarebbe un bug?** `LoadStatus.SUCCESS` vale `0` — feature-libs/organization/administration/core/model/organization-item-status.ts:L7-L10.
7. **Cosa succede allo stato NgRx dopo un update di budget, anche fallito?** L'effect emette `OrganizationClearData`, e il meta-reducer imposta `state = undefined` per tutta la feature `organization` — feature-libs/organization/administration/core/store/effects/budget.effect.ts:L122-L139, feature-libs/organization/administration/core/store/reducers/index.ts:L209-L227.
8. **Perché** `ItemService.save` **legge** `form.value` **prima di** `form.disable()`**?** Perché in alcuni casi `disable()` trasforma il valore in un oggetto vuoto (commento nel sorgente) — feature-libs/organization/administration/components/shared/item.service.ts:L56-L61.
9. **Come si aggiunge una vista basata sul framework in un'altra libreria senza toccare l'enum originale?** Con module augmentation di `OrganizationTableType` più un'assegnazione runtime, poi una sottoclasse di `ListService` con il nuovo `tableType`, come fa l'account summary — feature-libs/organization/account-summary/components/model/augmented.model.ts:L10-L16, feature-libs/organization/account-summary/components/services/account-summary-unit-list.service.ts:L15-L17.
10. **Perché** `UnitOrderFacade` **usa** `async: true`**?** Il resolver aggiunge un `delay(0)` dopo aver risolto il feature, per dare tempo alla feature NgRx lazy di essere pronta — core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L36, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L53-L55, feature-libs/organization/unit-order/root/facade/unit-order.facade.ts:L13-L33.
11. **Perché un componente proiettato in** `cx-org-card` **vede il** `MessageService` **della card?** `CardComponent` lo dichiara nei `providers`, che in Angular sono visibili anche al contenuto proiettato (a differenza dei `viewProviders`) — feature-libs/organization/administration/components/shared/card/card.component.ts:L34-L34, angular:adev/src/content/guide/di/debugging-and-troubleshooting-di.md:L285-L287.

TAVOLA 28.1 **4.1 Prima apertura di /organization/budgets: lazy loading, guardie, child route**

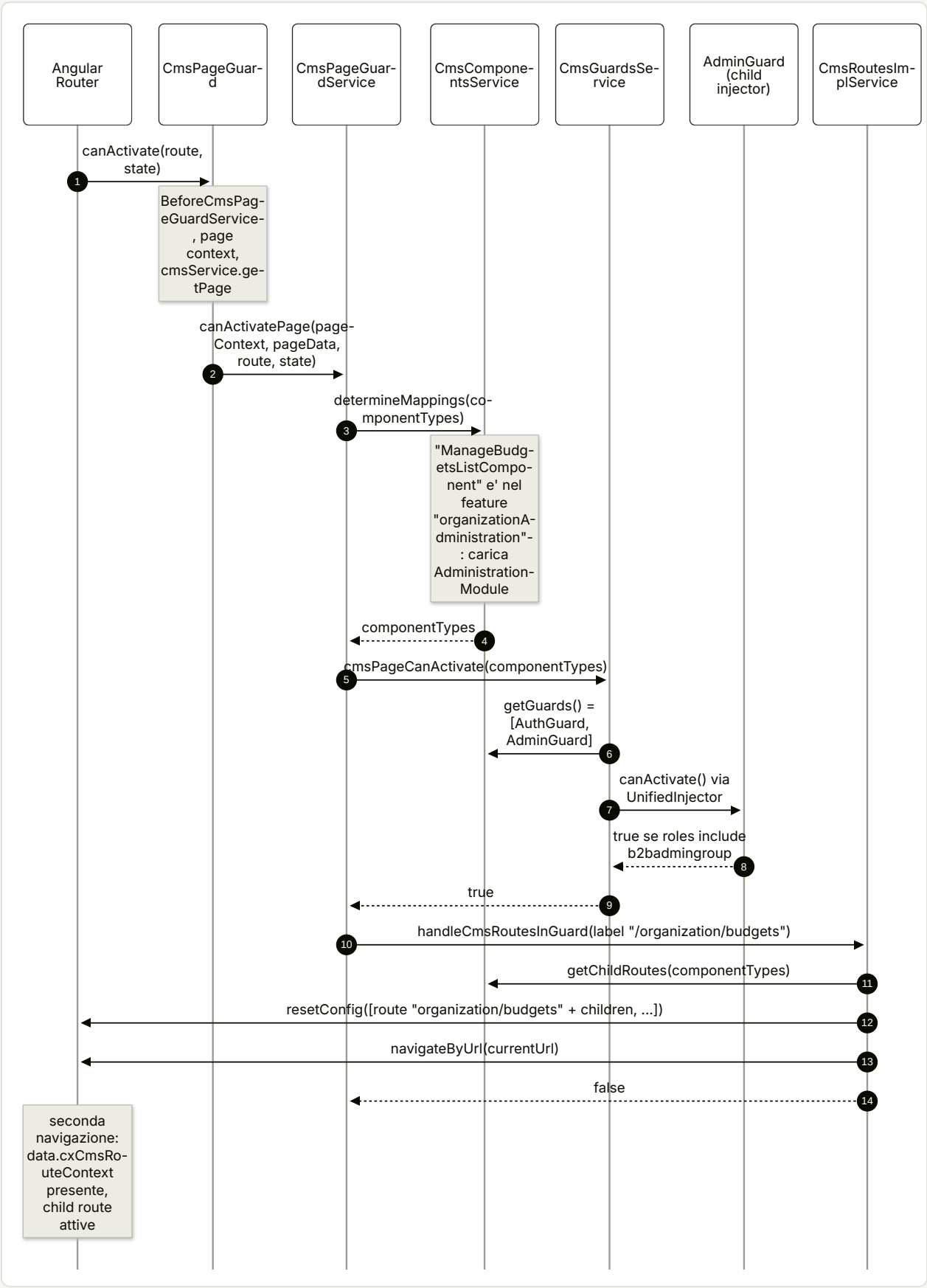
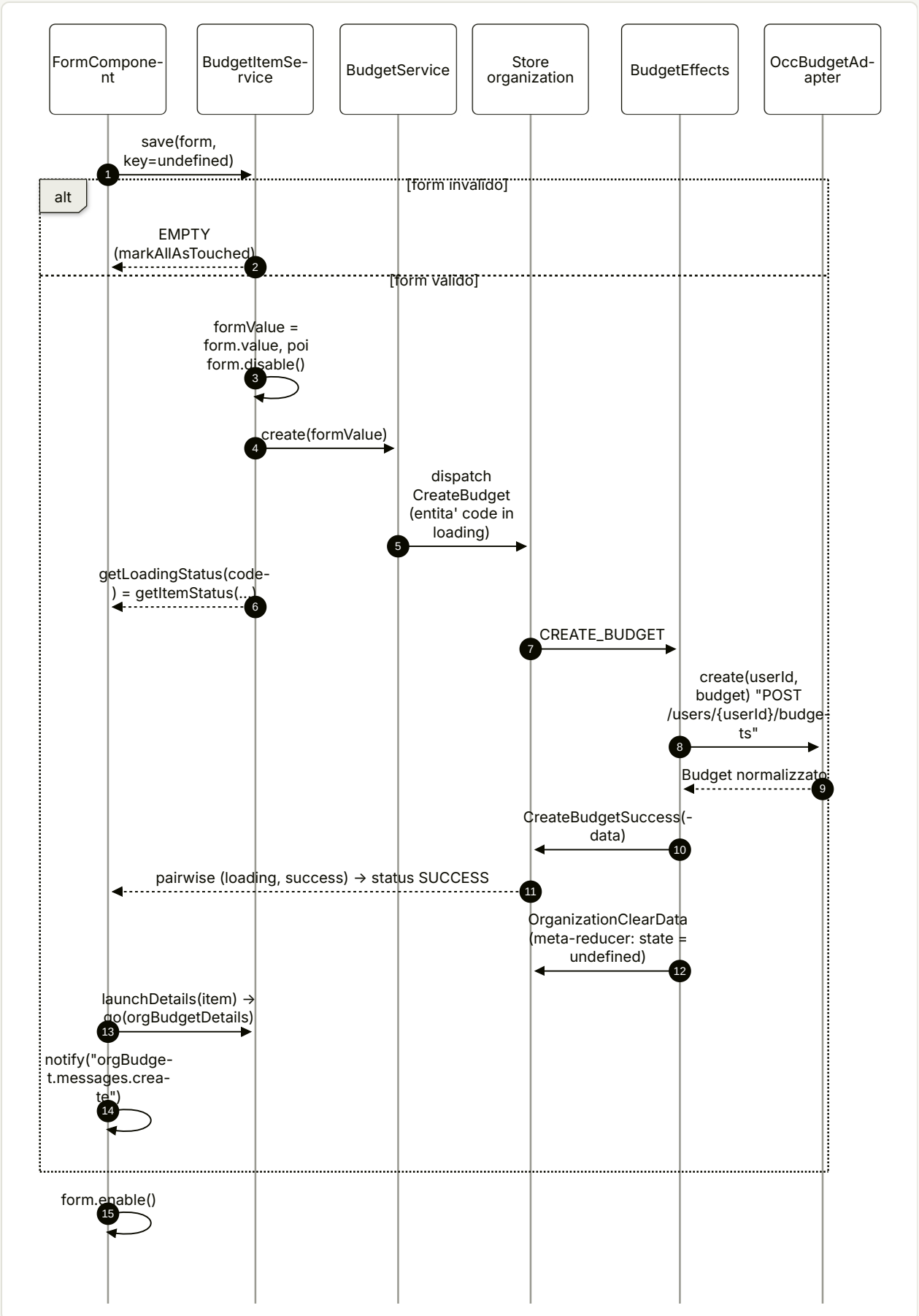


TAVOLA 28.2 **4.3 Salvataggio del form budget (create)**



29

ASM (Assisted Service Module)

Il customer support agent che emula un cliente: ricerca, sessione, carrello, Customer 360.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

L'Assisted Service Module (`@spartacus/asm`) permette a un **customer support agent** (CS agent: commesso in negozio o operatore di call center) di autenticarsi nello storefront con le proprie credenziali e di **emulare** un cliente: cercarlo, aprire una sessione "come se fosse lui", vederne e modificarne il carrello, legare un carrello anonimo al suo account, crearne l'account, consultare una vista aggregata (Customer 360). Tutto questo avviene nello stesso storefront usato dai clienti, sopra una barra ASM che si apre solo quando serve.

Dal punto di vista architetturale ASM è interessante per tre ragioni:

1. **È una feature lazy che però deve modificare il cuore dell'autenticazione.** La parte eager (`AsmRootModule`) sostituisce con `useExisting` quattro servizi del core — `AuthStorageService`, `AuthService`, `AuthHttpHeaderService`, `ProtectedRoutesService` — e aggiunge un interceptor HTTP (`feature-libs/asm/root/asm-root.module.ts:L65-L92`). Il resto (store NgRx, componenti, adapter OCC) è caricato on demand con il nome feature `'asm'` (`feature-libs/asm/root/feature-name.ts:L7-L7`, `projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts:L18-L27`).
2. **Due "identità" condividono un unico slot token.** Il core ha un solo token OAuth in `AuthStorageService`; ASM aggiunge un `TokenTarget` (`CSAgent` / `User`) che dice di chi è quel token, più un secondo token "emulato" del cliente da ripristinare (`feature-libs/asm/root/services/asm-auth-storage.service.ts:L18-L21`, `feature-libs/asm/root/services/asm-auth-storage.service.ts:L37-L50`). L'emulazione OCC, invece, è ottenuta impostando lo `userId` al `customerId` del cliente invece di `current` (`feature-libs/asm/root/services/csagent-auth.service.ts:L134-L141`).
3. **Non deve pesare sull'utente normale.** Lo dice esplicitamente il sorgente: "This service is added to avoid any polution of the UI and runtime performance for the ordinary production user" (`feature-libs/asm/root/services/asm-enabler.service.ts:L13-L17`). La UI viene caricata solo se l'URL contiene `asm=true`, se ASM è stato usato prima nel browser o se si arriva da un deep link di emulazione (`feature-libs/asm/root/services/asm-enabler.service.ts:L42-L49`).

Questo capitolo presuppone il capitolo 22 — Autenticazione (token, `AuthInterceptor`, `AuthHttpHeaderService`, `UserIdService`) e il capitolo 12 — Feature module e lazy loading (`featureModules`, `facadeFactory`). I dialog ASM usano il launch dialog descritto in 18 — Layout, breakpoint, launch dialog; lo store usa i loader di 19 — State management.

2. Mappa dei file

La libreria ha 12 entry point ng-packagr, uno per ogni `ng-package.json`: `feature-libs/asm/{,root,core,occ,components,assets}/ng-package.json` e `feature-libs/asm/customer-360/{,root,core,occ,components,assets}/ng-package.json`. L'entry point principale, per esempio, pubblica in `../../dist/asm` con `entryFile: ./public_api.ts` — `feature-libs/asm/ng-package.json:L1-L8`.

PATH	RUOLO
`feature-libs/asm/asm.module.ts`	`AsmModule = AsmComponentsModule + AsmCoreModule + AsmOccModule` (target del lazy load)

PATH	RUOLO
feature-libs/asm/root/asm-root.module.ts	eager: override servizi auth, route deep link `asmDeepLink`, interceptor, `defaultAsmConfig`
feature-libs/asm/root/asm-loader.module.ts	`APP_INITIALIZER` non bloccante → `AsmEnablerService.load()`
feature-libs/asm/root/asm-constants.ts	`ASM_ENABLED_LOCAL_STORAGE_KEY = 'asm_enabled'`
feature-libs/asm/root/feature-name.ts	`ASM_FEATURE = 'asm'`
feature-libs/asm/root/config/asm-config.ts	`AsmConfig` (abstract, `useExisting: Config`, augment di `Config`)
feature-libs/asm/root/config/default-asm-config.ts	`defaultAsmConfig` (timer, ricerca, customer list, header user id)
feature-libs/asm/root/services/asm-enabler.service.ts	decide se caricare la UI ASM
feature-libs/asm/root/services/asm-auth-storage.service.ts	`AsmAuthStorageService`, enum `TokenTarget`, token emulato
feature-libs/asm/root/services/csagent-auth.service.ts	`CsAgentAuthService`: login/logout agente, avvio emulazione
feature-libs/asm/root/services/asm-auth.service.ts	`AsmAuthService`: override di `AuthService`
feature-libs/asm/root/services/asm-auth-http-header.service.ts	`AsmAuthHttpHeaderService`: token agente sulle richieste marcate
feature-libs/asm/root/services/asm-protected-routes.service.ts	nessuna route protetta per l'agente loggato
feature-libs/asm/root/services/asm-deep-link.service.ts	parametri e navigazione del deep link
feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts	header `sap-commerce-cloud-user-id`
feature-libs/asm/root/facade/*.facade.ts	`AsmBindCartFacade`, `AsmCreateCustomerFacade`, `AsmCustomerListFacade` (proxy lazy)
feature-libs/asm/root/model/augmented-core.model.ts	augment di `LAUNCH_CALLER` con `ASM_*`
feature-libs/asm/core/asm-core.module.ts	store, connector, facade, `MODULE_INITIALIZER` di persistenza
feature-libs/asm/core/store/	stato `asm`: due loader di ricerca + `asmUi`; effects e meta-reducer
feature-libs/asm/core/facade/asm.service.ts	`AsmService` (dispatch/select sullo store)
feature-libs/asm/core/services/asm-state-persistence.service.ts	sync `{ui, emulatedUserToken, tokenTarget}` ↔ storage `asm`
feature-libs/asm/core/connectors/	`AsmAdapter`, `AsmConnector`, token normalizer
feature-libs/asm/occ/adapters/occ-asm.adapter.ts	`OccAsmAdapter` verso `/assistedservicewebservices/...`
feature-libs/asm/occ/adapters/default-occ-asm-config.ts	endpoint di default
feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts	barra ASM e orchestrazione della sessione
feature-libs/asm/components/ (altri)	login form, selezione cliente, emulazione, bind cart, dialog, timer, customer list, creazione cliente

PATH	RUOLO
feature-libs/asm/components/services/asm-component.service.ts	servizio UI condiviso (logout, deep link, azioni dialog)
feature-libs/asm/customer-360/root/asm-customer-360-root.module.ts	feature `asmCustomer360` dipendente da `asm`, `SiteContextInterceptor`
feature-libs/asm/customer-360/core/services/asm-customer-360.service.ts	`AsmCustomer360Service` (Command che aggrega le query)
feature-libs/asm/customer-360/occ/adapters/occ-asm-customer-360.adapter.ts	POST `.../customer360`
feature-libs/asm/customer-360/components/	dialog C360, config delle tab, host di sezione, 11 sezioni
projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts	wiring nell'app demo

File aggiuntivi

PATH	RUOLO
feature-libs/asm/public_api.ts	Entry `@spartacus/asm`: riesporta solo `asm.module` (`:L10-L10`).
feature-libs/asm/schematics/add-asm/index.ts	`addAsmFeatures()`: factory degli schematics `ng-add/add` di `@spartacus/asm` (vedi 3.16).
feature-libs/asm/schematics/add-asm/index_spec.ts	Test dello schematic: lazy loading, nessuna installazione delle dipendenze, scss, `angular.json`, eager loading.
feature-libs/asm/_index.scss	Entry SCSS: core Spartacus + `styles/index` + `customer-360/index`.
feature-libs/asm/styles/_index.scss	Importa core e `components/index`.
feature-libs/asm/styles/components/_index.scss	Importa i 13 file componente e definisce `$asm-components-allowlist` con `@extend` dei placeholder.
feature-libs/asm/styles/components/_asm-bind-cart-dialog.component.scss	Placeholder `%cx-asm-bind-cart-dialog`.
feature-libs/asm/styles/components/_asm-bind-cart.component.scss	Placeholder `%cx-asm-bind-cart`.
feature-libs/asm/styles/components/_asm-create-customer-form.component.scss	Placeholder `%cx-asm-create-customer-form`.
feature-libs/asm/styles/components/_asm-main-ui.component.scss	Placeholder `%cx-asm-main-ui` (barra ASM).
feature-libs/asm/styles/components/_asm-save-cart-dialog.component.scss	Placeholder `%cx-asm-save-cart-dialog`.
feature-libs/asm/styles/components/_asm-session-timer.component.scss	Placeholder `%cx-asm-session-timer`.
feature-libs/asm/styles/components/_asm-switch-customer-dialog.component.scss	Placeholder `%cx-asm-switch-customer-dialog`.
feature-libs/asm/styles/components/_asm-toggle-ui.component.scss	Placeholder `%cx-asm-toggle-ui`.
feature-libs/asm/styles/components/_csagent-login-form.component.scss	Placeholder `%cx-csagent-login-form`.

PATH	RUOLO
feature-libs/asm/styles/components/_customer-emulation.component.scss	Placeholder %cx-customer-emulation.
feature-libs/asm/styles/components/_customer-list.component.scss	Placeholder %cx-customer-list.
feature-libs/asm/styles/components/_customer-selection.component.scss	Placeholder %cx-customer-selection.
feature-libs/asm/styles/components/_dot-spinner.component.scss	Placeholder %cx-dot-spinner.

3. Come funziona passo per passo

3.1 Packaging: cosa è eager e cosa è lazy

L'app importa `AsmRootModule` eagerly e dichiara `featureModules[ASM_FEATURE].module: () => import('@spartacus/asm').then((m) => m.AsmModule)`; non sono dichiarati `cmsComponents`, perché la UI ASM non è un componente CMS (projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts:L18-L27). `AsmModule` si limita a importare componenti, core e occ (feature-libs/asm/asm.module.ts:L12-L15).

`AsmRootModule` (feature-libs/asm/root/asm-root.module.ts:L34-L95) contiene tutto ciò che deve esistere **prima** che la feature sia caricata:

- importa `AsmLoaderModule` e `PageLayoutModule`;
- registra una route figlia con `path: null` (con `// @ts-ignore`), `CmsPageGuard`, `PageLayoutComponent` e `data.cxRoute: 'asmDeepLink'` nel contesto della home page CMS (feature-libs/asm/root/asm-root.module.ts:L38-L51); il path reale arriva dalla `RoutingConfig` di default `asmDeepLink.paths: ['assisted-service/emulate']` (feature-libs/asm/root/asm-root.module.ts:L56-L64) — vedi il meccanismo delle route semantiche nel capitolo 14 — Routing;
- fornisce `defaultAsmConfig` (feature-libs/asm/root/asm-root.module.ts:L55-L55);
- sostituisce i servizi auth del core e aggiunge `UserIdHttpHeaderInterceptor` (feature-libs/asm/root/asm-root.module.ts:L65-L92).

Il token `DELEGATED_AUTH_HTTP_HEADER_SERVICE` è fornito anch'esso con `AsmAuthHttpHeaderService`, come provider di retro-compatibilità per il fix CXSPA-12514 da rimuovere quando il toggle `enableExpiredRefreshTokenHandlers` sarà completamente attivo (feature-libs/asm/root/asm-root.module.ts:L77-L83). Il token vive nel core senza factory ed esiste perché altre librerie (es. Punchout) possano delegare all'override ASM senza importarlo a compile-time (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L72-L82).

3.2 Attivazione: `AsmLoaderModule` e `AsmEnablerService`

`AsmLoaderModule` registra un `APP_INITIALIZER` la cui factory ritorna una funzione `void`: chiama `AsmEnablerService.load()` senza restituire una Promise, quindi **non blocca il bootstrap** (feature-libs/asm/root/asm-loader.module.ts:L16-L40). Il commento motiva la scelta: router e state "aren't available in an optimized way during the APP_INITIALIZER" (feature-libs/asm/root/asm-loader.module.ts:L29-L34). Per l'ordine degli initializer vedi 10 — Bootstrap.

`AsmEnablerService.isEnabled()` (feature-libs/asm/root/services/asm-enabler.service.ts:L42-L49) restituisce vero se:

- `isLaunched()`: la query string di `Location.path()` contiene esattamente il parametro `asm=true` (feature-libs/asm/root/services/asm-enabler.service.ts:L55-L58);
- `isUsedBefore()`: `localStorage['asm_enabled'] === 'true'` (feature-libs/asm/root/services/asm-enabler.service.ts:L70-L79, costante in feature-libs/asm/root/asm-constants.ts:L7-L7);
- `isEmulateInURL()`: `location.path().indexOf('assisted-service/emulate?') > 0` (feature-libs/asm/root/services/asm-enabler.service.ts:L63-L65).

Al primo lancio via `asm=true` scrive `asm_enabled='true'`, così le visite successive riaprono la barra anche senza parametro (feature-libs/asm/root/services/asm-enabler.service.ts:L43-L47).

Se abilitato, `addUi()` risolve la feature `'asm'` (lazy load di `AsmModule`) e poi lancia `LAUNCH_CALLER.ASM` con il `LaunchDialogService` (feature-libs/asm/root/services/asm-enabler.service.ts:L84-L88). `LAUNCH_CALLER.ASM` è già nell'enum dello storefront (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L93-L94); la config di layout di default, fornita dai componenti, lo aggancia all'outlet `'cx-storefront'` con `AsmMainUiComponent` (feature-libs/asm/components/default-asm-layout.config.ts:L10-L17). `LaunchDialogService.launch()` trova la config del caller, sceglie la strategia di render e la invoca (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L78-L97); per una config con `outlet` è `OutletRenderStrategy`, che aggiunge il componente all'outlet in posizione `BEFORE` (default) e forza il render dell'outlet (core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L45-L84). Per gli outlet vedi 17 — Outlet.

Il lazy load di `AsmModule` porta con sé `AsmCoreModule`, che registra lo store (`AsmStoreModule`) e un `MODULE_INITIALIZER` che invoca `AsmStatePersistenceService.initSync()` (feature-libs/asm/core/asm-core.module.ts:L15-L35).

3.3 La duplicazione nel core: `AuthService.refreshAuthConfig()`

C'è un secondo punto che "sa" se ASM è attivo, e sta nel core. `AuthService` contiene una copia della logica di rilevamento (`isAsmEnabled`, `isLaunched`, `isEmulateInURL`, `isUsedBefore` con letterale `'asm_enabled'`) e la usa in `refreshAuthConfig()`: se il toggle `authorizationCodeFlowByDefault` è attivo e ASM è abilitato, il client OAuth diventa `'asm_client'` e `isUsingASMClient` passa a `true` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L263-L327). `refreshAuthConfig()` e `checkOAuthParamsInUrl()` sono invocati da un initializer del core, solo in browser e dopo che la config è stabile (core-libs/core/src/auth/user-auth/user-auth.module.ts:L43-L54). Con i default attuali `authorizationCodeFlowByDefault` vale `true` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029), quindi **il percorso normale del login agente è l'Authorization Code flow** (§3.6), non il form username/password.

3.4 `AsmConfig`

`AsmConfig` è una classe astratta `@Injectable({ providedIn: 'root', useExisting: Config })` con chiave `asm` e sotto-chiavi `agentSessionTimer`, `customerSearch`, `customerList`, `asmSessionSupport`, `userIdHttpHeader`, `createCustomer`; augmenta l'interfaccia `Config` di `@spartacus/core` (feature-libs/asm/root/config/asm-config.ts:L11-L59). I default sono in feature-libs/asm/root/config/default-asm-config.ts:L11-L84 (dettaglio nella §6). Siccome `defaultAsmConfig` è fornito dal modulo eager, è disponibile prima del lazy load (feature-libs/asm/root/asm-root.module.ts:L55-L55). Il meccanismo generale è nel capitolo 11 — Configurazione.

3.5 Il modello di stato dell'autenticazione ASM

`AsmAuthStorageService extends AuthStorageService` (feature-libs/asm/root/services/asm-auth-storage.service.ts:L37-L139) aggiunge due `BehaviorSubject`:

- `_tokenTarget$`, inizializzato a `TokenTarget.User`: dice se il token corrente è dell'agente o di un utente (feature-libs/asm/root/services/asm-auth-storage.service.ts:L38-L39);
- `emulatedUserToken$`: il token del cliente che era loggato quando l'agente ha fatto login "sopra" di lui (caso in-store), da ripristinare al logout dell'agente (feature-libs/asm/root/services/asm-auth-storage.service.ts:L41-L50).

Il costruttore legge **sincronicamente** lo stato persistito con chiave `'asm'` tramite `StatePersistenceService.readStateFromStorage` e ripristina token emulato e target (feature-libs/asm/root/services/asm-auth-storage.service.ts:L52-L69). Il motivo si deduce dal codice: la sincronizzazione completa (`initSync`) parte solo dopo il lazy load (§3.11), mentre il target del token serve già alle prime richieste intercettate (§3.8).

Combinando questi due valori con lo `userId` del core si ottengono i predicati chiave:

PREDICATO	DEFINIZIONE	RIFERIMENTO
agente loggato	token presente && target `CSAgent` && `access_token` ≠ quello del token emulato	feature-libs/asm/root/services/csagent-auth.service.ts:L148-L164
cliente emulato	`userId` ≠ `anonymous` e ≠ `current`	feature-libs/asm/root/services/csagent-auth.service.ts:L171-L173, core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L81-L88
utente "loggato" (per il resto dello storefront)	token presente && (target `User` || (target `CSAgent` && emulato))	feature-libs/asm/root/services/asm-auth.service.ts:L201-L214

L'ultima riga è il trucco centrale: durante un'emulazione il token è dell'agente, ma per guard, carrello e checkout l'utente risulta loggato e le chiamate OCC usano `/users/<customerId>/...`.

3.6 Login dell'agente

Authorization Code flow (default). Con `authorizationCodeFlowByDefault` il template mostra il link "Sign In as Agent" invece del form (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.html:L45-L63, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.html:L116-L124). Il click chiama `CsAgentAuthService.authorizeCustomerSupportAgentWhenUseCodeFlow()` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L386-L388), che:

1. legge sincronicamente `customerId` (da `UserAccountFacade.get()`) e il token utente corrente (feature-libs/asm/root/services/csagent-auth.service.ts:L90-L101);
2. **prima del redirect a pagina intera** persiste lo stato di emulazione: dispatch `Logout`; se c'era un cliente loggato `setUserId(customerId)` + `setEmulatedUserToken(userToken)` + `Login`, altrimenti `userId = anonymous` e token emulato pulito (feature-libs/asm/root/services/csagent-auth.service.ts:L103-L120). Il commento spiega che `initLoginFlow()` naviga via e ciò che segue "does not run reliably" (feature-libs/asm/root/services/csagent-auth.service.ts:L104-L106);
3. passa il target a `CSAgent` e chiama `initLoginFlow()`; in errore torna a `User` (feature-libs/asm/root/services/csagent-auth.service.ts:L122-L126).

Al ritorno dall'Authorization Server l'initializer del core chiama `refreshAuthConfig()` (che imposta di nuovo `asm_client`) e poi `checkOAuthParamsInUrl()`, che per DI è l'override di `AsmAuthService` (feature-libs/asm/root/services/asm-auth.service.ts:L69-L95): se non si usa il client ASM delega al core; altrimenti

`tryLogin()` e, se il token è stato ricevuto, target `CSAgent`, `userId = anonymous` **solo se non c'è già un'emulazione pendente** (CXSPA-13777), dispatch `Login` e `authRedirectService.redirect()`. Il core, simmetricamente, salta l'impostazione di `userId = current` quando `isEmulated` o `isUsingASMClient` è vero (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L97-L104).

Password flow (toggle disattivato). `CSAgentLoginFormComponent` emette `{userId, password}` (feature-libs/asm/components/csagent-login-form/csagent-login-form.component.ts:L55-L64); `AsmMainUiComponent.loginCustomerSupportAgent()` chiama `authorizeCustomerSupportAgent()` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L376-L384), che passa il target a `CSAgent` **prima** di `authorizeWithPasswordFlow()`, poi dispatcha `Logout` e, come nel code flow, avvia subito l'emulazione del cliente già loggato oppure imposta `anonymous` (feature-libs/asm/root/services/csagent-auth.service.ts:L46-L84). Deduzione dal codice: il token restituito dal password flow finisce nello stesso `AuthStorageService` del token utente, quindi il target deve già dire "CSAgent" quando arriva.

3.7 Che cosa vede il resto dello storefront mentre l'agente è loggato

`AsmAuthService` ritocca il comportamento di `AuthService`:

- `loginWithCredentials()` rifiuta il login cliente (messaggio `asm.auth.agentLoggedInError`) se c'è un token con target `CSAgent` (feature-libs/asm/root/services/asm-auth.service.ts:L97-L134);
- `loginWithRedirect()`: se l'agente è attivo ma non emula nessuno, invece dell'errore reindirizza a `'/'`, per gestire guard come `CheckoutAuthGuard` che mandano a `/login` (feature-libs/asm/root/services/asm-auth.service.ts:L136-L156);
- `coreLogout()` in emulazione **non revoca** i token: pulisce il token emulato, `clearUserId()` e dispatcha `Logout`; altrimenti delega al core (feature-libs/asm/root/services/asm-auth.service.ts:L180-L196).

`AsmProtectedRoutesService.isUrlProtected()` restituisce `false` per qualsiasi URL quando l'agente è loggato, così può navigare le pagine CMS anche senza emulare (feature-libs/asm/root/services/asm-protected-routes.service.ts:L15-L43).

3.8 Le richieste HTTP: quale token, quali header

Le chiamate ai web service ASM sono marcate dall'adapter con l'header interceptor `cx-use-csagent-token` (`USE_CUSTOMER_SUPPORT_AGENT_TOKEN`, definito nel core: core-libs/core/src/occ/utils/interceptor-util.ts:L10-L10) e costruite con `{ baseSite: false, prefix: false }`, cioè fuori dal prefisso `/occ/v2/{site}` (feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L99-L121).

L'`AuthInterceptor` del core chiede a `AuthHttpHeaderService` se intercettare gli errori e se aggiungere l'header, prende il token stabile e chiama `alterRequest()` (core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L31-L48). Per DI quel servizio è `AsmAuthHttpHeaderService`, che:

- estende `shouldAddAuthorizationHeader()` e `shouldCatchError()` alle richieste con `cx-use-csagent-token` (feature-libs/asm/root/services/asm-auth-http-header.service.ts:L60-L76);
- in `alterRequest()`, per le richieste CS agent senza `Authorization`, aggiunge l'header con il token e rimuove `cx-use-csagent-token` "to avoid problems with CORS" (feature-libs/asm/root/services/asm-auth-http-header.service.ts:L84-L106);
- per le richieste OCC **non** marcate, quando il target è `CSAgent` e c'è un token emulato con `access_token`, usa il token del cliente (`getTokenForRequest` / `shouldUseEmulatedUserToken`, feature-libs/asm/root/services/asm-auth-http-header.service.ts:L108-L133). È il caso in-store: l'agente ha fatto login sopra la sessione di un cliente, e le chiamate "normali" continuano col token di quest'ultimo;

- su refresh token scaduto, se l'agente è loggato imposta il logout in corso, esegue `logoutCustomerSupportAgent()` e mostra `asm.csagentTokenExpired`; altrimenti comportamento del core (feature-libs/asm/root/services/asm-auth-http-header.service.ts:L149-L167).

Altri due interceptor completano il quadro:

- `UserIdHttpHeaderInterceptor`: no-op se `asm.userIdHttpHeader.enable` è falsy; altrimenti legge `OCC_HTTP_TOKEN.sendUserIdAsHeader` dal context della richiesta: se stringa usa quel valore, se `true` usa `takeUserId()` scartando gli id costanti (`anonymous`, `current`, ...), e imposta l'header `sap-commerce-cloud-user-id` (feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L44-L84). Il context token è nel core (core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20) e lo valorizzano, fra gli adapter core, product search e user cost centers (core-libs/core/src/occ/adapters/product/occ-product-search.adapter.ts:L52-L52, core-libs/core/src/occ/adapters/user/occ-user-cost-centers.adapter.ts:L33-L33). Il JSDoc della config avverte che l'header può rompere istanze Commerce che non lo consentono in CORS (feature-libs/asm/root/config/asm-config.ts:L39-L50);
- `SiteContextInterceptor` (registrato da `AsmCustomer360RootModule`, feature-libs/asm/customer-360/root/asm-customer-360-root.module.ts:L29-L33) aggiunge `lang` e `curr` come query param alle richieste verso il base URL OCC che contengono `/assistedservicewebservices/` (feature-libs/asm/customer-360/root/interceptors/site-context.interceptor.ts:L54-L72).

> DA VERIFICARE — L'ordine effettivo fra `UserIdHttpHeaderInterceptor`, `SiteContextInterceptor` e `AuthInterceptor` dipende dall'ordine con cui l'app importa i moduli che li registrano (multi provider `HTTP_INTERCEPTORS`); il codice ASM non lo fissa. Per il meccanismo generale vedi 06 — HttpClient e interceptor.

3.9 Ricerca cliente: store, effect, adapter

`CustomerSelectionComponent` cerca per nome (`searchTerm`) o per numero d'ordine (`searchOrder`) con `debounceTime(300)`, solo da 3 caratteri in su, con `pageSize = asm.customerSearch.maxResults`; i due campi si escludono (feature-libs/asm/components/customer-selection/customer-selection.component.ts:L96-L192). `AsmService` — servizio `providedIn: 'root'` che lavora sullo store, non una facade proxy — dispatcha `CustomerSearch` (feature-libs/asm/core/facade/asm.service.ts:L20-L79).

Lo stato `asm` (feature-libs/asm/core/store/asm-state.ts:L15-L23) contiene due `LoaderState<CustomerSearchPage>` (`customerSearchResult`, `customerListCustomersSearchResult`) e `asmUi`; `AsmStoreModule` registra la feature store con reducer, meta-reducer e gli effects `CustomerEffects` e `AsmSessionEffects` (feature-libs/asm/core/store/asm-store.module.ts:L16-L25). I reducer di ricerca sono `StateUtils.loaderReducer` (feature-libs/asm/core/store/reducers/index.ts:L24-L34).

`CustomerEffects.customerSearch$` usa `switchMap` (una nuova ricerca cancella la precedente) sul connector; un 400 con messaggio `'Wrong orderId'` è convertito in un successo vuoto, gli altri errori in `CustomerSearchFail` con l'errore normalizzato (feature-libs/asm/core/store/effects/customer.effect.ts:L18-L43). `OccAsmAdapter.customerSearch()` è una POST con body `{query, orderId, customerListId}` e query param `baseSite`, `sort` (default `byNameAsc` solo se non c'è `customerListId`), `pageSize`, `page` (feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L131-L186).

3.10 Avvio e fine dell'emulazione

Il submit della selezione emette `{ customerId, parameters: { orderId } }` (feature-libs/asm/components/customer-selection/customer-selection.component.ts:L221-L232). `AsmMainUiComponent.startCustomerEmulationSession()` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L394-L418):

- chiama `CsAgentAuthService.startCustomerEmulationSession(customerId)`, che pulisce il token emulato, dispatcha `Logout`, imposta `userId = customerId` e dispatcha `Login` (feature-libs/asm/root/services/csagent-auth.service.ts:L134-L141);
- se `asm.asmSessionSupport.enabled`, crea un evento `{ eventType: 'StartSession' }` (`AsmSessionCreationAction` → `AsmSessionEffects` → POST `asmSessionEvents`: feature-libs/asm/core/store/effects/asm-session.effect.ts:L14-L41, feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L70-L97);
- se riceve `parameters` delega la navigazione ad `AsmDeepLinkService.handleNavigation()` (§3.12). I `parameters` arrivano dal deep link, ma anche dalla selezione cliente, che passa sempre `{ orderId }`: per esempio, dopo una ricerca per numero d'ordine si naviga a `orderDetails` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.html:L109-L112, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L406-L411, feature-libs/asm/root/services/asm-deep-link.service.ts:L51-L56). Senza `customerId` mostra `asm.error.noCustomerId`.

Quando `isUserLoggedIn()` diventa vero (target `CSAgent` && emulato) lo stream `customer$` chiama `handleCustomerSessionStartRedirection()`, che naviga a `'/'` se la sessione è stata appena avviata (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L166-L175, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L363-L374).

Fine emulazione (l'agente resta). "End session" in `CustomerEmulationComponent` chiama `authService.logout()` (feature-libs/asm/components/customer-emulation/customer-emulation.component.ts:L69-L71); il percorso standard di logout arriva a `AsmAuthService.coreLogout()` che, essendo in emulazione, pulisce solo token emulato e `userId` senza revoca (feature-libs/asm/root/services/asm-auth.service.ts:L180-L196).

Logout dell'agente. `logoutCustomerSupportAgent()` (feature-libs/asm/root/services/csagent-auth.service.ts:L186-L209) revoca i token, dispatcha l'azione stringa `'[Auth] Logout Customer Support Agent'`, rimette il target a `User`; se era in corso un'emulazione con token emulato (caso in-store) ripristina il token del cliente con `userId = current` e `Login`, altrimenti chiama `authService.logout()`. Il logout è attivato dal pulsante, dalla scadenza del timer di sessione (feature-libs/asm/components/asm-session-timer/asm-session-timer.component.ts:L23-L97) o dal refresh token scaduto (§3.8). Quando `customerSupportAgentLoggedIn$` diventa falso la barra chiude i dialog aperti (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L153-L162).

3.11 Persistenza dello stato ASM

`AsmStatePersistenceService.initSync()` sincronizza con chiave `'asm'` lo stato `{ ui, emulatedUserToken, tokenTarget }` tramite `StatePersistenceService.syncWithStorage`, filtrando le emissioni finché la slice `store.asm` non esiste (la feature potrebbe essere ancora in caricamento) (feature-libs/asm/core/services/asm-state-persistence.service.ts:L48-L87). Il `refresh_token` del token emulato **non** viene persistito "to minimize risk of user account hijacking" (feature-libs/asm/core/services/asm-state-persistence.service.ts:L74-L79). `onRead()` riporta i valori nello store (`AsmUiUpdate`) e nell'`AsmAuthStorageService` (feature-libs/asm/core/services/asm-state-persistence.service.ts:L93-L105). La lettura anticipata nel costruttore di `AsmAuthStorageService` (§3.5) copre la finestra fra bootstrap e lazy load.

3.12 Deep link `assisted-service/emulate?customerId=...`

1. L'URL attiva la UI (`isEmulateInURL`, §3.2).
2. `AsmDeepLinkService` legge i parametri da `winRef.location.search` **una sola volta**, nel costruttore (feature-libs/asm/root/services/asm-deep-link.service.ts:L18-L24); `getParamsInUrl()` scarta le chiavi con valore falsy (feature-libs/asm/root/services/asm-deep-link.service.ts:L69-L79). I parametri sono `customerId`, `orderId`, `ticketId`, `cartId`, `cartType` (feature-libs/asm/root/model/asm.models.ts:L34-L44).

3. `AsmMainUiComponent.subscribeForDeeplink()` naviga a `'/'` ("to avoid 404") solo se l'URL corrente è un deep link di emulazione (`isEmulateInURL()`, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L283-L286) e, quando l'agente è loggato e c'è `customerId`: se un cliente è già loggato e non è stato emulato via deep link apre la conferma di switch, altrimenti chiama `startSessionWithParameters` in un `setTimeout` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L282-L316).
4. `confirmSwitchCustomer()` carica il cliente target con `userAccountFacade.getById()` e apre `ASM_SWITCH_CUSTOMER`; se è lo stesso cliente si limita a `setEmulatedByDeepLink(true)` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L318-L345). Con `SWITCH` il dialog fa logout del cliente corrente, lasciando che la sottoscrizione del punto 3 avvii la nuova emulazione (feature-libs/asm/components/asm-switch-customer-dialog/asm-switch-customer-dialog.component.ts:L61-L66).
5. Dopo l'avvio, `AsmDeepLinkService.handleNavigation()` naviga con priorità: `cartType==='active'` → `cart`; `cartType==='saved'` + `cartId` → `my-account/saved-cart/<id>`; `orderId` → `orderDetails`; `ticketId` → `supportTicketDetails` (feature-libs/asm/root/services/asm-deep-link.service.ts:L44-L64).
6. `AsmBindCartComponent` reagisce a deep link con `cartType` `inactive`/`active`: nasconde il pulsante bind e memorizza `deepLinkCartId`; solo per `inactive` mostra "save" e precompila il campo cart id; poi mostra l'alert e chiama `handleDeepLinkNavigation()` (feature-libs/asm/components/asm-bind-cart/asm-bind-cart.component.ts:L286-L314, in particolare feature-libs/asm/components/asm-bind-cart/asm-bind-cart.component.ts:L300-L310).

3.13 La barra ASM: orchestrazione in `AsmMainUiComponent`

Oltre a quanto visto, `ngOnInit` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L142-L254):

- se `asmCustomer360` è configurato ne avvia subito il lazy load (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L143-L150);
- osserva lo stato UI: con `status === 403` fa logout agente, `hideUi()` e azzera `status` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L183-L189);
- reagisce alla chiusura dei dialog: con `selectedUser` avvia l'emulazione (e va a `orders` per `ORDER_HISTORY`), con `customerId` (cliente appena creato) emula e torna a `'/'`, poi `ACTIVE_CART` → `cart`, `CUSTOMER_360` → apre il dialog C360; `'forbidden'` → logout + `hideUi()` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L190-L233);
- quando l'agente è loggato, dopo `debounceTime(500)`, lancia una ricerca fittizia `query: 'autoSearchToAvoidUnauthorizedLogin', pageSize: 1` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L235-L251). Lo scopo si ricostruisce dal codice: se l'utente autenticato non ha i permessi da agente, la ricerca fallisce con 403; il reducer `asmUi` fonde l'errore di `CustomerSearchFail` nello stato UI (feature-libs/asm/core/store/reducers/asm-ui.reducer.ts:L21-L26) e il `status: 403` fa scattare il logout descritto sopra.

`hideUi()` nasconde il componente tramite `@HostBinding('class.hidden')`, rimuove `asm_enabled` dal localStorage, chiama `oAuthLibWrapperService.refreshAuthConfig()` e `authService.updateIsUsingASMClient(false)` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L420-L425, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L113-L113). Il template mostra il login agente se non loggato, `cx-customer-emulation` se c'è un cliente, `cx-customer-selection` altrimenti; tutto nascosto se `collapsed` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.html:L83-L125).

Gli altri componenti in breve:

- `AsmBindCartComponent` (OnPush): valida che il cart id non sia quello attivo; se il carrello attivo del cliente ha articoli apre `ASM_BIND_CART`, altrimenti fa bind diretto e ricarica il carrello corrente (feature-libs/asm/components/asm-bind-cart/asm-bind-cart.component.ts:L77-L179); con `REPLACE` salva il vecchio carrello

come saved cart (feature-libs/asm/components/asm-bind-cart/asm-bind-cart.component.ts:L270-L284). Il bind passa da AsmBindCartService, un Command che usa uid dell'utente corrente (feature-libs/asm/core/facade/asm-bind-cart.service.ts:L23-L45), e da OccAsmAdapter.bindCart(), POST form-urlencoded (feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L188-L217).

- CustomerListComponent: liste clienti via AsmCustomerListService (QueryService per le liste, store per i clienti della lista: feature-libs/asm/core/facade/asm-customer-list.service.ts:L22-L84); chiude con 'forbidden' su 403 (feature-libs/asm/components/customer-list/customer-list.component.ts:L150-L214); startColumnAction() chiude con { actionType, selectedUser } (feature-libs/asm/components/customer-list/customer-list.component.ts:L305-L315).
- AsmCreateCustomerFormComponent: valida i campi, chiama la facade e chiude con l'utente creato; mappa ValidationError e AssistedServiceDuplicatedUidError su chiavi i18n (feature-libs/asm/components/asm-create-customer-form/asm-create-customer-form.component.ts:L82-L190).
- AsmSessionTimerComponent: countdown al secondo, a 0 logout di agente e cliente, reset su navigazione e cambio userId (feature-libs/asm/components/asm-session-timer/asm-session-timer.component.ts:L23-L97).
- I dialog ASM sono inlineRoot: true + DIALOG_TYPE.DIALOG (feature-libs/asm/components/default-bind-cart-layout.config.ts:L14-L22, feature-libs/asm/components/customer-list/default-customer-list-layout.config.ts:L10-L18); i relativi LAUNCH_CALLER sono aggiunti sia a livello di tipo sia a runtime (feature-libs/asm/root/model/augmented-core.model.ts:L9-L23).

3.14 Facade root e lazy loading dei servizi

Le tre facade root (AsmBindCartFacade, AsmCreateCustomerFacade, AsmCustomerListFacade) sono proxy facadeFactory sulla feature ASM_FEATURE (feature-libs/asm/root/facade/asm-bind-cart.facade.ts:L12-L26, feature-libs/asm/root/facade/asm-create-customer.facade.ts:L14-L25, feature-libs/asm/root/facade/asm-customer-list.facade.ts:L14-L39); facadeProviders nel core ASM collega ciascuna facade all'implementazione con useExisting (feature-libs/asm/core/facade/facade-providers.ts:L17-L33). Il proxy connette subito il risultato (connectable + ReplaySubject), quindi il metodo viene eseguito anche senza subscribe e più sottoscrizioni non lo ripetono (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L92). Vedi 20 — Facade, command & query.

3.15 Customer 360

Feature e dipendenze. AsmCustomer360RootModule dichiara featureModules.asmCustomer360.dependencies: [ASM_FEATURE] e l'alias asmCustomer360Core → asmCustomer360 (feature-libs/asm/customer-360/root/asm-customer-360-root.module.ts:L18-L36, nomi in feature-libs/asm/customer-360/root/feature-name.ts:L7-L8). L'app demo carica @spartacus/asm/customer-360 (projects/storefrontapp/src/app/spartacus/features/asm/asm-customer-360-feature.module.ts:L21-L33); AsmCustomer360Module importa core, occ, components e fornisce defaultAsmCustomer360Config (feature-libs/asm/customer-360/asm-customer-360.module.ts:L16-L24). AsmCustomer360CoreModule importa StoreFinderModule per la sezione mappa (feature-libs/asm/customer-360/core/asm-customer-360-core.module.ts:L12-L16).

Configurazione a tab. AsmCustomer360Config (chiave asmCustomer360, feature-libs/asm/customer-360/components/config/asm-customer-360-config.ts:L11-L21) contiene tabs, dateFormat, dateTimeFormat; ogni tab ha i18nNameKey e components, ogni componente { component?, requestData?, config? } (feature-libs/asm/customer-360/root/model/asm-customer-360-tabs-config.ts:L9-L13, feature-libs/asm/customer-360/root/model/asm-customer-360-tab-config.ts:L11-L21). requestData è un AsmCustomer360Query { type?, additionalRequestParameters? }, dove type può essere un AsmCustomer360Type o una stringa qualsiasi (feature-libs/asm/customer-360/root/model/asm-customer-360.model.ts:L243-L258); i 12 tipi standard sono in feature-libs/asm/customer-360/root/model/asm-customer-360.model.ts:L77-L90. Il default definisce 6 tab

(overview, profile, activity, feedback, promotions, store locations) (feature-libs/asm/customer-360/components/config/default-asm-customer-360-config.ts:L21-L142).

Caricamento dati. Il click su Customer 360 apre il dialog `ASM_CUSTOMER_360` passando `{ customer }` (feature-libs/asm/components/customer-emulation/customer-emulation.component.ts:L73-L84); il caller è aggiunto all'enum e mappato su `AsmCustomer360Component` come dialog inline root (feature-libs/asm/customer-360/root/model/augmented-core.model.ts:L9-L15, feature-libs/asm/customer-360/components/default-asm-customer-360-layout.config.ts:L14-L22). Il componente carica separatamente l'overview e i dati della tab corrente (feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L116-L164). `setTabData()` chiama `AsmCustomer360Facade.get360Data(currentTab.components)` e mappa ogni componente della tab sull'elemento della risposta con lo stesso `type` (feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L267-L294).

La facade è un proxy `facadeFactory` sulla feature `asmCustomer360` (feature-libs/asm/customer-360/root/facade/asm-customer-360.facade.ts:L13-L26). L'implementazione `AsmCustomer360Service` è un `Command` che raccoglie i `requestData` dei componenti e, se ce ne sono, invia **una sola** richiesta con `options.userId = customer.customerId`; altrimenti restituisce `{ value: [] }` (feature-libs/asm/customer-360/core/services/asm-customer-360.service.ts:L33-L73). L'adapter OCC fa POST su `/assistedservicewebservices/${baseSiteId}/users/${userId}/customer360` con body `{ customer360Queries }`, header CS agent e normalizer `ASM_CUSTOMER_360_NORMALIZER` (feature-libs/asm/customer-360/occ/adapters/default-occ-asm-customer-360-config.ts:L9-L18, feature-libs/asm/customer-360/occ/adapters/occ-asm-customer-360.adapter.ts:L54-L89).

Framework delle sezioni. Il template rende ogni componente della tab in `cx-asm-customer-360-section` passando `[component]`, `[config]`, `[data]` e ascoltando `(navigate)` (feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.html:L195-L205). `AsmCustomer360SectionComponent` fornisce **per istanza** `AsmCustomer360SectionContextSource` (e `AsmCustomer360SectionContext` con `useExisting`), inoltra gli input nei subject della sorgente e rende il componente con `*ngComponentOutlet`; al destroy completa i subject (feature-libs/asm/customer-360/components/sections/asm-customer-360-section/asm-customer-360-section.component.ts:L23-L75, feature-libs/asm/customer-360/components/sections/asm-customer-360-section/asm-customer-360-section.component.html:L1-L1). Il componente di sezione, creato dentro quell'host, inietta il context e legge `data$`, `config$`, scrive su `navigate$` (feature-libs/asm/customer-360/components/sections/asm-customer-360-section-context.model.ts:L14-L29). Esempi: la sezione active cart arricchisce le entry con `ProductService.get(code, ProductScope.DETAILS)` (feature-libs/asm/customer-360/components/sections/asm-customer-360-active-cart/asm-customer-360-active-cart.component.ts:L40-L75); la sezione activity mappa i tipi a rotte ed emette su `navigate$` (feature-libs/asm/customer-360/components/sections/asm-customer-360-activity/asm-customer-360-activity.component.ts:L91-L118); la sezione mappa combina `config$` e `data$` e chiama `storeFinderService.findStoresAction(...)` (feature-libs/asm/customer-360/components/sections/asm-customer-360-map/asm-customer-360-map.component.ts:L61-L91).

Navigazione dalla sezione. `navigate$` → `(navigate)` → `navigateTo()` chiude il dialog con un `AsmDialogActionEvent { actionType: NAVIGATE, route }` (feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L218-L225, feature-libs/asm/customer-360/core/utils.ts:L16-L27); `AsmMainUiComponent` lo passa ad `AsmComponentService.handleAsmDialogAction()`, che chiama `routingService.go(route)` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L268-L274, feature-libs/asm/components/services/asm-component.service.ts:L111-L118).

3.16 Installazione via schematics e stili

`addAsmFeatures()` valida l'installazione Spartacus, risolve le dipendenze tra feature con `analyzeCrossFeatureDependencies` ed esegue in catena `analyzeApplication`, `addFeatures`, `addPackageJsonDependenciesForLibrary(peerDependencies, ...)` e `finalizeInstallation` (feature-libs/asm/

schematics/add-asm/index.ts:L25-L43); è registrata sia come `ng-add` sia come `add` (feature-libs/asm/schematics/collection.json:L4-L16). Lo spec copre l'installazione senza feature, il lazy loading, l'assenza dei moduli dipendenti generati, gli stili e l'eager loading (feature-libs/asm/schematics/add-asm/index_spec.ts:L41-L134). Negli stili, `$asm-components-allowlist` contiene sei selettori ripetuti due volte (`cx-asm-toggle-ui`, `cx-csagent-login-form`, `cx-customer-emulation`, `cx-customer-selection`, `cx-dot-spinner`, `cx-customer-list`) (feature-libs/asm/styles/components/_index.scss:L17-L24): il loop `@each` emette quindi due volte lo stesso `@extend`, senza effetti visibili (Sass deduplica l'estensione). DA VERIFICARE: deduplicazione non verificata compilando, `node_modules` assente.

---

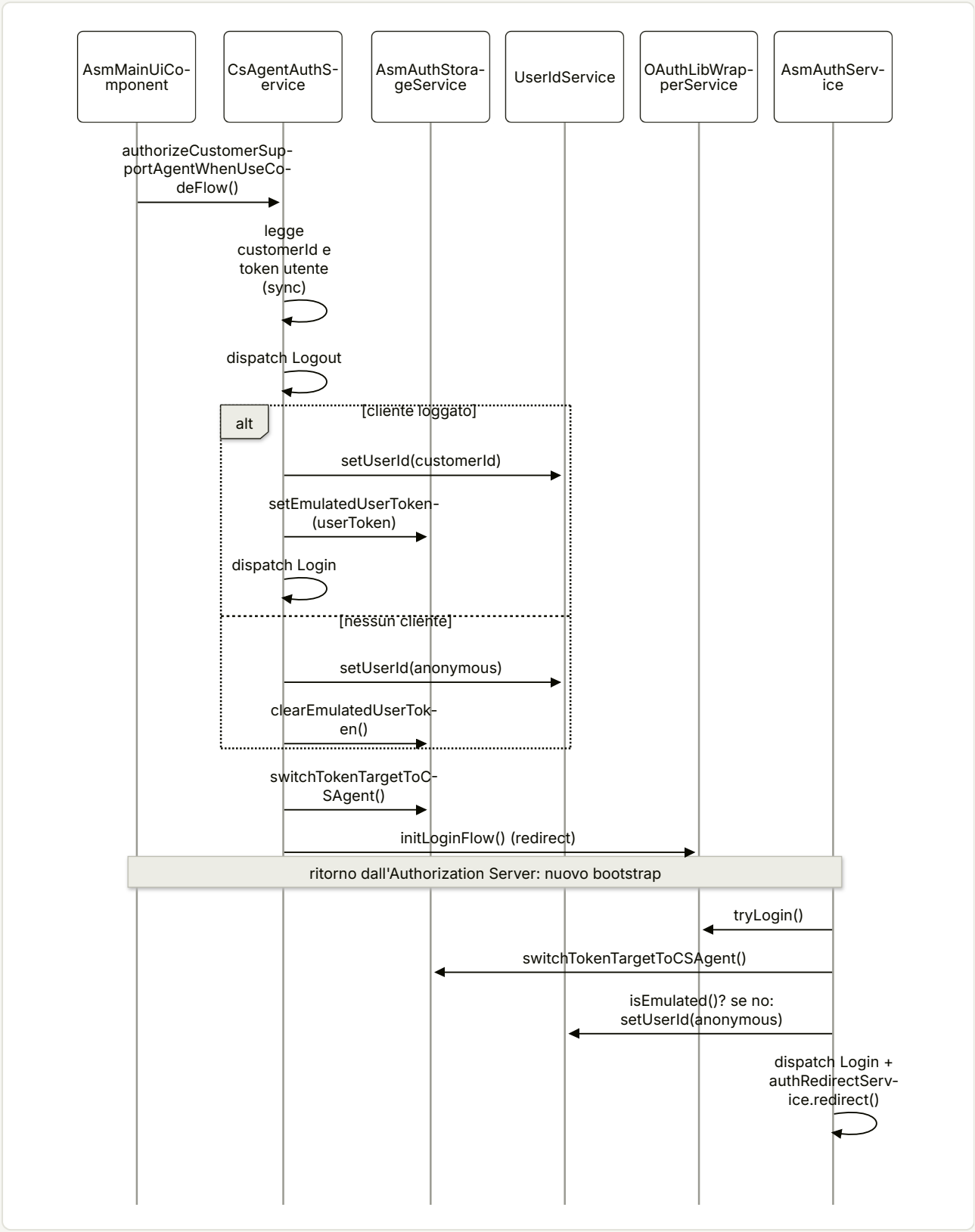
4. Diagrammi

4.1 Bootstrap e attivazione della UI ASM

■ **TAVOLA 29.1** diagramma a tutta pagina, in fondo al capitolo

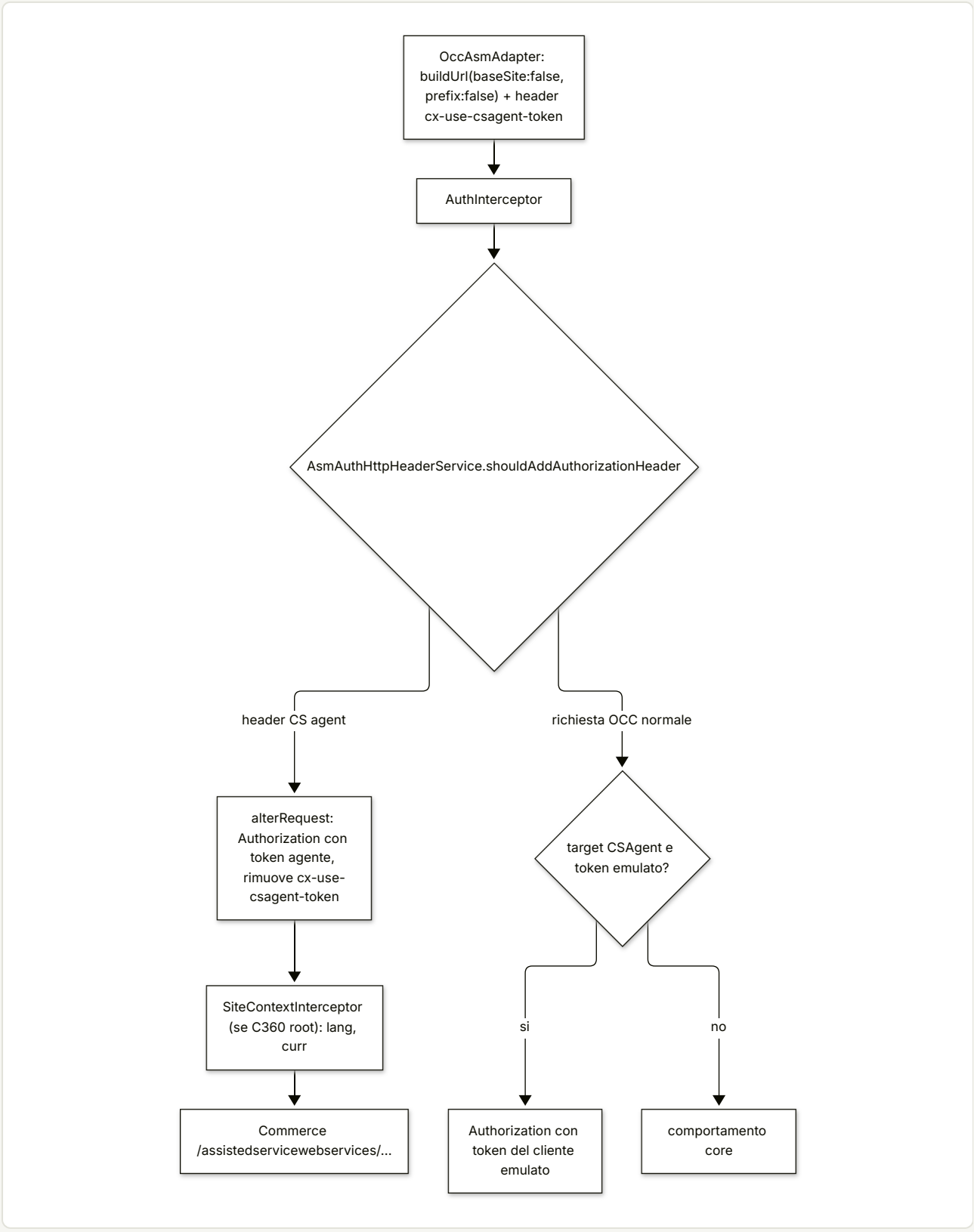
Riferimenti: projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts:L18-L27, feature-libs/asm/root/asm-root.module.ts:L65-L76, feature-libs/asm/root/services/asm-auth-storage.service.ts:L52-L69, core-libs/core/src/auth/user-auth/user-auth.module.ts:L43-L54, feature-libs/asm/root/asm-loader.module.ts:L16-L40, feature-libs/asm/root/services/asm-enabler.service.ts:L42-L88, feature-libs/asm/core/asm-core.module.ts:L15-L35, core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L45-L84.

4.2 Login agente con Authorization Code flow ed emulazione in-store



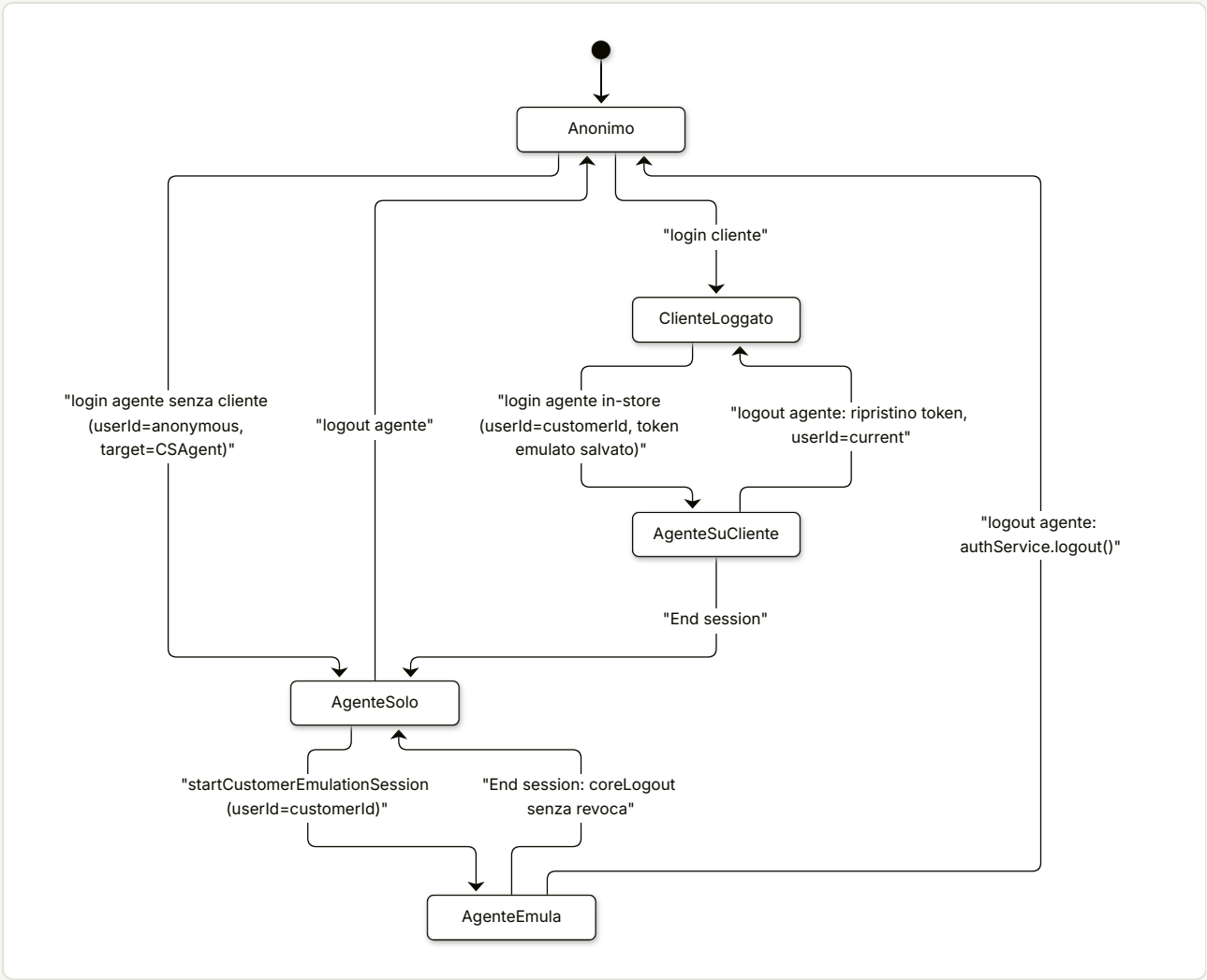
Riferimenti: `feature-libs/asm/root/services/csagent-auth.service.ts:L89-L127`, `feature-libs/asm/root/services/asm-auth.service.ts:L69-L95`.

4.3 Richiesta verso `assistedservicewebservices`



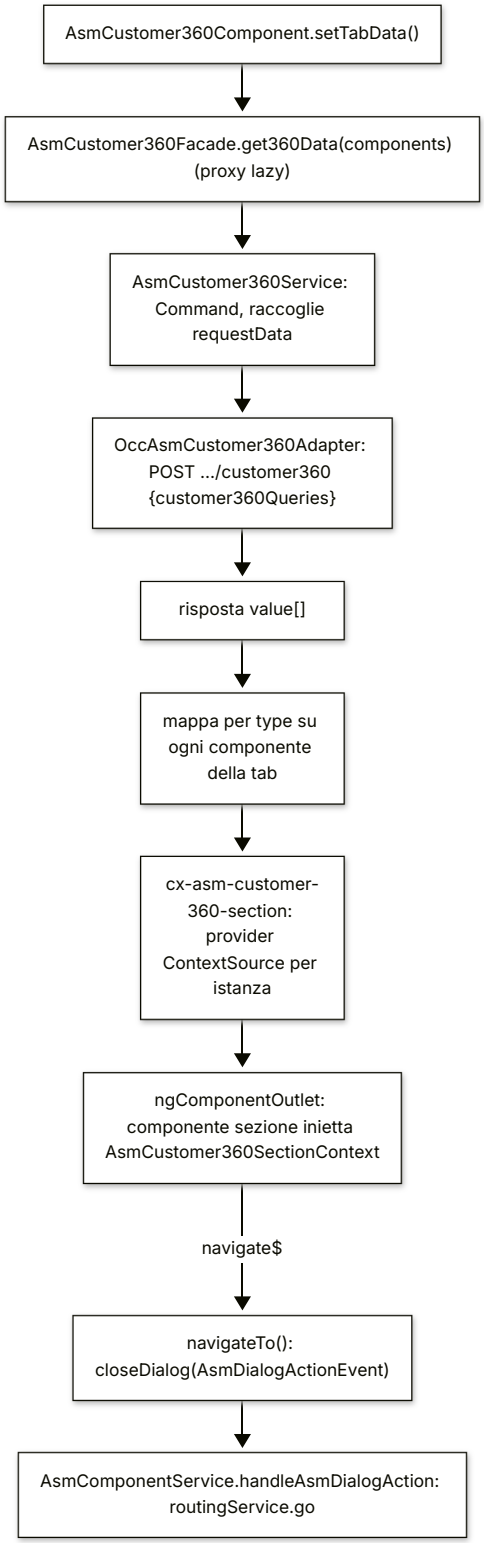
Riferimenti: `feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L99-L121`, `core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L31-L48`, `feature-libs/asm/root/services/asm-auth-http-header.service.ts:L60-L133`, `feature-libs/asm/customer-360/root/interceptors/site-context.interceptor.ts:L54-L72`. L'ordine fra interceptor nel diagramma è illustrativo (vedi NOTA in §3.8).

4.4 Stati di sessione



Riferimenti: `feature-libs/asm/root/services/csagent-auth.service.ts:L46-L84`, `feature-libs/asm/root/services/csagent-auth.service.ts:L134-L141`, `feature-libs/asm/root/services/csagent-auth.service.ts:L186-L209`, `feature-libs/asm/root/services/asm-auth.service.ts:L180-L196`. Nota: "End session" dallo stato AgenteSuCliente passa da `coreLogout()`, che pulisce il token emulato (`feature-libs/asm/root/services/asm-auth.service.ts:L185-L189`): dopo, il logout agente non può più ripristinare il cliente.

4.5 Customer 360: dalla tab alla sezione



Riferimenti: `feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L267-L294`, `feature-libs/asm/customer-360/core/services/asm-customer-360.service.ts:L33-L73`, `feature-libs/asm/customer-360/occ/adapters/occ-asm-customer-360.adapter.ts:L54-L89`, `feature-libs/`

asm/customer-360/components/sections/asm-customer-360-section/asm-customer-360-section.component.ts:L23-L75, feature-libs/asm/components/services/asm-component.service.ts:L111-L118.

---

5. Estratti di codice reale

5.1 Initializer non bloccante

feature-libs/asm/root/asm-loader.module.ts:L29-L40

```ts
/**
 *
 * We do not like to block the UI, which is why we delgate loading of ASM
 * to a real component; the router and state aren't available in an optimized
 * way during the APP_INITIALIZER.
 */
export function asmFactory(asmEnablerService: AsmEnablerService): () => void {
  const isReady = () => {
    asmEnablerService.load();
  };
  return isReady;
}
```

5.2 Le tre condizioni di attivazione e il caricamento della UI

feature-libs/asm/root/services/asm-enabler.service.ts:L42-L58

```ts
  isEnabled(): boolean {
    if (this.isLaunched() && !this.isUsedBefore()) {
      if (this.winRef.localStorage) {
        this.winRef.localStorage.setItem(ASM_ENABLED_LOCAL_STORAGE_KEY, 'true');
      }
    }
    return this.isLaunched() || this.isUsedBefore() || this.isEmulateInURL();
  }

  /**
   * Indicates whether ASM is launched through the URL,
   * using the asm flag in the URL.
   */
  protected isLaunched(): boolean {
    const params = this.location.path().split('?')[1];
    return !!params && params.split('&').includes('asm=true');
  }
```

feature-libs/asm/root/services/asm-enabler.service.ts:L84-L88

```ts
protected addUi(): void {
  this.featureModules
    .resolveFeature('asm')
    .subscribe(() => this.launchDialogService.launch(LAUNCH_CALLER.ASM));
}
```

5.3 Route senza path e override dei servizi auth

feature-libs/asm/root/asm-root.module.ts:L38-L76

```ts
  RouterModule.forChild([
    {
      // @ts-ignore
      path: null,
      canActivate: [CmsPageGuard],
      component: PageLayoutComponent,
      data: {
        cxRoute: 'asmDeepLink',
        cxCmsRouteContext: {
          id: HOME_PAGE_CONTEXT,
          type: PageType.CONTENT_PAGE,
        },
      },
    },
  ]),
],
providers: [
  provideDefaultConfig(defaultAsmConfig),
  provideDefaultConfig(<RoutingConfig>{
    routing: {
      routes: {
        asmDeepLink: {
          paths: ['assisted-service/emulate'],
        },
      },
    },
  }),
  {
    provide: AuthStorageService,
    useExisting: AsmAuthStorageService,
  },
  {
    provide: AuthService,
    useExisting: AsmAuthService,
  },
  {
```

```
    provide: AuthHttpHeaderService,
    useExisting: AsmAuthHttpHeaderService,
  },
```

5.4 Login agente con password flow

feature-libs/asm/root/services/csagent-auth.service.ts:L63-L83

```ts
  this.authStorageService.switchTokenTargetToCSAgent();
  try {
    await this.oAuthLibWrapperService.authorizeWithPasswordFlow(
      userId,
      password
    );
    this.store.dispatch(new AuthActions.Logout());

    if (customerId !== undefined && userToken !== undefined) {
      // OCC specific user id handling. Customize when implementing different backend
      this.userIdService.setUserId(customerId);
      this.authStorageService.setEmulatedUserToken(userToken);
      this.store.dispatch(new AuthActions.Login());
    } else {
      // When we can't get the customerId just end all current sessions
      this.userIdService.setUserId(OCC_USER_ID_ANONYMOUS);
      this.authStorageService.clearEmulatedUserToken();
    }
  } catch {
    this.authStorageService.switchTokenTargetToUser();
  }
```

5.5 Emulazione e logout agente

feature-libs/asm/root/services/csagent-auth.service.ts:L134-L141

```ts
public startCustomerEmulationSession(customerId: string): void {
  this.authStorageService.clearEmulatedUserToken();

  // OCC specific user id handling. Customize when implementing different backend
  this.store.dispatch(new AuthActions.Logout());
  this.userIdService.setUserId(customerId);
  this.store.dispatch(new AuthActions.Login());
}
```

feature-libs/asm/root/services/csagent-auth.service.ts:L195-L208

```ts
    await this.oAuthLibWrapperService.revokeAndLogout();

    this.store.dispatch({ type: '[Auth] Logout Customer Support Agent' });
    this.authStorageService.setTokenTarget(TokenTarget.User);

    if (isCustomerEmulated && emulatedToken) {
      this.store.dispatch(new AuthActions.Logout());
      this.authStorageService.setToken(emulatedToken);
      this.userIdService.setUserId(OCC_USER_ID_CURRENT);
      this.authStorageService.clearEmulatedUserToken();
      this.store.dispatch(new AuthActions.Login());
    } else {
      this.authService.logout();
    }
```

5.6 Ritorno dal code flow (CXSPA-13777)

feature-libs/asm/root/services/asm-auth.service.ts:L69-L95

```ts
override async checkOAuthParamsInUrl(): Promise<void> {
  const isUsingASMClient = await firstValueFrom(this.isUsingASMClient());
  if (!isUsingASMClient) {
    await super.checkOAuthParamsInUrl();
    return;
  }

  try {
    const loginResult = await this.oAuthLibWrapperService.tryLogin();
    const token = this.authStorageService.getItem('access_token');

    if (loginResult.result && token && loginResult.tokenReceived) {
      this.authStorageService.switchTokenTargetToCSAgent();
      // The customer userId is restored eagerly from storage before this handler runs
      // (AuthStatePersistenceService). Preserve an emulation started before the code-flow
      // redirect; only anonymize for a pure agent login (no emulation pending) — CXSPA-13777.
      const isEmulated = await firstValueFrom(
        this.userIdService.isEmulated()
      );
      if (!isEmulated) {
        this.userIdService.setUserId(OCC_USER_ID_ANONYMOUS);
      }
      this.store.dispatch(new AuthActions.Login());
      this.authRedirectService.redirect();
    }
  } catch {}
}
```

5.7 "Loggato" per lo storefront

feature-libs/asm/root/services/asm-auth.service.ts:L201-L214

```ts
isUserLoggedIn(): Observable<boolean> {
  return combineLatest([
    this.authStorageService.getToken(),
    this.userIdService.isEmulated(),
    this.authStorageService.getTokenTarget(),
  ]).pipe(
    map(
      ([token, isEmulated, tokenTarget]) =>
        Boolean(token?.access_token) &&
        (tokenTarget === TokenTarget.User ||
          (tokenTarget === TokenTarget.CSAgent && isEmulated))
    )
  );
}
```

5.8 Header CS agent e token emulato

feature-libs/asm/root/services/asm-auth-http-header.service.ts:L84-L133

```ts
public alterRequest(
  request: HttpRequest<any>,
  token?: AuthToken
): HttpRequest<any> {
  const hasAuthorizationHeader = !!this.getAuthorizationHeader(request);
  const isCSAgentRequest = this.isCSAgentTokenRequest(request);
  const tokenForRequest = this.getTokenForRequest(request, token);

  let req = super.alterRequest(request, tokenForRequest);

  if (!hasAuthorizationHeader && isCSAgentRequest) {
    req = request.clone({
      setHeaders: {
        ...this.createAuthorizationHeader(tokenForRequest),
      },
    });
    return InterceptorUtil.removeHeader(
      USE_CUSTOMER_SUPPORT_AGENT_TOKEN,
      req
    );
  }
  return req;
}

protected getTokenForRequest(
  request: HttpRequest<any>,
```

```
    token?: AuthToken
  ): AuthToken | undefined {
    if (this.shouldUseEmulatedUserToken(request)) {
      return this.authStorageService.getEmulatedUserToken();
    }
    return token;
  }

  protected shouldUseEmulatedUserToken(request: HttpRequest<any>): boolean {
    if (this.isCSAgentTokenRequest(request)) {
      return false;
    }

    let tokenTarget: TokenTarget | undefined;
    this.authStorageService
      .getTokenTarget()
      .subscribe((target) => (tokenTarget = target))
      .unsubscribe();

    return Boolean(
      tokenTarget === TokenTarget.CSAgent &&
        this.authStorageService.getEmulatedUserToken()?.access_token
    );
  }
```

5.9 Interceptor `sap-commerce-cloud-user-id`

feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L48-L69

```ts
    // Casting as <AsmConfig> to avoid circular dependencies with @spartacus/asm/core.
    if (!(<AsmConfig>this.config).asm?.userIdHttpHeader?.enable) {
      return next.handle(httpRequest);
    }

    const asmContext = httpRequest.context.get(OCC_HTTP_TOKEN);

    let userIdObservable: Observable<string | undefined>;

    if (typeof asmContext.sendUserIdAsHeader === 'string') {
      userIdObservable = of(asmContext.sendUserIdAsHeader);
    } else if (asmContext.sendUserIdAsHeader) {
      userIdObservable = this.userIdService
        .takeUserId()
        .pipe(
          map((userId) =>
            this.uniqueUserIdConstants.has(userId) ? undefined : userId
          )
        );
```

```
    } else {
      return next.handle(httpRequest);
    }
```

5.10 Adapter OCC: fuori dal prefisso e con marker CS agent

feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L99-L121

```ts
  protected getHeaders(): HttpHeaders {
    return InterceptorUtil.createHeader(
      USE_CUSTOMER_SUPPORT_AGENT_TOKEN,
      true,
      new HttpHeaders()
    );
  }

  customerLists(): Observable<CustomerListsPage> {
    const headers = this.getHeaders();
    const params: HttpParams = new HttpParams().set(
      'baseSite',
      this.activeBaseSite
    );

    const url = this.occEndpointsService.buildUrl(
      'asmCustomerLists',
      {},
      {
        baseSite: false,
        prefix: false,
      }
    );
```

5.11 Reducer UI che assorbe l'errore e meta-reducer di logout

feature-libs/asm/core/store/reducers/asm-ui.reducer.ts:L13-L31

```ts
export function reducer(state: AsmUi = initialState, action: Action): AsmUi {
  switch (action.type) {
    case AsmActions.ASM_UI_UPDATE: {
      return {
        ...state,
        ...(action as AsmActions.AsmUiUpdate).payload,
      };
    }
    case AsmActions.CUSTOMER_SEARCH_FAIL: {
      return {
        ...state,
```

```
        ...(action as AsmActions.CustomerSearchFail).payload,
      };
    }
    default: {
      return state;
    }
  }
}
```

feature-libs/asm/core/store/reducers/index.ts:L44-L56

```ts
export function clearCustomerSupportAgentAsmState(
  reducer: ActionReducer<AsmState, Action>
): ActionReducer<AsmState, Action> {
  return function (state: AsmState | undefined, action: Action) {
    if (action.type === AsmActions.LOGOUT_CUSTOMER_SUPPORT_AGENT) {
      state = {
        ...(state as AsmState),
        customerSearchResult: {} as StateUtils.LoaderState<CustomerSearchPage>,
      };
    }
    return reducer(state, action);
  };
}
```

5.12 Persistenza senza `refresh_token`

feature-libs/asm/core/services/asm-state-persistence.service.ts:L62-L87

```ts
  protected getAsmState(): Observable<SyncedAsmState> {
    return combineLatest([
      this.store.pipe(
        // Since getAsmState() may be called while the module is lazy loaded
        // The asm state slice may not exist yet in the first store emissions.
        filter((store) => !!store.asm),
        select(AsmSelectors.getAsmUi)
      ),
      this.authStorageService.getEmulatedUserTokenState(),
      this.authStorageService.getTokenTarget(),
    ]).pipe(
      map(([ui, emulatedUserToken, tokenTarget]) => {
        let emulatedToken = emulatedUserToken;
        if (emulatedToken) {
          emulatedToken = { ...emulatedUserToken } as AuthToken;
          // To minimize risk of user account hijacking we don't persist emulated user refresh_token
          delete emulatedToken.refresh_token;
        }
        return {
```

```
        ui,
        emulatedUserToken: emulatedToken,
        tokenTarget,
      };
    })
  );
}
```

5.13 Customer 360: una richiesta per tab

feature-libs/asm/customer-360/core/services/asm-customer-360.service.ts:L33-L66

```ts
this.asmCustomer360Command$ = this.commandService.create(
  (tabComponents) => {
    return this.userAccountFacade.get().pipe(
      take(1),
      concatMap((customer) => {
        const queries = tabComponents.reduce(
          (requests: Array<AsmCustomer360Query>, component) => {
            if (component.requestData) {
              return requests.concat(component.requestData);
            }
            return requests;
          },
          []
        );

        if (queries.length > 0) {
          const request: AsmCustomer360Request = {
            queries,
            options: {
              userId: customer?.customerId ?? '',
            },
          };
          return this.asmCustomer360Connector.getAsmCustomer360Data(
            request
          );
        } else {
          return of({
            value: [],
          });
        }
      })
    );
  }
);
```

5.14 Host di sezione con context per istanza

feature-libs/asm/customer-360/components/sections/asm-customer-360-section/asm-customer-360-section.component.ts:L23-L52

```ts
@Component({
  selector: 'cx-asm-customer-360-section',
  templateUrl: './asm-customer-360-section.component.html',
  providers: [
    AsmCustomer360SectionContextSource,
    {
      provide: AsmCustomer360SectionContext,
      useExisting: AsmCustomer360SectionContextSource,
    },
  ],
  imports: [NgComponentOutlet],
})
export class AsmCustomer360SectionComponent implements OnDestroy {
  @Input()
  component: Type<unknown>;

  @Input()
  set customer(customer: User) {
    this.source.customer$.next(customer);
  }

  @Input()
  set config(config: AsmCustomer360SectionConfig) {
    this.source.config$.next(config);
  }

  @Input()
  set data(data: Observable<unknown>) {
    this.source.data$.next(data);
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
asm.agentSessionTimer.startingDelayInSeconds	number	600 (max 60000)	feature-libs/asm/root/config/asm-config.ts:L17-L19; default feature-libs/asm/root/config/default-asm-config.ts:L13-L15	feature-libs/asm/components/asm-session-timer/asm-session-timer.component.ts:L83-L97
asm.customerSearch.maxResults	number	20	feature-libs/asm/root/config/asm-config.ts:L20-L22; feature-libs/asm/root/config/default-asm-config.ts:L16-L18	feature-libs/asm/components/customer-selection/customer-selection.component.ts:L166-L170

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
asm.customerList.pageSize	number	5 (fallback 5)	feature-libs/asm/root/config/asm-config.ts:L24-L24; feature-libs/asm/root/config/default-asm-config.ts:L21-L21	feature-libs/asm/components/customer-list/customer-list.component.ts:L154-L155
asm.customerList.showAvatar	boolean	true	feature-libs/asm/root/config/asm-config.ts:L25-L25; feature-libs/asm/root/config/default-asm-config.ts:L22-L22	feature-libs/asm/components/customer-list/customer-list.component.html:L289-L289
asm.customerList.columns[]	{header LocalizationKey, icon?, renderer?, actionType?}[]	7 colonne	feature-libs/asm/root/config/asm-config.ts:L26-L34; feature-libs/asm/root/config/default-asm-config.ts:L23-L75	feature-libs/asm/components/customer-list/customer-list.component.ts:L156-L156, feature-libs/asm/components/customer-list/customer-list.component.ts:L257-L279
asm.asmSessionSupport.enabled	boolean	false	feature-libs/asm/root/config/asm-config.ts:L36-L38; feature-libs/asm/root/config/default-asm-config.ts:L19-L19	feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L151-L152
asm.userIdHttpHeader.enable	boolean	true	feature-libs/asm/root/config/asm-config.ts:L39-L50; feature-libs/asm/root/config/default-asm-config.ts:L77-L79	feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L49-L51
asm.createCustomer.enable	boolean	true	feature-libs/asm/root/config/asm-config.ts:L51-L53; feature-libs/asm/root/config/default-asm-config.ts:L80-L82	solo template: feature-libs/asm/components/customer-list/customer-list.component.html:L487-L487, feature-libs/asm/components/customer-selection/customer-selection.component.html:L41-L41
routing.routes.asmDeepLink.paths	string[]	['assisted-service/emulate']	feature-libs/asm/root/asm-root.module.ts:L56-L64	route cxRoute: 'asmDeepLink' in feature-libs/asm/root/asm-root.module.ts:L38-L51
backend.occ.endpoints.asmCustomerSearch / asmCustomerLists / asmBindCart / asmCreateCustomer / asmSessionEvent	string | OccEndpoint	/assistedservicewebservices/... (vedi §5)	feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L9-L22; tipi feature-libs/asm/occ/model/occ-asm-endpoints.model.ts:L9-L22	feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L78-L90, feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L114-L121, feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L169-L176, feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L203-L210, feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L226-L233

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.endpoints.asmCustomer360	string | OccEndpoint	/assistedservicewebservices/${baseSiteId}/users/${userId}/customer360	feature-libs/asm/customer-360/occ/adapters/default-occ-asm-customer-360-config.ts:L9-L18	feature-libs/asm/customer-360/occ/adapters/occ-asm-customer-360.adapter.ts:L63-L75
launch.ASM	LayoutConfig	outlet cx-storefront, AsmMainUiComponent	feature-libs/asm/components/default-asm-layout.config.ts:L10-L17	feature-libs/asm/root/services/asm-enabler.service.ts:L84-L88
launch.ASM_CUSTOMER_LIST / ASM_BIND_CART / ASM_SAVE_CART / ASM_SWITCH_CUSTOMER / ASM_CREATE_CUSTOMER_FORM	LayoutConfig	dialog inlineRoot	feature-libs/asm/components/default-bind-cart-layout.config.ts:L14-L22, feature-libs/asm/components/default-save-cart-layout.config.ts:L14-L22, feature-libs/asm/components/default-switch-customer-layout.config.ts:L14-L22 (registrate in feature-libs/asm/components/asm-components.module.ts:L98-L104), feature-libs/asm/components/customer-list/default-customer-list-layout.config.ts:L10-L18, feature-libs/asm/components/asm-create-customer-form/default-asm-create-customer-form-layout.config.ts:L10-L18	openDialogAndSubscribe nei componenti
launch.ASM_CUSTOMER_360	LayoutConfig	dialog inlineRoot AsmCustomer360Component	feature-libs/asm/customer-360/components/default-asm-customer-360-layout.config.ts:L14-L22	feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L256-L266, feature-libs/asm/components/customer-emulation/customer-emulation.component.ts:L73-L94
pagination	PaginationConfig	rangeCount: 2, addStart/addEnd: false, addFirst/addLast/addDots: true, dotsLabel: '···'	feature-libs/asm/components/default-asm-pagination.config.ts:L9-L22	core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L370-L375 (chiave **globale**, vedi §9)
asmCustomer360.tabs / dateFormat / dateTimeFormat	AsmCustomer360TabsConfig	6 tab, 'MM-dd-yyyy', 'dd-MM-yy hh:mm a'	feature-libs/asm/customer-360/components/config/default-asm-customer-360-config.ts:L21-L142	feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L142-L143
featureModules.asmCustomer360.dependencies	string[]	[ASM_FEATURE]	feature-libs/asm/customer-360/root/asm-customer-360-root.module.ts:L21-L28	core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L116
toggle authorizationCodeFlowByDefault	boolean	true	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L161-L161, default core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029	feature-libs/asm/components/asm-main-ui/asm-main-ui.component.html:L45-L63, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L316-L327

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `enableExpiredRefreshTokenHandlers`	boolean	true	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L464-L464`, default `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1072-L1072`	`core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L266-L266`
localStorage `asm_enabled`	`'true'`	—	`feature-libs/asm/root/asm-constants.ts:L7-L7`	`feature-libs/asm/root/services/asm-enabler.service.ts:L42-L79`, `feature-libs/asm/components/services/asm-component.service.ts:L83-L87`, `core-libs/core/src/auth/user-auth/facade/auth.service.ts:L263-L295`
storage `asm` (StatePersistence)	`{ui, emulatedUserToken, tokenTarget}`	—	`feature-libs/asm/core/services/asm-state-persistence.service.ts:L43-L43`	`feature-libs/asm/root/services/asm-auth-storage.service.ts:L57-L61`
chunk i18n `asm`, `asmCustomer360`	`TranslationChunksConfig`	`['asm']`, `['asmCustomer360']`	`feature-libs/asm/assets/translations/translations.ts:L9-L29`, `feature-libs/asm/customer-360/assets/translations/translations.ts:L9-L11`	config `i18n.chunks` dell'app (`projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts:L28-L38`)

7. Commenti e note notevoli dal sorgente

- "The ASM loader module takes care of loading the ASM UI only in case there's a reason to do so." e la motivazione del non-blocco del bootstrap — `feature-libs/asm/root/asm-loader.module.ts:L12-L34`.
- "This service is added to avoid any polution of the UI and runtime performance for the ordinary production user." — `feature-libs/asm/root/services/asm-enabler.service.ts:L13-L17`.
- "The providers below preserve backward compatibility for the CXSPA-12514 fix and should be removed once the enableExpiredRefreshTokenHandlers feature toggle is fully rolled out." — `feature-libs/asm/root/asm-root.module.ts:L77-L79`.
- "// OCC specific user id handling. Customize when implementing different backend" (tre occorrenze): l'emulazione è un dettaglio OCC — `feature-libs/asm/root/services/csagent-auth.service.ts:L72-L72`, `feature-libs/asm/root/services/csagent-auth.service.ts:L110-L110`, `feature-libs/asm/root/services/csagent-auth.service.ts:L137-L137`.
- "isEmulated is set to true before switchTokenTargetToCSAgent() so that isUserLoggedIn() stays true throughout and no spurious LogoutEvent fires." — `feature-libs/asm/root/services/csagent-auth.service.ts:L111-L112`.
- "// TODO(#8248): Create new loading state outside of store": `getCustomerSupportAgentTokenLoading()` ritorna sempre `of(false)` — `feature-libs/asm/root/services/csagent-auth.service.ts:L178-L181`.
- CXSPA-13777: "Preserve an emulation started before the code-flow redirect; only anonymize for a pure agent login" — `feature-libs/asm/root/services/asm-auth.service.ts:L82-L84`.

- Refuso nel JSDoc di `AsmAuthService`: "working for both user na CS agent" — `feature-libs/asm/root/services/asm-auth.service.ts:L34-L37`.
- "When we start emulation from the UI (not by ASM login) we can't restore user session on cs agent logout. Only available solution is to drop session we could restore, to avoid account hijack." — `feature-libs/asm/root/services/asm-auth-storage.service.ts:L132-L138`.
- "To minimize risk of user account hijacking we don't persist emulated user refresh_token" — `feature-libs/asm/core/services/asm-state-persistence.service.ts:L77-L77`.
- `AsmComponentService.unload()` ammette che rimuove solo lo storage e non la UI dal DOM perché "outlets currently do not support this" — `feature-libs/asm/components/services/asm-component.service.ts:L76-L82`.
- "//Always route to home page to avoid 404" — `feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L284-L285`.
- "// TODO(#12660): Remove default value once backend is updated" (descrizione `'-'` del saved cart creato dal bind) — `feature-libs/asm/components/asm-bind-cart/asm-bind-cart.component.ts:L279-L280`.
- Il JSDoc dell'interceptor cita "OCC_ASM_TOKEN", ma il codice usa `OCC_HTTP_TOKEN` — `feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L25-L28`.
- JSDoc di `AsmService.createAsmSessionEvent` copiato erroneamente ("Get the state of the ASM UI") — `feature-libs/asm/core/facade/asm.service.ts:L71-L78`.
- L'azione di logout agente dice "Used to clear store data (ui, search results)", ma il meta-reducer pulisce solo i risultati di ricerca — `feature-libs/asm/core/store/actions/logout-agent.action.ts:L12-L14`, `feature-libs/asm/core/store/reducers/index.ts:L48-L53`.

---

8. Extension point

EXTENSION POINT	COME	EXPORT	RICETTA
`AsmConfig` (timer, ricerca, colonne customer list, header user id, session event, create customer)	`provideConfig({ asm: {...} })`	`@spartacus/asm/root` (`feature-libs/asm/root/public_api.ts:L7-L7`)	50 — Ricette config e DI
Servizi auth ASM (`AsmAuthService`, `AsmAuthStorageService`, `AsmAuthHttpHeaderService`, `CsAgentAuthService`, `AsmEnablerService`, `AsmDeepLinkService`, `AsmProtectedRoutesService`)	sottoclasse + provider (vedi nota sotto)	`@spartacus/asm/root` (`feature-libs/asm/root/services/index.ts:L7-L13`)	50 — Ricette config e DI
`AsmAdapter` / `OccAsmAdapter`, normalizer `CUSTOMER_SEARCH_PAGE_NORMALIZER`, `CUSTOMER_LISTS_NORMALIZER`, endpoint `asm*`	adapter custom, converter multi, override `backend.occ.endpoints`	`@spartacus/asm/core` (`feature-libs/asm/core/connectors/asm.adapter.ts:L18-L47`, `feature-libs/asm/core/connectors/converters.ts:L11-L17`), `@spartacus/asm/occ`	52 — Ricette OCC
Layout `launch.ASM*` (outlet/dialog dei componenti)	`provideConfig(<LayoutConfig>{ launch: {...} })`	via config	51 — Ricette CMS, outlet, layout

EXTENSION POINT	COME	EXPORT	RICETTA
Componenti UI ASM e `AsmComponentService`	sostituzione del componente nella config `launch`	`@spartacus/asm/components` (feature-libs/asm/components/public_api.ts:L7-L22)	51 — Ricette CMS, outlet, layout
Tab e sezioni Customer 360	`provideConfig({ asmCustomer360: { tabs: [...] } })`, `requestData.type` stringa custom	`@spartacus/asm/customer-360/root` e `/components` (feature-libs/asm/customer-360/components/public_api.ts:L7-L11)	53 — Feature lazy, i18n
`AsmCustomer360Adapter` / `ASM_CUSTOMER_360_NORMALIZER`	adapter o converter custom	`@spartacus/asm/customer-360/core`	52 — Ricette OCC
Traduzioni `asm`, `asmCustomer360`	`i18n.resources` + chunk	`@spartacus/asm/assets`, `@spartacus/asm/customer-360/assets`	53 — Feature lazy, i18n
Store (`AsmActions`, `AsmSelectors`, `StateWithAsm`)	dispatch/select	`@spartacus/asm/core` (feature-libs/asm/core/public_api.ts:L11-L13); effects e reducer **non** esportati	54 — Upgrade-safe

Override dei servizi auth. `AsmRootModule` registra `{ provide: AuthService, useExisting: AsmAuthService }` ecc. (feature-libs/asm/root/asm-root.module.ts:L65-L87). Per sostituire, ad esempio, `AsmAuthService` con una sottoclasse conviene fornire il proprio servizio sul token `AsmAuthService` (così anche il `useExisting` del core lo segue) — esempio illustrativo, non presente nel repo:

```ts
// esempio (non nel repo): simboli esportati da @spartacus/asm/root
@Injectable({ providedIn: 'root' })
export class MyAsmAuthService extends AsmAuthService { /* ... */ }

providers: [{ provide: AsmAuthService, useExisting: MyAsmAuthService }]
```

> DA VERIFICARE — L'effetto dipende dall'ordine dei provider nell'injector root (il proprio provider deve vincere su quello `providedIn: 'root'` di `AsmAuthService`); non è stato verificato in un'app in esecuzione (`node_modules` assente).

Colonne della customer list. `asm.customerList.columns` è un array (per la semantica del merge di array nella config vedi 11 — Configurazione); i tipi di azione disponibili sono quelli di `CustomerListColumnActionType` (feature-libs/asm/root/model/customer-list.model.ts:L28-L33), interpretati in `AsmMainUiComponent` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L190-L233).

Sezioni Customer 360 custom: limite. Una sezione custom dovrebbe iniettare `AsmCustomer360SectionContext`, ma la classe **non è esportata**: l'index delle sezioni esporta solo le 11 sezioni (feature-libs/asm/customer-360/components/sections/index.ts:L7-L17). Si può configurare una tab che riusa le sezioni standard, oppure un componente che non usa il context (ricevendo però i dati solo tramite il context, cioè di fatto nulla). Questo è un vincolo di API pubblica: vedi 54 — API pubblica vs interna.

Interceptor non esportati. `UserIdHttpHeaderInterceptor` e `SiteContextInterceptor` non sono esportati dai rispettivi `public_api.ts` (feature-libs/asm/root/public_api.ts:L7-L14, feature-libs/asm/customer-360/root/public_api.ts:L7-L10): per disattivare l'header user id si usa la config `asm.userIdHttpHeader.enable: false`.

9. Trappole, casi limite, comportamenti controintuitivi

1. **Due copie della logica "ASM abilitato".** `AsmEnablerService` (root) e `AuthService` (core) duplicano letterali `'asm_enabled'`, `'asm=true'`, `'assisted-service/emulate?'`: cambiarne una non cambia l'altra — feature-libs/asm/root/services/asm-enabler.service.ts:L42-L79, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L263-L295.
2. `asm=true` **esatto.** `asm=TRUE` o `?asm` non attivano — feature-libs/asm/root/services/asm-enabler.service.ts:L55-L58.
3. **La route del deep link ha** `path: null`**.** Il path vero viene dalla `RoutingConfig` `asmDeepLink`: cercarlo nella definizione Angular è inutile — feature-libs/asm/root/asm-root.module.ts:L38-L51.
4. **Il 403 arriva alla UI passando per il reducer** `asmUi`**.** `CustomerSearchFail` fonde l'intero errore nello stato UI; `AsmMainUiComponent` vede `status: 403`, fa logout e lo azzera — feature-libs/asm/core/store/reducers/asm-ui.reducer.ts:L21-L26, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L183-L188. Insieme alla ricerca fittizia `autoSearchToAvoidUnauthorizedLogin` è il meccanismo che espelle chi non è agente (§3.13).
5. **Default** `pagination` **globale.** `AsmComponentsModule` fornisce `defaultAsmPaginationConfig` con `provideDefaultConfig` (feature-libs/asm/components/asm-components.module.ts:L97-L105), cioè un `DefaultConfigChunk` (core-libs/core/src/config/config-providers.ts:L55-L61). Al lazy load, `ConfigurationService` lo fonde nella config ambient e poi in-place nella config globale, sopra i default root e sotto la config dell'app (core-libs/core/src/config/services/configuration.service.ts:L76-L106). Il default storefront ha `addStart: true, addEnd: true` (core-libs/storefront/shared/components/list-navigation/pagination/config/default-pagination.config.ts:L9-L14), ASM `false`; `PaginationConfig` è `useExisting: Config` e il builder rilegge la config a ogni accesso (core-libs/storefront/shared/components/list-navigation/pagination/config/pagination.config.ts:L11-L17, core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L370-L375). Dedotto dal codice: dopo l'apertura della barra ASM anche le paginazioni non-ASM cambiano aspetto, a meno che l'app configuri `pagination` esplicitamente. `DA VERIFICARE` a runtime (dipende anche da `disableConfigUpdates`, core-libs/core/src/config/services/configuration.service.ts:L102-L105).
6. **Meta-reducer parziale.** Il logout agente azzera solo `customerSearchResult`, non la lista clienti né `asmUi`, malgrado il JSDoc — feature-libs/asm/core/store/reducers/index.ts:L44-L56, feature-libs/asm/core/store/actions/logout-agent.action.ts:L12-L14.
7. **Azione come oggetto letterale.** `CsAgentAuthService` dispatcha `{ type: '[Auth] Logout Customer Support Agent' }` invece della classe `LogoutCustomerSupportAgent` (presumibilmente per non far dipendere il root dal core ASM, come dichiara il commento sulle dipendenze circolari in feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L48-L48); il meta-reducer confronta solo `type` — feature-libs/asm/root/services/csagent-auth.service.ts:L197-L197, feature-libs/asm/core/store/actions/logout-agent.action.ts:L9-L17.
8. **Bind cart con** `uid`**, non** `customerId` — feature-libs/asm/core/facade/asm-bind-cart.service.ts:L26-L38. E `AsmBindCartService` non è esportato da `@spartacus/asm/core` — feature-libs/asm/core/facade/index.ts:L7-L9.
9. `OccAsmAdapter` **ignora** `currentPage` (usa solo `page`) e non usa la `AsmConfig` iniettata — feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L131-L176, feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L54-L60.
10. **Base site e user id "ultimo noto".** `OccAsmAdapter` e `OccAsmCustomer360Adapter` memorizzano `activeBaseSite`/`currentUserId` da sottoscrizioni aperte nel costruttore e mai chiuse — feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L61-L67, feature-libs/asm/customer-360/occ/adapters/occ-asm-customer-360.adapter.ts:L41-L43.

11. **Endpoint senza slash.** `asmSessionEvent` non inizia con `/`, a differenza degli altri — feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L13-L18.
12. **Mutazione della config globale.** `CustomerListComponent.updateCustomerListColumns()` modifica in-place gli oggetti colonna (`headerLocalizationKey = 'hideHeaders'`), e la modifica persiste fra aperture del dialog — feature-libs/asm/components/customer-list/customer-list.component.ts:L257-L279.
13. **Context C360 mai alimentato in parte.** Il padre non lega l'input `customer` della sezione, quindi `context.customer$` non emette; `savedCarts$`, `activeCart$`, `orderHistory$` non sono mai alimentati — feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.html:L198-L204, feature-libs/asm/customer-360/components/sections/asm-customer-360-section-context-source.model.ts:L20-L32. Inoltre `data` è tipato `Observable<unknown>` ma riceve un oggetto — feature-libs/asm/customer-360/components/sections/asm-customer-360-section/asm-customer-360-section.component.ts:L49-L52.
14. **Chiave i18n con case diverso.** Il template usa `asmCustomer360.header.newSubtitle`, il JSON en ha `newSubTitle` — feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.html:L15-L19, feature-libs/asm/customer-360/assets/translations/en/asmCustomer360.json:L8-L10.
15. **Letture sincrone con `subscribe().unsubscribe()`.** Funzionano solo perché le sorgenti sono `BehaviorSubject`/store sincroni; un override che le rende asincrone rompe login e logout — feature-libs/asm/root/services/csagent-auth.service.ts:L53-L61, feature-libs/asm/root/services/asm-auth.service.ts:L98-L111.
16. **Errori inghiottiti.** `AsmAuthService.checkOAuthParamsInUrl()` ha un `catch {}` vuoto: un fallimento del ritorno dal code flow ASM non lascia traccia — feature-libs/asm/root/services/asm-auth.service.ts:L76-L94.
17. **"End session" dopo un login in-store perde il cliente.** `coreLogout()` in emulazione pulisce il token emulato, per cui il successivo logout agente non può più ripristinare la sessione del cliente — feature-libs/asm/root/services/asm-auth.service.ts:L185-L189, feature-libs/asm/root/services/csagent-auth.service.ts:L200-L208.
18. **Sottoscrizioni che si accumulano.** `showC360Dialog()` aggiunge una nuova sottoscrizione a `dialogClose` a ogni apertura; la sottoscrizione al 403 in `ngOnInit` non è mai chiusa — feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L256-L275, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L183-L189.
19. **Timer che non riparte.** Il reset del timer di sessione avviene solo se `timeLeft > 0` — feature-libs/asm/components/asm-session-timer/asm-session-timer.component.ts:L77-L81.
20. **La UI non sparisce davvero.** `hideUi()` aggiunge la classe `hidden` e rimuove lo storage, ma il componente resta nell'outlet — feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L420-L425, feature-libs/asm/components/services/asm-component.service.ts:L76-L82.
21. **Versione nel `package.json`.** `feature-libs/asm/package.json` dichiara `221121.22.0`, non `2612.0.0` — feature-libs/asm/package.json:L3-L3.

---

10. Come fare debug

- **La barra non compare.** Controlla le tre condizioni (§3.2): `asm=true` esatto nella query, `localStorage.asm_enabled`, URL di deep link. Verifica che la feature `asm` sia configurata in `featureModules` (projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts:L21-L27) e che esista `launch.ASM` (in dev mode `LaunchDialogService` logga "No configuration provided for caller", core-libs/storefront/

layout/launch-dialog/services/launch-dialog.service.ts:L94-L96). Per "dimenticare" ASM: rimuovi `asm_enabled` e la chiave `asm` dallo storage.

- **Il login agente porta al login cliente (o viceversa).** Verifica il valore di `authorizationCodeFlowByDefault` e se, dopo il redirect, `isUsingASMClient` è vero: lo imposta `refreshAuthConfig()` solo se il core rileva ASM con la propria logica (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L263-L327). Con il client ASM, `AsmAuthService.checkOAuthParamsInUrl()` è l'unico che processa il callback e inghiotte gli errori (feature-libs/asm/root/services/asm-auth.service.ts:L76-L94): un breakpoint lì è il primo passo.
- **Lo stato della sessione.** Con un breakpoint in `AsmAuthStorageService` leggi `_tokenTarget$.value` e `getEmulatedUserToken()`; in `UserIdService` lo `userId` (`anonymous`/`current`/customerId). La tabella della §3.5 dice cosa ci si deve aspettare in ogni stato.
- **Richieste ASM 401/CORS.** Nel tab Network le chiamate a `/assistedservicewebservices/` devono avere `Authorization` e **non** avere `cx-use-csagent-token` (rimosso in feature-libs/asm/root/services/asm-auth-http-header.service.ts:L94-L104). Se compare `sap-commerce-cloud-user-id` e il backend lo rifiuta in CORS, disattiva `asm.userIdHttpHeader.enable` (feature-libs/asm/root/config/asm-config.ts:L39-L50). Con C360 configurato le stesse richieste hanno `lang` e `curr` in query.
- **Logout "inspiegabile" subito dopo il login.** Probabile 403 sulla ricerca fittizia (`autoSearchToAvoidUnauthorizedLogin`, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L235-L251): l'utente non ha il gruppo agente in Commerce. Con Redux DevTools cerca `[Asm] Customer Search Fail`: nel diff della stessa azione `asm.asmUi.status` diventa `403` (feature-libs/asm/core/store/reducers/asm-ui.reducer.ts:L21-L26); subito dopo `AsmMainUiComponent` fa logout e dispatcha `[Asm] UI Update` con `status: undefined` (feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L183-L188).
- **Store.** Slice `asm` (`customerSearchResult`, `customerListCustomersSearchResult`, `asmUi`), azioni di ricerca `[Asm] ...` (feature-libs/asm/core/store/actions/customer.action.ts:L14-L26) e `'[Auth] Logout Customer Support Agent'` (feature-libs/asm/core/store/actions/logout-agent.action.ts:L9-L10). La slice esiste solo dopo il lazy load.
- **Customer 360 vuoto.** Verifica che la feature `asmCustomer360` sia configurata (altrimenti il pulsante non appare e la colonna è nascosta: feature-libs/asm/components/customer-list/customer-list.component.ts:L234-L279), che il body della POST `.../customer360` contenga i `type` attesi e che la risposta li restituisca con lo stesso `type` (il match è per uguaglianza, feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L282-L291).

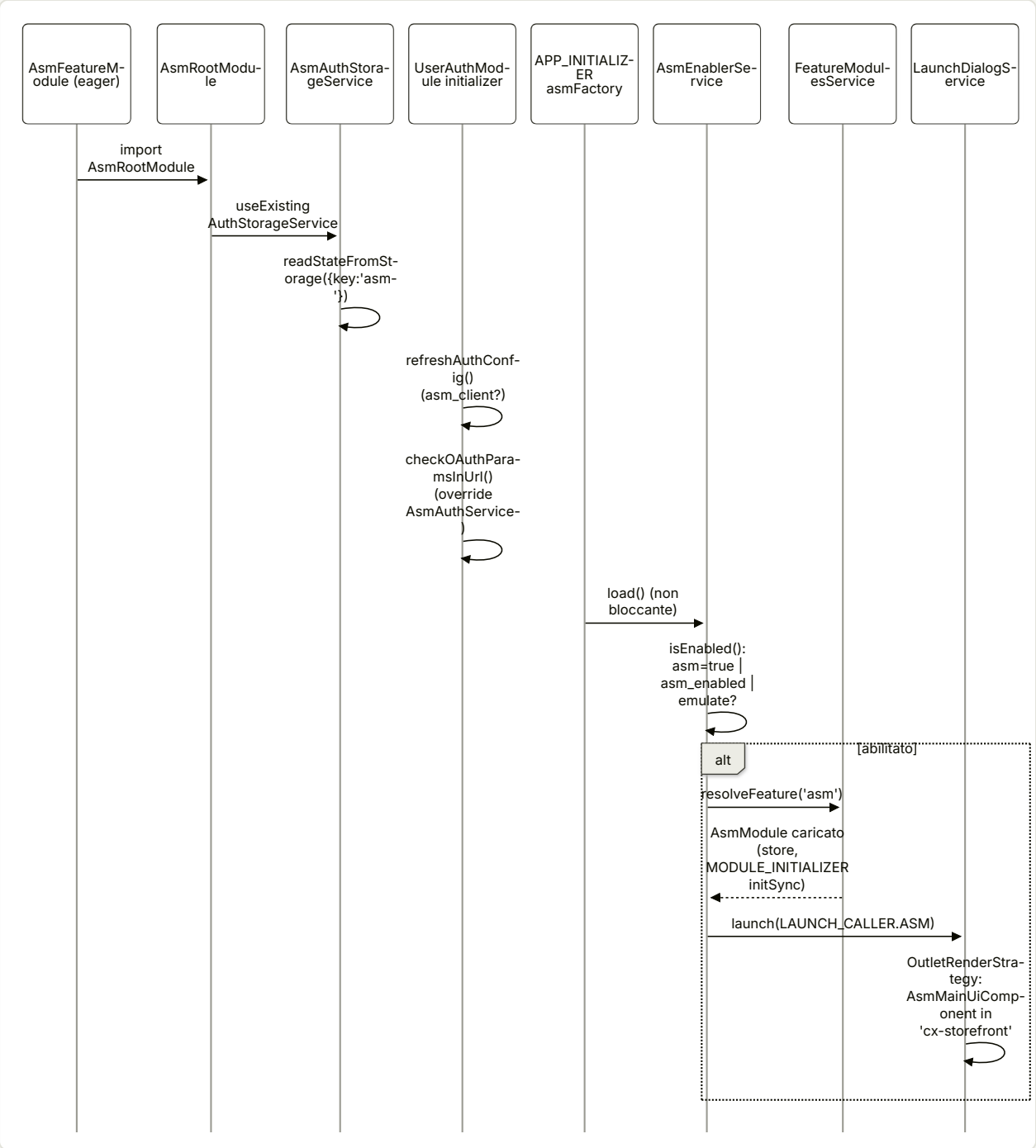
---

11. Domande di autoverifica

1. **Perché `AsmRootModule` è eager se tutta la UI è lazy?** Perché deve sostituire i servizi auth del core (`AuthStorageService`, `AuthService`, `AuthHttpHeaderService`, `ProtectedRoutesService`) e registrare l'interceptor prima di qualsiasi richiesta, oltre a fornire config di default e route del deep link — feature-libs/asm/root/asm-root.module.ts:L34-L95.
2. **Quali sono le tre condizioni che attivano la barra ASM?** `asm=true` come parametro esatto, `localStorage.asm_enabled === 'true'`, URL contenente `assisted-service/emulate?` — feature-libs/asm/root/services/asm-enabler.service.ts:L42-L79.
3. **Come fa lo storefront a considerare "loggato" un cliente mentre il token è dell'agente?** `AsmAuthService.isUserLoggedIn()` è vero anche con target `CSAgent` se lo `userId` è emulato (né `anonymous` né `current`) — feature-libs/asm/root/services/asm-auth.service.ts:L201-L214, core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L81-L88.

4. **Che cosa succede alle richieste OCC "normali" se l'agente fa login mentre un cliente è già loggato?** Con target `CSAgent` e un token emulato con `access_token`, `AsmAuthHttpHeaderService` usa il token del cliente per le richieste non marcate CS agent — feature-libs/asm/root/services/asm-auth-http-header.service.ts:L108-L133.
5. **Perché nel code flow lo stato di emulazione è persistito prima di `initLoginFlow()`?** Perché `initLoginFlow()` fa un redirect a pagina intera e il codice successivo non gira in modo affidabile; il token agente arriva poi in `AsmAuthService.checkOAuthParamsInUrl()`, che non anonimizza se un'emulazione è pendente — feature-libs/asm/root/services/csagent-auth.service.ts:L103-L123, feature-libs/asm/root/services/asm-auth.service.ts:L80-L91.
6. **Come viene espulso dalla barra un utente autenticato che non è un agente?** Dopo il login la barra lancia una ricerca fittizia; il 403 produce `CustomerSearchFail`, il reducer `asmUi` fonde `status: 403` nello stato UI e `AsmMainUiComponent` fa logout e `hideUi()` — feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L235-L251, feature-libs/asm/core/store/reducers/asm-ui.reducer.ts:L21-L26, feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L183-L189.
7. **Perché il `refresh_token` del cliente emulato non sopravvive a un reload?** `AsmStatePersistenceService` lo cancella prima di persistere, per ridurre il rischio di account hijacking — feature-libs/asm/core/services/asm-state-persistence.service.ts:L74-L79.
8. **Quante richieste HTTP fa Customer 360 per una tab con tre sezioni?** Una: `AsmCustomer360Service` aggrega i `requestData` di tutti i componenti della tab in un'unica POST `customer360Queries` (l'overview è caricata a parte) — feature-libs/asm/customer-360/core/services/asm-customer-360.service.ts:L33-L73, feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L116-L164.
9. **Come riceve i dati un componente di sezione C360 e perché è difficile scriverne uno custom?** L'host `cx-asm-customer-360-section` fornisce per istanza un `AsmCustomer360SectionContextSource` registrato anche come `AsmCustomer360SectionContext`; la sezione lo inietta. Ma `AsmCustomer360SectionContext` non è esportato dall'API pubblica — feature-libs/asm/customer-360/components/sections/asm-customer-360-section/asm-customer-360-section.component.ts:L23-L52, feature-libs/asm/customer-360/components/sections/index.ts:L7-L17.
10. **Perché le URL ASM non contengono `/occ/v2/<site>`?** L'adapter le costruisce con `{ baseSite: false, prefix: false }`; il base site passa come query param `baseSite` (search, liste, bind cart, create customer) oppure come segmento di path `${baseSiteId}` (`asmSessionEvent`, `asmCustomer360`). Gli endpoint di default iniziano con `/assistedservicewebservices/`, tranne `asmSessionEvent`, che è senza slash iniziale — feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L107-L121, feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L78-L90, feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L9-L22, feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L17-L18, feature-libs/asm/customer-360/occ/adapters/default-occ-asm-customer-360-config.ts:L13-L14.

TAVOLA 29.1 **4.1 Bootstrap e attivazione della UI ASM**



30

Product configurator

Prodotti configurabili: pagina di configurazione, conflitti, prezzi, overview.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Un prodotto *configurabile* è un prodotto il cui contenuto non è fissato a catalogo: il cliente sceglie valori di attributi (colore, potenza, accessori, quantità di componenti) e un motore di regole lato backend verifica la consistenza, calcola i prezzi e segnala conflitti. La libreria `@spartacus/product-configurator` porta questo flusso nello storefront: bottone "Configure" su PDP/PLP, pagina di configurazione interattiva a gruppi/tab, overview riepilogativa, add-to-cart della configurazione, modifica dal carrello e visualizzazione read-only da ordini, quote e saved cart.

Il codice deve risolvere quattro problemi che lo distinguono da quasi tutte le altre feature lib.

1. **Più motori di configurazione dietro la stessa UI.** Il repo supporta il *Variant Configurator* (VC, tipo `CPQCONFIGURATOR`), SAP CPQ cloud (tipo `CLOUDCPQCONFIGURATOR`) e un configuratore minimale *textfield* (tipo `TEXTFIELD`) — `feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L60-L70`. VC e CPQ condividono store, facade e componenti ("rulebased"), e si distinguono solo nell'adapter scelto a runtime — `feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L160-L190`.
2. **Una configurazione appartiene a un "owner" che cambia nel tempo.** Prima dell'add-to-cart la configurazione è legata a un prodotto; dopo, a una entry di carrello; in un ordine, a una entry d'ordine. Il modello `CommonConfigurator.Owner` e la chiave `ownerType/ownerId` sono il filo conduttore di tutto — `feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L11-L29`, `feature-libs/product-configurator/common/shared/utils/configurator-model-utils.ts:L19-L27`.
3. **Lo stato vive nell'URL.** Tipo di configuratore, owner e modalità (configurazione/overview, display-only, resolve issues, expert mode…) sono codificati in route e query param e riletti da un unico servizio, `ConfiguratorRouterExtractorService` — `feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L28-L64`.
4. **Update incrementali e serializzati.** Ogni modifica di attributo invia al backend **un solo attributo cambiato**; gli update sono messi in coda con `concatMap` e il risultato finale è "consolidato" solo quando non ci sono più richieste pendenti — `feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L171-L207`, `:L403-L514`.

La libreria segue lo schema root (eager) / lazy delle feature lib (capitolo 12): un modulo root statico porta route, layout, nomi CMS che scatenano il lazy loading e un error handler HTTP; il resto (store NgRx, effects, adapter OCC, ~70 componenti) è caricato quando il CMS chiede uno dei componenti configuratore.

2. Mappa dei file

Tutti i path sono relativi a `feature-libs/product-configurator/` salvo dove indicato in forma completa.

PATH	RUOLO
`feature-libs/product-configurator/public_api.ts`	entry point principale vuoto (`export {}`)
`feature-libs/product-configurator/package.json`	versione e peerDependencies (cart, checkout, order, storefront, `@ng-select/ng-select`, `@ngrx/*`)

PATH	RUOLO
common/common-configurator.module.ts	CommonConfiguratorModule = OCC common + componenti carrello/PDP
common/core/model/common-configurator.model.ts	namespace CommonConfigurator (Owner, OwnerType), enum ConfiguratorType, ReadOnlyPostfix
common/core/model/augmented-core.model.ts	augmentation di Product e OrderEntry
common/core/model/product-configurator.config.ts	ProductConfiguratorConfig e augmentation di Config
common/shared/utils/configurator-model-utils.ts	ConfiguratorModelUtils (owner key, owner iniziale, createOwner)
common/shared/utils/common-configurator-utils.service.ts	CommonConfiguratorUtilsService (owner id composti, cart id, issue count)
common/components/service/configurator-router-data.ts	namespace ConfiguratorRouter (PageType, Data)
common/components/service/configurator-router-extractor.service.ts	ConfiguratorRouterExtractorService
common/occ/default-occ-configurator-product-config.ts	endpoint/scope prodotto configurator, configuratorProductCard
common/components/configure-product/*	bottone "Configure" (CMS + outlet ProductListOutlets.ITEM_ACTIONS)
common/components/configure-cart-entry/*	link edit/display/resolve issues su entry carrello/ordine/quote/saved cart
common/components/configurator-cart-entry-info/*	info configurazione in CartOutlets.ITEM_DETAILS
common/components/configurator-issues-notification/*	notifica issue in CartOutlets.ITEM_CONFIGURATOR_ISSUES
common/components/configurator-cart-entry-bundle-info/*	line item CPQ in CartOutlets.ITEM_BUNDLE_DETAILS
common/assets/translations/*	chunk i18n configurator (17 lingue)
rulebased/root/rulebased-configurator-root.module.ts	RulebasedConfiguratorRootModule (eager)
rulebased/root/rulebased-configurator-root-feature.module.ts	18 nomi CMS che triggerano la feature lazy
rulebased/root/default-rulebased-routing-config.ts	route VC
rulebased/root/variant/*	route + layout VC interattivo/overview, VariantConfiguratorPageLayoutHandler
rulebased/root/cpq/*	CpqConfiguratorRootModule, route/layout CPQ, CpqConfiguratorPageLayoutHandler
rulebased/root/http-interceptors/configurator-bad-request.handler.ts	ConfiguratorBadRequestHandler
rulebased/rulebased-configurator.module.ts	RulebasedConfiguratorModule (lazy)
rulebased/core/rulebased-configurator-core.module.ts	core: state, router listener, connector, listener logout/lingua
rulebased/core/connectors/*	RulebasedConfiguratorAdapter, RulebasedConfiguratorConnector

PATH	RUOLO
`rulebased/core/model/configurator.model.ts`	namespace `Configurator`
`rulebased/core/state/*`	feature `productConfigurator`: actions, reducer, effects (basic/cart/variant), selectors
`rulebased/core/facade/*`	`ConfiguratorCommonsService`, `ConfiguratorCartService`, `ConfiguratorGroupsService`, routing listener
`rulebased/core/events/*`	listener `LogoutEvent`/`LanguageSetEvent`
`rulebased/core/services/*`	`ConfiguratorExpertModeService`, `ConfiguratorQuantityService`
`rulebased/occ/variant/*`	`VariantConfiguratorOccModule`, adapter VC, converter, endpoint `ccpconfigurator`
`rulebased/cpq/*`	entry `rulebased/cpq`: `RulebasedCpqConfiguratorModule`, converter, `CpqConfiguratorOccAdapter`, `CpqConfiguratorOccService`
`rulebased/components/attribute/composition/*`	direttiva `cxConfiguratorAttributeComponent`, contesto, config `assignment`
`rulebased/components/attribute/types/*`	un componente per ogni `UiType` + classi base
`rulebased/components/{form,group,add-to-cart-button,overview-*,conflict-*,…}`	componenti di pagina
`textfield/*`	entry `textfield`: root, core (store a singola configurazione), occ, 4 componenti
`projects/storefrontapp/src/app/spartacus/features/product-configurator/*.ts`	wiring nella demo app

File aggiuntivi

PATH	RUOLO
`feature-libs/product-configurator/_index.scss`	entry SCSS: importa gli stili di `common`, `rulebased`, `textfield`
`feature-libs/product-configurator/vitest.config.ts`	configurazione Vitest: forza `NX_TSCONFIG_PATH` su `tsconfig.spec.json`, alias helper di test, esclude `textfield/root/**` dalla coverage

3. Come funziona passo per passo

3.1 Packaging e wiring: entry point, root eager, feature lazy

L'entry principale non esporta nulla — `feature-libs/product-configurator/public_api.ts:L16-L19`. Gli entry point secondari sono `common`, `common/assets`, `rulebased`, `rulebased/root`, `rulebased/cpq`, `textfield`, `textfield/root`, mappati in `tsconfig.json:L181-L200` e `tsconfig.json:L223-L224`. Il `package.json` dichiara `221121.22.0` — `feature-libs/product-configurator/package.json:L2-L3` — che è lo stesso valore di `@spartacus/core` e `@spartacus/cart` allo SHA (`core-libs/core/package.json:L3-L3`, `feature-libs/cart/package.json:L3-L3`).

Nella demo app il wiring rulebased è in due pezzi:

- **statico**: `ProductConfiguratorRulebasedFeatureModule` importa `RulebasedConfiguratorRootModule` e, solo se `environment.cpq`, `CpqConfiguratorRootModule`; registra la feature lazy `productConfiguratorRulebased` che

punta a `RulebasedConfiguratorWrapperModule` — projects/storefrontapp/src/app/spartacus/features/product-configurator/product-configurator-rulebased-feature.module.ts:L22-L40;

- **lazy**: il wrapper importa `RulebasedConfiguratorModule` e, con `environment.cpq`, `RulebasedCpqConfiguratorModule` — projects/storefrontapp/src/app/spartacus/features/product-configurator/rulebased-configurator-wrapper.module.ts:L12-L21.

Il textfield è analogo: statico `TextfieldConfiguratorRootModule`, lazy `TextfieldConfiguratorModule` sotto `PRODUCT_CONFIGURATOR_TEXTFIELD_FEATURE` — projects/storefrontapp/src/app/spartacus/features/product-configurator/product-configurator-textfield-feature.module.ts:L21-L33. L'installazione via schematics passa per `addProductConfiguratorFeatures` — feature-libs/product-configurator/schematics/add-product-configurator/index.ts:L25-L44 (capitolo 43).

Root rulebased. `RulebasedConfiguratorRootModule` importa `CommonConfiguratorModule`, il root-feature module, i moduli di routing VC interattivo/overview, `RulebasedConfiguratorRoutingModule.forRoot()` e registra `ConfiguratorBadRequestHandler` come `HttpErrorHandler` multi — feature-libs/product-configurator/rulebased/root/rulebased-configurator-root.module.ts:L21-L37. Il nome feature è `'productConfiguratorRulebased'` — feature-libs/product-configurator/rulebased/root/feature-name.ts:L7-L8 — e i 18 tipi CMS che ne scatenano il caricamento (da `ConfiguratorForm` a `ConfiguratorOverviewSidebar`) sono in feature-libs/product-configurator/rulebased/root/rulebased-configurator-root-feature.module.ts:L11-L55. Il meccanismo `featureModules` + `cmsComponents` è descritto nel capitolo 12.

Root CPQ. `CpqConfiguratorRootModule` aggiunge i moduli route/layout CPQ e fornisce `routing.protected: true` ("force early login") come **default** config, tramite `provideDefaultConfig` — feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-root.module.ts:L16-L21. Conseguenza: salvo override con `provideConfig` dell'app, importarlo rende protetto l'intero storefront, non solo le pagine CPQ, eccetto le route configurate con `protected: false` esplicito — core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L81 (vedi §9).

Common. `CommonConfiguratorModule = CommonConfiguratorOccModule + CommonConfiguratorComponentsModule` — feature-libs/product-configurator/common/common-configurator.module.ts:L11-L14. È importato dai root module, quindi è eager: aggiunge gli endpoint prodotto `configurator` (`fields=code,configurable,configuratorType`) e `configuratorProductCard`, e inserisce lo scope `configurator` nello scope `list` dei prodotti — feature-libs/product-configurator/common/occ/default-occ-configurator-product-config.ts:L10-L30. Augmenta inoltre `Product` (`configurable`, `configuratorType`) e `OrderEntry` (`statusSummaryList`, `configurationInfos`) — feature-libs/product-configurator/common/core/model/augmented-core.model.ts:L8-L19 — e `Config` con `productConfigurator?: ProductConfiguratorConfig` — feature-libs/product-configurator/common/core/model/product-configurator.config.ts:L9-L15.

Lazy rulebased. `RulebasedConfiguratorModule = VariantConfiguratorOccModule + RulebasedConfiguratorCoreModule + RulebasedConfiguratorComponentsModule` — feature-libs/product-configurator/rulebased/rulebased-configurator.module.ts:L12-L19. Il core fornisce esplicitamente `RulebasedConfiguratorConnector` ("otherwise lazy loading does not work"), la config core di default e istanzia nel costruttore i listener di logout e cambio lingua — feature-libs/product-configurator/rulebased/core/rulebased-configurator-core.module.ts:L16-L34. Il CPQ lazy è `RulebasedCpqConfiguratorModule = CpqConfiguratorCommonModule + CpqConfiguratorOccModule` — feature-libs/product-configurator/rulebased/cpq/rulebased-cpq-configurator.module.ts:L14-L17.

3.2 Owner e tipi: il modello comune

`CommonConfigurator.Owner` ha `type`, `key` ("used as key in the configuration store"), `id` (codice prodotto, numero di entry, oppure documento+entry) e `configuratorType` ("Derived from the cxRoute") — feature-libs/

product-configurator/common/core/model/common-configurator.model.ts:L11-L29. `OwnerType` ammette `product`, `cartEntry`, `orderEntry`, `quoteEntry`, `savedCartEntry` — feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L48-L54.

- La chiave è `ownerType + '/' + ownerId`, con errore se manca una delle parti — feature-libs/product-configurator/common/shared/utils/configurator-model-utils.ts:L19-L27; viene assegnata da `setOwnerKey` — feature-libs/product-configurator/common/shared/utils/common-configurator-utils.service.ts:L47-L49.
- L'owner "iniziale" ha `key`, `id` e `configuratorType` a `'INITIAL'` e `type = CommonConfigurator.OwnerType.PRODUCT` (`'product'`); `isInitialOwner` guarda solo `configuratorType` — feature-libs/product-configurator/common/shared/utils/configurator-model-utils.ts:L12-L48, in particolare :L33-L40.
- `createOwner` ha come default `ConfiguratorType.VARIANT` — feature-libs/product-configurator/common/shared/utils/configurator-model-utils.ts:L57-L68.
- Per entry di documenti (ordine, quote, saved cart) l'id è `documentId + '+' + entryNumber`; `decomposeOwnerId` lancia se le parti non sono esattamente 2 — feature-libs/product-configurator/common/shared/utils/common-configurator-utils.service.ts:L57-L75.
- `getCartId` usa il `guid` per l'utente anonimo, altrimenti `code` — feature-libs/product-configurator/common/shared/utils/common-configurator-utils.service.ts:L81-L88 (stessa regola del carrello, capitolo 25).
- VARIANT e TEXTFIELD sono "attribute based", CPQ è "bundle based" — feature-libs/product-configurator/common/shared/utils/common-configurator-utils.service.ts:L125-L146.

Il postfisso `_READ_ONLY` marca un tipo view-only (es. la route `configureCPQCONFIGURATOR_READ_ONLY`) — feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L60-L70, feature-libs/product-configurator/rulebased/root/default-rulebased-routing-config.ts:L9-L28.

3.3 Route e `ConfiguratorRouterExtractorService`

Le route semantiche seguono la convenzione `configure<TYPE>` / `configureOverview<TYPE>`:

CXROUTE	PATH	DEFINITO IN
`configureCPQCONFIGURATOR`	`configure/vc/:ownerType/entityKey/:entityKey`	feature-libs/product-configurator/rulebased/root/default-rulebased-routing-config.ts:L9-L28
`configureOverviewCPQCONFIGURATOR`	`configure-overview/vc/…/displayOnly/:displayOnly` e senza `displayOnly`	idem
`configureCLOUDCPQCONFIGURATOR`	`configure/cpq/:ownerType/entityKey/:entityKey`	feature-libs/product-configurator/rulebased/root/cpq/default-cpq-interactive-routing-config.ts:L9-L17
`configureOverviewCLOUDCPQCONFIGURATOR`	`configure-overview/cpq/…` (due path)	feature-libs/product-configurator/rulebased/root/cpq/default-cpq-overview-routing-config.ts:L9-L20
`configureTEXTFIELD` / `configureOverviewTEXTFIELD`	`configure/textfield/:ownerType/entityKey/:entityKey` …	feature-libs/product-configurator/textfield/root/default-textfield-routing-config.ts:L9-L23

Le route Angular sono registrate con un trucco: `path: null` (con `@ts-ignore`), `data.cxRoute`, `PageLayoutComponent` e `CmsPageGuard` — feature-libs/product-configurator/rulebased/root/variant/variant-configurator-interactive.module.ts:L20-L38. Il path reale viene poi iniettato dal meccanismo

`RoutingConfig` di Spartacus a partire da `cxRoute` (capitolo 14); la pagina è una normale pagina CMS (capitolo 15).

`extractRouterData()` trasforma lo stato del router NgRx in un `ConfiguratorRouter.Data` — feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L28-L64:

1. emette solo se esiste il parametro `entityKey`, e solo per stati senza `nextState` (navigazione completata) — :L28-L34;
2. i flag di query (`resolveIssues`, `skipConflicts`, `forceReload`, `expMode`, `displayRestartDialog`, `navigateToCheckout`, `isBundleOverview`) sono veri solo se uguali alla stringa `'true'` — :L38-L64;
3. `pageType` è `OVERVIEW` se il semanticRoute contiene `configureOverview`, altrimenti `CONFIGURATION` — :L38-L64, costanti in :L21-L22;
4. l'owner nasce da `ownerType`/`entityKey` (fallback: tipo `product` con `rootProduct`), il `configuratorType` è il suffisso del semanticRoute dopo `configureOverview` o `configure`, altrimenti `Error('Not able to determine configurator type')` — :L71-L113.

La forma completa di `ConfiguratorRouter.Data` è in feature-libs/product-configurator/common/components/service/configurator-router-data.ts:L8-L30. È questo il punto in cui **il tipo di configuratore entra nel sistema**: tutto ciò che segue (scelta dell'adapter, layout, comportamento CPQ vs VC) dipende da `owner.configuratorType` ricavato qui.

3.4 Gli ingressi: PDP/PLP e carrello

Bottone "Configure". `ConfigureProductModule` registra `ConfigureProductComponent` come componente CMS e lo inserisce anche nell'outlet `ProductListOutlets.ITEM_ACTIONS` in posizione AFTER — feature-libs/product-configurator/common/components/configure-product/configure-product.module.ts:L24-L42 (outlet: capitolo 17). Il componente legge il prodotto dal `ProductListItemContext` (PLP) oppure da `CurrentProductService` con scope DETAILS + `configurator` (PDP) — feature-libs/product-configurator/common/components/configure-product/configure-product.component.ts:L51-L62. È visibile solo se `product.configurable` e non è un base product read-only — feature-libs/product-configurator/common/components/configure-product/configure-product.component.html:L1-L2. Al click naviga a `cxRoute: 'configure' + product.configuratorType` con `ownerType: product`, `entityKey` = codice, e query `displayRestartDialog` e `productCode` — feature-libs/product-configurator/common/components/configure-product/configure-product.component.ts:L130-L148.

Componenti carrello. `CommonConfiguratorComponentsModule` importa i moduli carrello e fornisce la config UI comune — feature-libs/product-configurator/common/components/common-configurator-components.module.ts:L16-L26:

- `ConfiguratorCartEntryInfoComponent` in `CartOutlets.ITEM_DETAILS` (AFTER) — feature-libs/product-configurator/common/components/configurator-cart-entry-info/configurator-cart-entry-info.module.ts:L30-L36;
- `ConfiguratorIssuesNotificationComponent` **sostituisce** (REPLACE) `CartOutlets.ITEM_CONFIGURATOR_ISSUES` — feature-libs/product-configurator/common/components/configurator-issues-notification/configurator-issues-notification.module.ts:L28-L34; con il toggle `configuratorIssuesNotificationForConfigurableOnly` la notifica è nascosta per entry non configurabili — feature-libs/product-configurator/common/components/configurator-issues-notification/configurator-issues-notification.component.ts:L66-L76;
- `ConfiguratorCartEntryBundleInfoComponent` in `CartOutlets.ITEM_BUNDLE_DETAILS` (AFTER) — feature-libs/product-configurator/common/components/configurator-cart-entry-bundle-info/configurator-cart-entry-bundle-info.module.ts:L22-L28.

Il conteggio delle issue legge solo gli status summary `ERROR` — feature-libs/product-configurator/common/shared/utils/common-configurator-utils.service.ts:L106-L117. Le line item CPQ sono ricavate dai `configurationInfos` in formato legacy o versionato (`CI#@#VERSION` ≥ 2) — feature-libs/product-configurator/common/components/configurator-cart-entry-bundle-info/configurator-cart-entry-bundle-info.service.ts:L49-L76; i prodotti delle line item (per i link PDP) si caricano con lo scope `configuratorProductCard` solo dopo la prima espansione della lista — feature-libs/product-configurator/common/components/configurator-cart-entry-bundle-info/configurator-cart-entry-bundle-info.component.ts:L99-L143. Oltre `cartEntryBundleLineItemsThreshold` (default 10) e con toggle CPQ container, invece di espandere si naviga all'overview — feature-libs/product-configurator/common/components/configurator-cart-entry-bundle-info/configurator-cart-entry-bundle-info.component.ts:L225-L246, feature-libs/product-configurator/common/components/config/default-common-configurator-ui-settings.config.ts:L9-L14.

Link edit/display. `ConfigureCartEntryComponent` mappa il tipo di abstract order sull'owner type (ORDER→`orderEntry`, QUOTE→`quoteEntry`, SAVED_CART→`savedCartEntry`, default `cartEntry`) e, senza `AbstractOrderContext`, assume il carrello attivo — feature-libs/product-configurator/common/components/configure-cart-entry/configure-cart-entry.component.ts:L84-L88, :L145-L162. L'entityKey è il numero di entry per il carrello, altrimenti `documentId+entryNumber`; il route è `configureOverview<type>` se read-only e il tipo non è già `_READ_ONLY` — :L170-L193, :L331-L334. Le query params includono sempre `forceReload: true`, `resolveIssues`, `navigateToCheckout` (se si è in `checkoutReviewOrder`) e `productCode` — :L91-L103, :L336-L342.

3.5 Lo store NgRx rulebased

Feature key `productConfigurator`, entity type `[Configurator] Configuration Data`; lo stato è `configurations: EntityProcessesLoaderState<Configuration>` indicizzato per owner key — feature-libs/product-configurator/rulebased/core/state/configurator-state.ts:L10-L19. Lo state module registra `StoreModule.forFeature` (reducer via token) ed `EffectsModule.forFeature(ConfiguratorEffects)` con gli effect basic, cart e variant — feature-libs/product-configurator/rulebased/core/state/rulebased-configurator-state.module.ts:L19-L29, feature-libs/product-configurator/rulebased/core/state/effects/index.ts:L11-L15.

Il reducer è `StateUtils.entityProcessesLoaderReducer(CONFIGURATOR_DATA, configuratorReducer)` — feature-libs/product-configurator/rulebased/core/state/reducers/index.ts:L14-L24 — cioè un'entità con loader **e** contatore di processi (capitolo 19). Il reducer interno usa una `Map` lazy action-type → handler — feature-libs/product-configurator/rulebased/core/state/reducers/configurator.reducer.ts:L30-L45.

Punti chiave della semantica:

- `UpdateConfiguration` (e Add/Copy/RemoveContainerRow) sono `EntityProcessesIncrementAction` che impostano anche `meta.loader.load = true`; i Success/Fail sono `EntityProcessesDecrementAction` — feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L161-L175, :L178-L209.
- `hasPendingChanges = processesCount > 0` sull'entità — feature-libs/product-configurator/rulebased/core/state/selectors/configurator.selector.ts:L44-L50, core-libs/core/src/state/utils/processes-loader/processes-loader.selectors.ts:L13-L17.
- `AddToCart` e `UpdateCartEntry` del configuratore incrementano invece i processi dell'entità **carrello** (`MULTI_CART_DATA`, cartId), decrementati da `CartAddEntrySuccess/Fail` e `CartUpdateEntrySuccess/Fail` della cart lib — feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L91-L105, feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L38-L39. Così il carrello risulta "non stabile" durante l'add-to-cart di una configurazione.
- `takeOverChanges` fonde i gruppi nuovi con quelli già in store solo se il payload ne contiene, preserva `showConflictSolverDialog`, imposta `issueNavigationDone: true` e azzera i `priceSupplements` — feature-

libs/product-configurator/rulebased/core/state/reducers/configurator.reducer.ts:L436-L478.

- UPDATE_CONFIGURATION_FINALIZE_SUCCESS porta isCartEntryUpdateRequired = true, azzera overview e porta newConfiguration a false se definito — feature-libs/product-configurator/rulebased/core/state/reducers/configurator.reducer.ts:L122-L134; il dialog conflitti si apre se immediateConflictResolution && !consistent — :L136-L144.
- UPDATE_CART_ENTRY riporta isCartEntryUpdateRequired = false — :L175-L181.
- Il gruppo corrente iniziale è il primo di flatGroups, saltando i conflitti se immediateConflictResolution — :L406-L434.
- SET_NEXT_OWNER_CART_ENTRY imposta nextOwner = owner cartEntry con il numero di entry restituito dall'add-to-cart — :L290-L305.
- RemoveConfiguration è un EntityLoaderResetAction che accetta una chiave o un array — feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L375-L380.

Action e selector sono esportati come namespace ConfiguratorActions e ConfiguratorSelectors — feature-libs/product-configurator/rulebased/core/state/actions/index.ts:L7-L8, feature-libs/product-configurator/rulebased/core/state/selectors/index.ts:L7-L8.

3.6 Il modello Configurator

Configurator.Configuration contiene configId, groups e flatGroups, prezzi, overview, owner/nextOwner, isCartEntryUpdateRequired, interactionState, pricingEnabled, immediateConflictResolution, newConfiguration, isPricingAsync — feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L135-L171. Lo stato puramente UI (gruppo corrente, parent del menu, gruppi visitati, modalità risoluzione conflitti, dialog) sta in InteractionState — :L177-L187 — **dentro** l'entità, non in uno slice separato: per questo va copiato quando l'owner cambia (§3.10).

GroupType distingue gruppi di attributi, sub-item, header conflitti, conflitti e righe container — :L295-L302. UiType elenca i tipi di attributo, inclusi quelli introdotti con CPQ (checkBoxListProduct, dropdownProduct, radioGroupProduct, container) — :L317-L343. UpdateType è Attribute / AttributeQuantity / ValueQuantity — :L364-L368. Costanti: ConflictIdPrefix, ConflictHeaderId, ContainerRowGroupIdPrefix, CustomUiTypeIndicator = '___', RetractValueCode — :L386-L390.

3.7 Facade

ConfiguratorCommonsService.getOrCreateConfiguration smista per owner type — feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L98-L119:

- cartEntry → ConfiguratorCartService.readConfigurationForCartEntry;
- orderEntry/quoteEntry/savedCartEntry → readConfigurationForOrderEntry;
- default (prodotto) → getOrCreateConfigurationForProduct, che dispatcha CreateConfiguration come side-effect (tap) se la configurazione non è "creata" e non è in loading né in errore, ed emette solo configurazioni create — :L457-L496. "Creata" significa configId non vuoto e flatGroups non vuoti, oppure un overview — feature-libs/product-configurator/rulebased/core/facade/utils/configurator-utils.service.ts:L120-L129.

updateConfiguration non fa nulla (solo warning in dev mode) se il carrello attivo ha un code e non è stabile; altrimenti dispatcha UpdateConfiguration con un **estratto** della configurazione — feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L127-L172. L'estratto contiene solo il percorso di gruppi fino al gruppo dell'attributo e l'unico attributo cambiato, e richiede changedAttribute.groupId — feature-libs/product-configurator/rulebased/core/facade/utils/configurator-utils.service.ts:L139-L201.

Altri metodi notevoli: `getConfigurationWithOverview` (dispatcha `GetConfigurationOverview` se manca) — feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L326-L356; `forceNewConfiguration` (`RemoveConfiguration` + `CreateConfiguration` con `forceReset: true`) — :L442-L455; `hasConflicts` (vero solo con `immediateConflictResolution === false` e primo gruppo CONFLICT_HEADER) — :L426-L436; `readAttributeDomain` (`attributeKey = attribute.key ?? attribute.name`) — :L519-L538.

`ConfiguratorCartService`:

- `readConfigurationForCartEntry` ritarda la lettura finché il carrello è stabile e il checkout non sta caricando, "this can lead to cache issues with promotions" — feature-libs/product-configurator/rulebased/core/facade/configurator-cart.service.ts:L41-L99;
- `readConfigurationForOrderEntry` scompone l'owner id e dispatcha `ReadOrderEntryConfiguration` (anche per quote e saved cart) — :L108-L153;
- `addToCart` usa `requireLoadedCart`, l'userId e quantità di default 1 — :L163-L190;
- `updateCartEntry` usa `configuration.owner.id` come numero di entry — :L199-L221.

`ConfiguratorGroupsService.navigateToGroup` marca visitato il gruppo corrente e dispatcha `ChangeGroup` — feature-libs/product-configurator/rulebased/core/facade/configurator-groups.service.ts:L281-L311; `navigateToContainerRow` attende i gruppi, trova la riga per `rowId` e naviga — :L126-L161; il gruppo conflitto da mostrare nel dialog è il primo `CONFLICT_GROUP` quando `showConflictSolverDialog` — :L231-L246. Il pattern facade/servizio è quello generale del capitolo 20, con una differenza: qui le facade sono classi concrete del core, non facade astratte con `facadeFactory` nel root.

3.8 Connector multi-adapter: come si sceglie VC o CPQ

`RulebasedConfiguratorConnector` **non** è `providedIn: 'root'` ("this would break lazy loading") e riceve gli adapter dal token statico multi `CONFIGURATOR_ADAPTER_LIST` — feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L18-L30. Ogni metodo sceglie l'adapter il cui `getConfiguratorType()` coincide con `owner.configuratorType`; se nessuno combacia lancia `No adapter found for configurator type: …` — :L160-L190. VC dichiara `ConfiguratorType.VARIANT`, CPQ `ConfiguratorType.CPQ` — feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.adapter.ts:L50-L52, feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.adapter.ts:L26-L28.

Il token riceve `VariantConfiguratorOccAdapter` da `VariantConfiguratorOccModule` (sempre, nel lazy rulebased) — feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.module.ts:L32-L85 — e `CpqConfiguratorOccAdapter` da `CpqConfiguratorOccModule` (solo se è caricato il modulo CPQ) — feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.module.ts:L20-L41.

Il contratto astratto è `RulebasedConfiguratorAdapter` (create/read/update, overview get/update, addToCart, read/update cart entry, read order entry, readPriceSummary, getConfiguratorType, searchVariants) — feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.adapter.ts:L12-L161. I metodi container row sono **opzionali**: se l'adapter non li implementa il connector restituisce `throwError("… is not supported for configurator type …")` — feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.adapter.ts:L58-L80, feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L63-L70, :L173-L183. `updateConfigurationOverview` del connector chiama l'update se c'è un `overview`, altrimenti il get — :L138-L150.

3.9 Adapter OCC Variant Configurator

`VariantConfiguratorOccModule` registra l'adapter e 8 converter multi (normalizer, serializer, prezzi, add-to-cart, update cart entry, overview) — feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.module.ts:L32-L85. Gli endpoint sono sotto `ccpconfigurator` — feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L9-L50.

OPERAZIONE	HTTP	RIFERIMENTO
create	GET con `expMode`, `forceReset` (+ `configIdTemplate`)	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.adapter.ts:L67-L96
read	GET con `groupId`, `expMode`, `attributeKeyRequiresDomain`; forza `newConfiguration: false`	:L98-L127
update	PATCH del payload serializzato	:L129-L162
add to cart	POST, risposta via `CART_MODIFICATION_NORMALIZER`	:L164-L184
update cart entry	PUT	:L239-L241
read order/quote/saved cart entry	GET overview; configurazione con soli `overview` e gruppi vuoti	:L244-L305
price summary	GET `pricing` con `groupId = interactionState.currentGroup`	:L307-L331

Dopo la create, l'adapter attiva l'expert mode se `kbKey` è presente e riassegna l'owner — `:L67-L96`. Molte chiamate usano `HttpContext` con `OCC_HTTP_TOKEN.sendUserIdAsHeader: true` per l'emulazione ASM — `:L392-L406` (capitolo 29).

Il normalizer imposta `pricingEnabled ?? true`, `immediateConflictResolution ?? false`, `isPricingAsync: true` e mantiene `newConfiguration` tri-stato — feature-libs/product-configurator/rulebased/occ/variant/converters/occ-configurator-variant-normalizer.ts:L30-L47; il tipo UI è derivato troncando tutto ciò che segue `___` prima del mapping (UI type custom) — `:L440-L461`. Il serializer invia `configId`, `rootProduct`, `complete` e i gruppi — feature-libs/product-configurator/rulebased/occ/variant/converters/occ-configurator-variant-serializer.ts:L13-L32. Pattern adapter/converter: capitolo 21.

3.10 Adapter CPQ

Gli endpoint CPQ sono sotto `cpqconfigurator`, organizzati per tab, attributi, quantità valore, container rows e letture "Full" — feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L13-L45. L'adapter delega a `CpqConfiguratorOccService` e per l'update sceglie `updateValueQuantity` se `updateType === VALUE_QUANTITY`, altrimenti `updateAttribute` — feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.adapter.ts:L58-L71.

- Update: un PATCH per singolo attributo con `tabId` e, se presente, `rowId` — feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.service.ts:L315-L354. Il serializer prende il primo attributo seguendo il primo figlio lungo l'estratto — feature-libs/product-configurator/rulebased/cpq/rest/cpq-configurator-utils.ts:L67-L78; quantità valore di default 1 — feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator-value-serializer.ts:L30-L34.
- Container rows: POST `rows`, POST `copyRow`, DELETE — feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.service.ts:L356-L396. I tab annidati hanno groupId `CONTAINER_ROW@…@<tabId>` e solo il tabId finale va al backend — feature-libs/product-configurator/rulebased/cpq/rest/

cpq-configurator-utils.ts:L52-L56, feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator-normalizer.ts:L259-L261.

- Lettura entry documenti: ORDER_ENTRY, QUOTE_ENTRY, altrimenti saved cart — feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.service.ts:L414-L454.
- Non supportati: `readPriceSummary` restituisce la configurazione invariata ("so that UI does not run into exception"); `updateConfigurationOverview` e `searchVariants` lanciano errori **sincroni** — feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.adapter.ts:L106-L110, :L146-L160.

Il normalizer CPQ calcola `complete`/`consistent` dalle liste di messaggi, imposta `pricingEnabled: true` e trasforma ogni tab in un gruppo — feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator-normalizer.ts:L58-L95. Senza `hasFullConfigurationState`, solo il tab selezionato riceve gli attributi — :L1188-L1196; con il toggle `productConfiguratorCPQContainer` spento gli attributi container diventano `NOT_IMPLEMENTED` — :L670-L674, :L965-L975. L'add-to-cart serializza `{userId, cartId, product: {code}, quantity, configId}` — feature-libs/product-configurator/rulebased/cpq/occ/converters/occ-configurator-cpq-add-to-cart-serializer.ts:L24-L31.

3.11 Effects: create, update, cambio gruppo, errori

Create. `createConfiguration$` chiama il connector, dispatcha imperativamente `UpdatePriceSummary` (con il primo gruppo con attributi) ed emette `CreateConfigurationSuccess` + `SearchVariants` — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L62-L105. `getFirstGroupWithAttributes` lancia se nessun gruppo ha attributi — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic-effect.service.ts:L112-L125.

Update. `updateConfiguration$` usa `concatMap`: gli update sono processati uno dopo l'altro senza cancellare quelli in volo — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L171-L207. Le container row hanno effect dedicati, sempre con `concatMap` — :L209-L238. `updateConfigurationSuccess$` agisce solo quando non ci sono più processi pendenti, ed emette `UpdateConfigurationFinalizeSuccess`, `UpdatePriceSummary`, `SearchVariants` e, se il gruppo applicabile differisce da quello del payload, `ChangeGroup` — :L403-L514. Dopo `ADD_CONTAINER_ROW_SUCCESS` si naviga nel primo tab della nuova riga — :L447-L459. `updatePriceSummary$` lavora solo con `pricingEnabled === true` — :L302-L313.

Errori. Un update fallito, a processi esauriti, produce `UpdateConfigurationFinalizeFail`, che `handleErrorOnUpdate$` converte in `ReadConfiguration` del primo gruppo con attributi: lo stato viene riallineato rileggendo dal backend — :L516-L585.

Cambio gruppo. `groupChange$` (switchMap) attende l'assenza di processi pendenti, legge il gruppo ed emette `SetCurrentGroup`, `SetMenuParentGroup`, `ReadConfigurationSuccess`, `UpdatePriceSummary` — :L587-L654. Per CPQ un tab già caricato è servito dallo store senza chiamata backend — :L693-L721, feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic-effect.service.ts:L34-L61.

Varianti. La ricerca varianti parte solo con `enableVariantSearch === true` e tipo VARIANT; altrimenti un secondo effect emette `SearchVariantsSuccess` vuoto per resettare il loading — feature-libs/product-configurator/rulebased/core/state/effects/configurator-variant.effect.ts:L25-L82.

Pulizia. `removeProductBoundConfigurations$` rimuove le entità la cui chiave contiene `product` — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L656-L681; `removeCartBoundConfigurations$` quelle `cartEntry` e quelle `product` con `nextOwner` — feature-libs/product-configurator/rulebased/core/state/effects/configurator-cart.effect.ts:L210-L248.

3.12 Add to cart, cambio di owner e overview

`ConfiguratorAddToCartButtonComponent.onAddToCart` marca visitato il gruppo corrente, attende che non ci siano processi pendenti e poi aggiorna l'entry (owner `cartEntry`) o aggiunge al carrello (owner `product`) — feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L348-L393.

Per un owner prodotto:

1. disabilita il bottone e chiama `ConfiguratorCartService.addToCart` — :L395-L413;
2. `addToCart$` chiama l'adapter ed emette `AddNextOwner` (con l'entryNumber) e `CartAddEntrySuccess`; senza `entryNumber` lancia `Entry number is required in addToCart response`, catturato come `CartAddEntryFail` — feature-libs/product-configurator/rulebased/core/state/effects/configurator-cart.effect.ts:L34-L94;
3. `addOwner$` emette `SetNextOwnerCartEntry` e `SetInteractionState`, che copia l'`interactionState` della configurazione prodotto sull'entità con la nuova chiave `cartEntry/<n>` — :L250-L285;
4. il componente attende `nextOwner !== undefined` e naviga; sulla pagina di configurazione rimuove l'eventuale configurazione cartEntry preesistente con lo stesso numero, mentre la product-bound resta fino all'uscita dalle route `configure*` — feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L395-L461;
5. dalla pagina di configurazione la navigazione passa prima per `configure<type>` con owner `cartEntry` e `replaceUrl: true` (così il tasto back torna alla configurazione), poi a `configureOverview<type>` — :L172-L202, commento in :L177-L180; dall'overview si va direttamente al carrello — :L261-L281.

Per un owner cartEntry l'update del carrello avviene solo se `isCartEntryUpdateRequired` — :L463-L482. `updateCartEntry$` emette `CartUpdateEntrySuccess` e solo dopo `RemoveConfiguration`, perché il backend CPQ può aver scartato la bozza — feature-libs/product-configurator/rulebased/core/state/effects/configurator-cart.effect.ts:L96-L139. La lettura della configurazione della entry emette success + `UpdatePriceSummary` + `SearchVariants` — :L141-L180.

In display-only l'uscita dall'overview porta a dettaglio ordine, dettaglio quote, quote del saved cart, carrello, PDP o review checkout a seconda dell'owner — feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L484-L518.

3.13 Listener e servizi con stato

- `ConfiguratorRouterListener` (istanziato da `ConfiguratorRouterModule`) rimuove le configurazioni legate al carrello e riporta la quantità a 1 quando il semanticRoute non contiene `configure` — feature-libs/product-configurator/rulebased/core/facade/routing/configurator-router.listener.ts:L25-L38, feature-libs/product-configurator/rulebased/core/facade/routing/configurator-router.module.ts:L10-L15.
- Al `LogoutEvent` si azzerano expert mode e configurazioni product-bound — feature-libs/product-configurator/rulebased/core/events/configurator-logout-event.listener.ts:L27-L35; al `LanguageSetEvent` si rimuovono le product-bound — feature-libs/product-configurator/rulebased/core/events/configurator-language-set-event.listener.ts:L25-L31.
- `ConfiguratorExpertModeService` tiene due `ReplaySubject(1)` ed è documentato come OCC-specifico e sostituibile — feature-libs/product-configurator/rulebased/core/services/configurator-expert-mode.service.ts:L10-L22; `ConfiguratorQuantityService` è un `BehaviorSubject<number>(1)` — feature-libs/product-configurator/rulebased/core/services/configurator-quantity.service.ts:L16-L35.

3.14 La pagina: form, gruppo e composizione dinamica degli attributi

`ConfiguratorFormComponent` (CMS `ConfiguratorForm`) ottiene la configurazione con `getOrCreateConfiguration(owner, configIdTemplate)` solo sulle pagine CONFIGURATION — feature-libs/product-configurator/rulebased/components/form/configurator-form.component.ts:L58-L70. In `ngOnInit` controlla il dialog conflitti, apre il restart dialog se richiesto e la configurazione non è nuova, gestisce `resolveIssues`, `rowId` (con toggle CPQ container) ed `expMode` — :L113-L192.

`ConfiguratorGroupComponent` calcola per ogni attributo la chiave `AttributeType_` + `uiTypeVariation` (se contiene ___) oppure + `uiType` — feature-libs/product-configurator/rulebased/components/group/configurator-group.component.ts:L121-L127. Il template rende per ogni attributo tre slot con la direttiva `cxConfiguratorAttributeComponent`: `Header`, il tipo, `Footer` — feature-libs/product-configurator/rulebased/components/group/configurator-group.component.html:L25-L59.

La direttiva è un piccolo "CMS mapping" interno:

1. legge la mappa `productConfigurator.assignment` da `ConfiguratorAttributeCompositionConfig` (`useExisting: Config`) — feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.config.ts:L10-L24, feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.directive.ts:L37-L39;
2. in `ngOnChanges` ricrea il componente solo se l'attributo cambia (deep equal) o cambia il gruppo — :L59-L72;
3. crea il componente con `vcr.createComponent` e un injector figlio che fornisce `ConfiguratorAttributeCompositionContext` (useValue); se manca l'assignment, warning in dev mode — :L74-L101.

Il contesto contiene componentKey, attribute, owner, group, language, expMode, flag di navigazione e `isPricingAsync` — feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.model.ts:L11-L21. Ogni modulo di tipo registra il proprio componente con `provideDefaultConfig({productConfigurator:{assignment:{…}}})` — feature-libs/product-configurator/rulebased/components/attribute/types/radio-button/configurator-attribute-radio-button.module.ts:L33-L43; Header e Footer usano le chiavi `Header`/`Footer` — feature-libs/product-configurator/rulebased/components/attribute/header/configurator-attribute-header.module.ts:L35-L35, feature-libs/product-configurator/rulebased/components/attribute/footer/configurator-attribute-footer.module.ts:L27-L27.

Gerarchia delle classi base: `ConfiguratorAttributeBaseComponent` (senza decorator, usa `inject`) → `ConfiguratorAttributeSelectionBaseComponent` (`@Directive()` astratta, `loading$`) → single/multi selection base — feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-base.component.ts:L20-L37, feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-selection-base.component.ts:L23-L28. `loading$` torna a false quando il loading della configurazione finisce, perché con CPQ API V2 un round-trip può non cambiare l'attributo e il componente non verrebbe ricreato — :L42-L56. La single selection chiama `updateConfiguration` con `selectedSingleValue` e `UpdateType.ATTRIBUTE` — feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-single-selection-base.component.ts:L92-L130. Chi usa `ConfiguratorAttributePriceChangeService` deve fornirlo a livello di componente — feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-base.component.ts:L24-L34, feature-libs/product-configurator/rulebased/components/attribute/types/radio-button/configurator-attribute-radio-button.component.ts:L29-L33.

`RulebasedConfiguratorComponentsModule` aggrega 24 moduli (price summary, add-to-cart, menu, form, overview, conflitti, restart dialog…) — feature-libs/product-configurator/rulebased/components/rulebased-

configurator-components.module.ts:L32-L60.

3.15 Layout e page layout handler

Il layout interattivo VC (VariantConfigurationTemplate) ha header con VariantConfigExitButton, navigation VariantConfigMenu (vuota su lg) e slot di contenuto dedicati — feature-libs/product-configurator/rulebased/root/variant/variant-configurator-interactive-layout.module.ts:L16-L66. L'overview VC aggiunge la sezione non standard headerDisplayOnly e registra VariantConfiguratorPageLayoutHandler come PAGE_LAYOUT_HANDLER multi — feature-libs/product-configurator/rulebased/root/variant/variant-configurator-overview-layout.module.ts:L22-L66. L'handler, se routerData.displayOnly, sostituisce gli slot dell'header con quelli di headerDisplayOnly (lg o xs via BreakpointService.isUp) — feature-libs/product-configurator/rulebased/root/variant/variant-configurator-page-layout-handler.ts:L32-L71.

CPQ usa un unico CpqConfigurationTemplate per pagina interattiva e overview, con sezioni extra headerDisplayOnly e navigationDisplayOnly — feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-layout.module.ts:L18-L92; CpqConfiguratorPageLayoutHandler antepone PreHeader sulla pagina di configurazione, usa gli slot display-only sull'overview read-only e toglie CpqConfigOverviewNavigation col toggle CPQ container spento — feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-page-layout-handler.ts:L97-L160. PAGE_LAYOUT_HANDLER e LayoutConfig sono trattati nel capitolo 18.

3.16 Conflitti e dialog

LAUNCH_CALLER dello storefront è aumentato con CONFLICT_SOLVER, CONFIGURATOR_OV_FILTER, CONFIGURATOR_RESTART_DIALOG, assegnando i valori anche a runtime — feature-libs/product-configurator/rulebased/components/config/augmented-config.model.ts:L14-L33. Il conflict solver è un dialog inlineRoot — feature-libs/product-configurator/rulebased/components/conflict-solver-dialog/default-configurator-conflict-solver-layout.config.ts:L10-L18 — lanciato da ConfiguratorConflictSolverDialogLauncherService (istanziato dal costruttore del modulo, feature-libs/product-configurator/rulebased/components/conflict-solver-dialog/configurator-conflict-solver-dialog.module.ts:L35-L41), che apre il dialog quando c'è un gruppo conflitto, ignora riemissioni con stesso gruppo e stesso navigationId, e lo chiude con CLOSE_NO_CONFLICTS_EXIST — feature-libs/product-configurator/rulebased/components/conflict-solver-dialog/configurator-conflict-solver-dialog-launcher.service.ts:L66-L92.

3.17 Error handler HTTP

ConfiguratorBadRequestHandler (400, priorità NORMAL) intercetta solo IllegalStateError con messaggio "base product is defined as 'make-to-stock'" e mostra configurator.httpHandlers.makeToStockBaseProductIssue — feature-libs/product-configurator/rulebased/root/http-interceptors/configurator-bad-request.handler.ts:L22-L82. Con il toggle productConfiguratorCPQContainer fa match anche sui 400 UnknownIdentifierError delle richieste product card, ma in quel caso handleError non aggiunge messaggi: il 400 viene "inghiottito" — :L84-L125, :L42-L48. Il meccanismo HttpErrorHandler è nel capitolo 41.

3.18 Textfield: un configuratore "template"

Il textfield è descritto come configuratore minimale (3 attributi senza dipendenze) che "serves as a template for other configurator implementations" — feature-libs/product-configurator/textfield/textfield-configurator.module.ts:L12-L23. Differenze rispetto al rulebased:

- root con CMS `TextfieldConfigurationForm`, due route `path: null` e template a slot unico — feature-libs/product-configurator/textfield/root/textfield-configurator-root.module.ts:L19-L57, feature-libs/product-configurator/textfield/root/textfield-configurator-root-feature.module.ts:L11-L11; feature `productConfiguratorTextfield` — feature-libs/product-configurator/textfield/root/feature-name.ts:L7-L8;
- connector a singolo adapter (classe astratta `ConfiguratorTextfieldAdapter`), fornito esplicitamente dal core — feature-libs/product-configurator/textfield/core/textfield-configurator-core.module.ts:L15-L19, feature-libs/product-configurator/textfield/core/connectors/configurator-textfield.connector.ts:L14-L16;
- store `productConfigurationTextfield` con un singolo `LoaderState` (una configurazione alla volta) — feature-libs/product-configurator/textfield/core/state/configuration-textfield-state.ts:L10-L20, feature-libs/product-configurator/textfield/core/state/reducers/index.ts:L17-L25;
- update **solo locale**: nessun effect chiama il backend per `UPDATE_CONFIGURATION` — feature-libs/product-configurator/textfield/core/facade/configurator-textfield.service.ts:L81-L101, :L253-L271;
- dopo add-to-cart/update entry, `RemoveConfiguration` + `CartActions.LoadCart` (ricarica intera del carrello) — feature-libs/product-configurator/textfield/core/state/effects/configurator-textfield.effect.ts:L55-L84 (add-to-cart), :L86-L120 (update entry);
- endpoint `configurator/textfield`, update entry via POST — feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L13-L28, feature-libs/product-configurator/textfield/occ/occ-configurator-textfield.adapter.ts:L154-L156.

3.19 Stili e traduzioni

Le traduzioni sono nel chunk `configurator`, esportate come `configuratorTranslationsXx` — feature-libs/product-configurator/common/assets/translations/translations.ts:L9-L15 (capitolo 37). Ogni `_index.scss` applica `@extend %<selector>` su una lista `!default` di selettori e su `cx-page-layout.<Template>` per i template — feature-libs/product-configurator/rulebased/_index.scss:L38-L59 (capitolo 38).

---

4. Diagrammi

4.1 Moduli eager e lazy

■ **TAVOLA 30.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/product-configurator/rulebased/root/rulebased-configurator-root.module.ts:L21-L37, feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-root.module.ts:L16-L21, feature-libs/product-configurator/rulebased/rulebased-configurator.module.ts:L12-L19, feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.module.ts:L32-L85, feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.module.ts:L20-L41.

4.2 Dal PDP alla configurazione creata (VC)

■ **TAVOLA 30.2** diagramma a tutta pagina, in fondo al capitolo

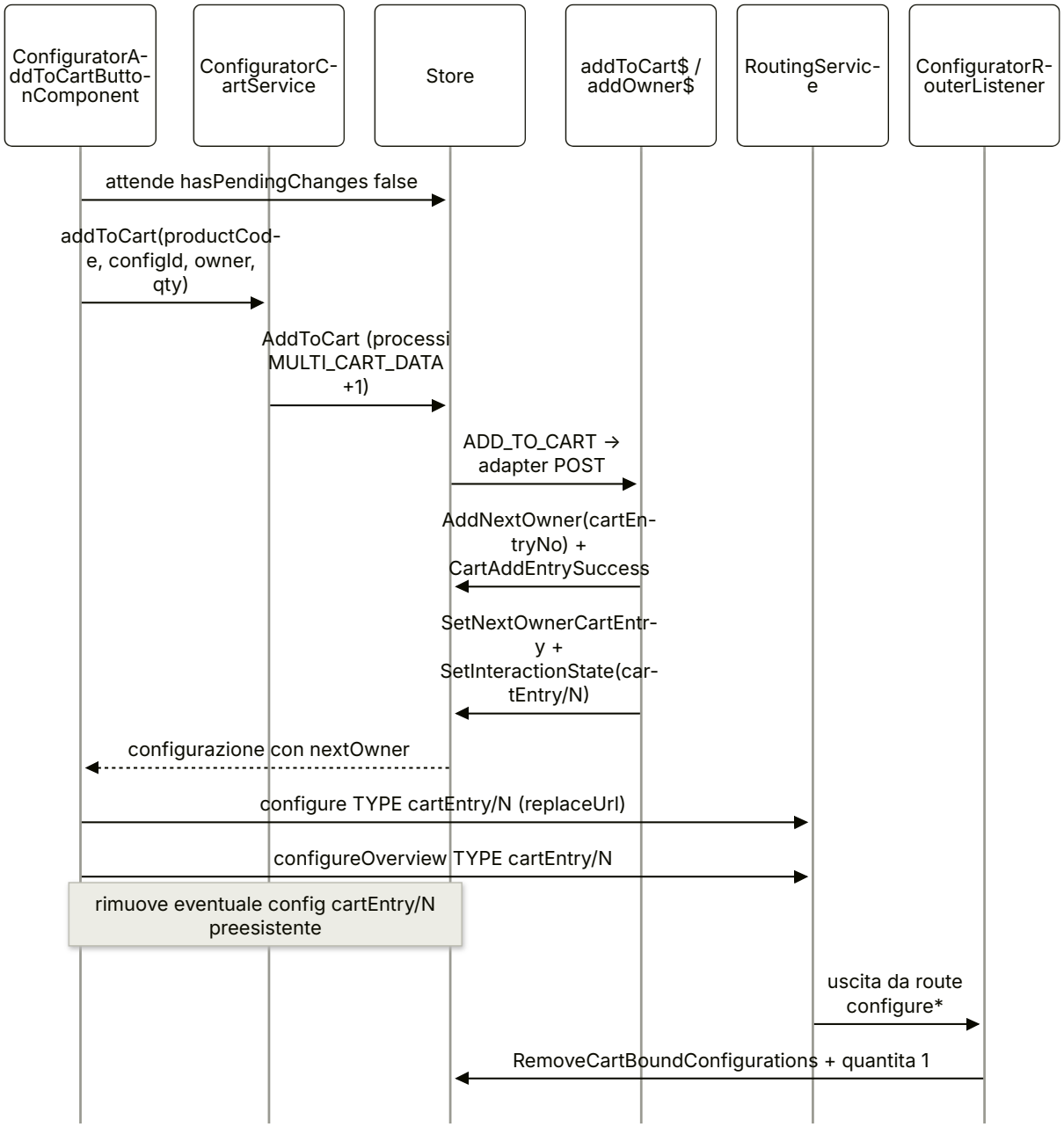
Riferimenti: feature-libs/product-configurator/common/components/configure-product/configure-product.component.ts:L130-L148, feature-libs/product-configurator/rulebased/root/variant/variant-configurator-interactive.module.ts:L20-L38, feature-libs/product-configurator/rulebased/components/form/configurator-form.component.ts:L58-L70, feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L457-L496, feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L62-L105, feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.adapter.ts:L67-L96.

4.3 Modifica di un attributo

■ **TAVOLA 30.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-single-selection-base.component.ts:L92-L130, feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L127-L172, feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L171-L207, :L403-L514, :L516-L585, feature-libs/product-configurator/rulebased/core/state/reducers/configurator.reducer.ts:L122-L144, feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-selection-base.component.ts:L42-L56.

4.4 Add to cart e cambio di owner



Riferimenti: `feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L348-L461`, `:L172-L202`, `feature-libs/product-configurator/rulebased/core/facade/configurator-cart.service.ts:L163-L190`, `feature-libs/product-configurator/rulebased/core/state/effects/configurator-cart.effect.ts:L34-L94`, `:L250-L285`, `feature-libs/product-configurator/rulebased/core/facade/routing/configurator-router.listener.ts:L25-L38`.

4.5 Risoluzione dell'owner e del tipo dall'URL

■ **TAVOLA 30.4** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L28-L34, :L71-L113, feature-libs/product-configurator/common/shared/utils/common-configurator-utils.service.ts:L47-L49, feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L160-L190.

---

5. Estratti di codice reale

5.1 Selezione dell'adapter

feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L18-L30

```ts
//Not provided in root, as this would break lazy loading
@Injectable()
export class RulebasedConfiguratorConnector {
  static CONFIGURATOR_ADAPTER_LIST = new InjectionToken<
    RulebasedConfiguratorAdapter[]
  >('ConfiguratorAdapterList');

  constructor(
    @Inject(RulebasedConfiguratorConnector.CONFIGURATOR_ADAPTER_LIST)
    protected adapters: RulebasedConfiguratorAdapter[],
    protected configUtilsService: CommonConfiguratorUtilsService,
    protected config: ConfiguratorCoreConfig
  ) {}
```

feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L160-L190

```ts
  protected getAdapter(configuratorType: string): RulebasedConfiguratorAdapter {
    const adapterResult = this.adapters.find((adapter) =>
      this.isAdapterMatching(adapter, configuratorType)
    );
    if (adapterResult) {
      return adapterResult;
    } else {
      throw new Error(
        'No adapter found for configurator type: ' + configuratorType
      );
    }
  }
  // ...
  protected isAdapterMatching(
    adapter: RulebasedConfiguratorAdapter,
    configuratorType: string
```

```
  ): boolean {
    return adapter.getConfiguratorType() === configuratorType;
  }
```

5.2 Registrazione dell'adapter CPQ nel token multi

feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.module.ts:L20-L41

```ts
@NgModule({
  imports: [CommonModule],
  providers: [
    provideDefaultConfigFactory(defaultOccCpqConfiguratorConfigFactory),
    // ...
    {
      provide: RulebasedConfiguratorConnector.CONFIGURATOR_ADAPTER_LIST,
      useClass: CpqConfiguratorOccAdapter,
      multi: true,
    },
  ],
})
export class CpqConfiguratorOccModule {}
```

5.3 Owner e tipo dal router

feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L71-L113

```ts
  createOwnerFromRouterState(
    routerState: RouterState
  ): CommonConfigurator.Owner {
    const owner: CommonConfigurator.Owner =
      ConfiguratorModelUtils.createInitialOwner();
    const params = routerState.state.params;
    if (params.ownerType) {
      const entityKey = params.entityKey;
      owner.type = params.ownerType;

      owner.id = entityKey;
    } else {
      owner.type = CommonConfigurator.OwnerType.PRODUCT;
      owner.id = params.rootProduct;
    }
    const semanticRoute = routerState.state.semanticRoute;
    if (semanticRoute) {
      const configuratorType =
        this.getConfiguratorTypeFromSemanticRoute(semanticRoute);
      owner.configuratorType = configuratorType;
```

```
  }
  this.configUtilsService.setOwnerKey(owner);
  return owner;
}
// ...
protected getConfiguratorTypeFromSemanticRoute(
  semanticRoute: string
): string {
  if (semanticRoute.startsWith(this.ROUTE_FRAGMENT_OVERVIEW)) {
    return semanticRoute.split(this.ROUTE_FRAGMENT_OVERVIEW)[1];
  } else if (semanticRoute.startsWith(this.ROUTE_FRAGMENT_CONFIGURE)) {
    return semanticRoute.split(this.ROUTE_FRAGMENT_CONFIGURE)[1];
  } else {
    throw new Error('Not able to determine configurator type');
  }
}
```

5.4 Create "lazy" dalla facade

feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L457-L496

```ts
protected getOrCreateConfigurationForProduct(
  owner: CommonConfigurator.Owner,
  configIdTemplate?: string
): Observable<Configurator.Configuration> {
  return this.store.pipe(
    select(
      ConfiguratorSelectors.getConfigurationProcessLoaderStateFactory(
        owner.key
      )
    ),
    tap((configurationState) => {
      if (
        (configurationState.value === undefined ||
          !this.configuratorUtils.isConfigurationCreated(
            configurationState.value
          )) &&
        configurationState.loading !== true &&
        configurationState.error !== true
      ) {
        this.store.dispatch(
          new ConfiguratorActions.CreateConfiguration({
            owner,
            configIdTemplate,
          })
        );
      }
    }),
```

```
    filter(
      (configurationState) =>
        configurationState.value !== undefined &&
        this.configuratorUtils.isConfigurationCreated(
          configurationState.value
        )
    ),
    // ...
```

5.5 Update serializzati con `concatMap`

feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L171-L207

```ts
updateConfiguration$: Observable<
  | ConfiguratorActions.UpdateConfigurationSuccess
  | ConfiguratorActions.UpdateConfigurationFail
> = createEffect(() =>
  this.actions$.pipe(
    ofType(ConfiguratorActions.UPDATE_CONFIGURATION),
    map((action: ConfiguratorActions.UpdateConfiguration) => action.payload),
    //concatMap (not mergeMap) so that updates are processed strictly one after
    //another: concatMap queues the updates without cancelling in-flight
    //emissions (switchMap would cancel and lose changes; each update only sends
    //one changed attribute at a time).
    concatMap((payload: Configurator.Configuration) => {
      return this.configuratorCommonsConnector
        .updateConfiguration(payload)
        .pipe(
          map((configuration: Configurator.Configuration) => {
            return new ConfiguratorActions.UpdateConfigurationSuccess({
              ...configuration,
              interactionState: {
                isConflictResolutionMode:
                  payload.interactionState.isConflictResolutionMode,
              },
            });
          }),
          catchError((error) => {
            const errorPayload = tryNormalizeHttpError(error, this.logger);
            return [
              new ConfiguratorActions.UpdateConfigurationFail({
                configuration: payload,
                error: errorPayload,
              }),
            ];
          })
        );
```

```
      })
    )
  );
```

5.6 Reducer di finalize

feature-libs/product-configurator/rulebased/core/state/reducers/configurator.reducer.ts:L122-L144

```ts
function handleActionUpdateConfigurationFinalizeSuccess(
  state: Configurator.Configuration,
  action: ConfiguratorActions.UpdateConfigurationFinalizeSuccess
): Configurator.Configuration | undefined {
  const result: Configurator.Configuration = takeOverChanges(action, state);
  checkConflictSolverDialog(result);
  result.isCartEntryUpdateRequired = true;
  result.overview = undefined;
  if (state.interactionState.newConfiguration !== undefined) {
    result.interactionState.newConfiguration = false;
  }
  return result;
}

function checkConflictSolverDialog(
  configuration: Configurator.Configuration
): void {
  configuration.interactionState.showConflictSolverDialog =
    configuration.immediateConflictResolution && !configuration.consistent;
  if (configuration.interactionState.showConflictSolverDialog) {
    configuration.interactionState.issueNavigationDone = true;
  }
}
```

5.7 Update cart entry e rimozione della bozza

feature-libs/product-configurator/rulebased/core/state/effects/configurator-cart.effect.ts:L109-L124

```ts
              switchMap((cartModification: CartModification) => {
                return [
                  new CartActions.CartUpdateEntrySuccess({
                    userId: payload.userId,
                    cartId: payload.cartId,
                    entryNumber: payload.cartEntryNumber,
                    quantity: cartModification.quantity,
                  }),
                  // The backend might have discarded the draft configuration (CPQ), so it needs
                  // to be re-read from the cart entry. Must not happen before the update has
```

```
                // succeeded, as pending configuration success actions would re-create it.
                new ConfiguratorActions.RemoveConfiguration({
                  ownerKey: payload.configuration.owner.key,
                }),
              ];
            }),
```

5.8 Composizione dinamica degli attributi

feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.directive.ts:L59-L101

```ts
  ngOnChanges(): void {
    const attributeChanged = !ObjectComparisonUtils.deepEqualObjects(
      this.lastRenderedAttribute,
      this.context.attribute
    );
    const groupChanged = this.lastRenderedGroupId !== this.context.group.id;
    // attribute can occur with same content twice in different groups
    // for example this happens for conflicts. An attribute is rendered differently (link from/to
conflict) based on
    // if it is part of conflict group or of ordinary group
    if (attributeChanged || groupChanged) {
      const key = this.context.componentKey;
      this.renderComponent(this.attrComponentAssignment[key], key);
    }
  }

  protected renderComponent(component: any, componentKey: string) {
    if (component) {
      this.lastRenderedAttribute = this.context.attribute;
      this.lastRenderedGroupId = this.context.group.id;
      this.vcr.clear();
      this.vcr.createComponent(component, {
        injector: this.getComponentInjector(),
      });
    } else {
      if (isDevMode()) {
        this.logger.warn(
          'No attribute type component available for: ' + componentKey
        );
      }
    }
  }
  // ...
```

feature-libs/product-configurator/rulebased/components/attribute/types/radio-button/configurator-attribute-radio-button.module.ts:L33-L43

```ts
providers: [
  provideDefaultConfig({
    productConfigurator: {
      assignment: {
        AttributeType_radioGroup: ConfiguratorAttributeRadioButtonComponent,
        AttributeType_radioGroup_add:
          ConfiguratorAttributeRadioButtonComponent,
      },
    },
  }),
],
```

5.9 Navigazione verso l'overview con history riscritta

feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L172-L202

```ts
protected navigateToOverview(
  configuratorType: string,
  owner: CommonConfigurator.Owner,
  productCode?: string
): void {
  // When navigating to the overview page, we change the browser history,
  // by changing the owner type and entity key (cart entry) in the configure URL.
  // With that the user can return to the configuration page
  // by using the browser back button.
  this.routingService
    .go(
      {
        cxRoute: 'configure' + configuratorType,
        params: { ownerType: 'cartEntry', entityKey: owner.id },
      },
      { replaceUrl: true }
    )
    .then(() => {
      this.routingService
        .go(
          {
            cxRoute: 'configureOverview' + configuratorType,
            params: { ownerType: 'cartEntry', entityKey: owner.id },
          },
          { queryParams: { productCode: productCode } }
        )
        // ...
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
productConfigurator.enableVariantSearch	boolean	false	feature-libs/product-configurator/rulebased/core/config/default-configurator-core.config.ts:L9-L13	feature-libs/product-configurator/rulebased/core/state/effects/configurator-variant.effect.ts:L31-L35, :L68-L73
productConfigurator.cartEntryBundleLineItemsThreshold	number	10	feature-libs/product-configurator/common/components/config/default-common-configurator-ui-settings.config.ts:L9-L14	feature-libs/product-configurator/common/components/configurator-cart-entry-bundle-info/configurator-cart-entry-bundle-info.component.ts:L268-L272
productConfigurator.updateDebounceTime.{quantity,input,date}	number	750 / 500 / 1500	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L13-L17	feature-libs/product-configurator/rulebased/components/attribute/quantity/configurator-attribute-quantity.component.ts:L79-L79 (quantity), feature-libs/product-configurator/rulebased/components/attribute/types/input-field/configurator-attribute-input-field.component.ts:L190-L198 (date/input), feature-libs/product-configurator/rulebased/components/attribute/types/numeric-input-field/configurator-attribute-numeric-input-field.component.ts:L135-L135 (input)
productConfigurator.addRetractOption	boolean	false	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L18-L18	normalizer VC: feature-libs/product-configurator/rulebased/occ/variant/converters/occ-configurator-variant-normalizer.ts:L264-L264
productConfigurator.descriptions.*	number	100 / 70	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L19-L22	feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-base.component.ts:L487-L487
productConfigurator.cpqContainerDropDownListThreshold	number	10	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L23-L23	componente container
productConfigurator.cpqHideBasePriceAndSelectedOptions	boolean	false	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L24-L24	feature-libs/product-configurator/rulebased/components/service/configurator-storefront-utils.service.ts:L731-L731
productConfigurator.overviewHeaderSelectors	string[]	.VariantConfigOverviewHeader, .CpqConfigHeader	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L25-L28	overview menu

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
productConfigurator.overviewMenuFilterTabVisible	Record<string, boolean>	{CPQCONFIGURATOR: true}	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L29-L31	feature-libs/product-configurator/rulebased/components/service/configurator-storefront-utils.service.ts:L713-L713
productConfigurator.updateConfigurationMessage.waitingTime	number	1000	feature-libs/product-configurator/rulebased/components/config/default-configurator-message.config.ts:L9-L15	feature-libs/product-configurator/rulebased/components/update-message/configurator-update-message.component.ts:L36-L37
productConfigurator.assignment	{[key]: component}	registrato dai moduli tipo	feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.config.ts:L10-L24	feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.directive.ts:L37-L39
routing.routes.configure* / configureOverview*	RoutingConfig	vedi §3.3	feature-libs/product-configurator/rulebased/root/default-rulebased-routing-config.ts:L9-L28 (e CPQ/textfield)	ConfiguratorRouterExtractorService via semanticRoute
routing.protected	boolean	default true solo con CPQ root (sovrascrivibile da provideConfig)	feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-root.module.ts:L19-L19	routing core (capitolo 22)
backend.occ.endpoints.*	string	ccpconfigurator/…, cpqconfigurator/…, configurator/textfield, products/${productCode}?fields=…	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L9-L50, feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L13-L45, feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L13-L28, feature-libs/product-configurator/common/occ/default-occ-configurator-product-config.ts:L10-L30	adapter via OccEndpointsService.buildUrl
backend.loadingScopes.product.list.include	string[]	['configurator']	feature-libs/product-configurator/common/occ/default-occ-configurator-product-config.ts:L22-L28	product loading core (capitolo 24)
layoutSlots.{Variant,Cpq,Textfield}*Template	LayoutConfig	vedi §3.15	moduli layout root	PageLayoutService + handler
launch.CONFLICT_SOLVER (e OV filter, restart)	LayoutConfig	dialog inline	feature-libs/product-configurator/rulebased/components/conflict-solver-dialog/default-configurator-conflict-solver-layout.config.ts:L10-L18	LaunchDialogService
toggle productConfiguratorCPQContainer	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L788-L788, :L1098-L1098	handler 400, layout handler CPQ, bundle info, normalizer CPQ, form

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `configuratorIssuesNotificationForConfigurableOnly`	boolean	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L862-L862`, `:L1107-L1107`	`feature-libs/product-configurator/common/components/configurator-issues-notification/configurator-issues-notification.component.ts:L66-L76`
toggle `productConfiguratorConsolidatedButtonDisabling`	boolean	`false` (`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1097-L1097`)	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L721-L721`	`feature-libs/product-configurator/rulebased/components/attribute/types/multi-selection-bundle/configurator-attribute-multi-selection-bundle.component.html:L19-L19`

Le abstract class di config sono `ConfiguratorCoreConfig` (`feature-libs/product-configurator/rulebased/core/config/configurator-core.config.ts:L10-L20`), `ConfiguratorUISettingsConfig` (`feature-libs/product-configurator/rulebased/components/config/configurator-ui-settings.config.ts:L10-L57`), e `CommonConfiguratorUISettingsConfig`: queste sono `providedIn: 'root'` con `useExisting: Config` (capitolo 11). `ConfiguratorMessageConfig` invece è una abstract class senza decorator — `feature-libs/product-configurator/rulebased/components/config/configurator-message.config.ts:L13-L15` — fornita solo da `ConfiguratorUpdateMessageModule` con un provider `{ provide: ConfiguratorMessageConfig, useExisting: Config }` — `feature-libs/product-configurator/rulebased/components/update-message/configurator-update-message.module.ts:L30-L31`; iniettarla fuori dall'injector che importa quel modulo non trova provider.

---

7. Commenti e note notevoli dal sorgente

- "Not provided in root, as this would break lazy loading" — `feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L18-L18`.
- "Explicit providing of connector because otherwise lazy loading does not work" — `feature-libs/product-configurator/rulebased/core/rulebased-configurator-core.module.ts:L16-L19`; idem nel text-field — `feature-libs/product-configurator/textfield/core/textfield-configurator-core.module.ts:L11-L14`.
- "// TODO: Inline this factory when we start releasing Ivy compiled libraries" — `feature-libs/product-configurator/rulebased/root/rulebased-configurator-root-feature.module.ts:L32-L32`, `feature-libs/product-configurator/textfield/root/textfield-configurator-root-feature.module.ts:L13-L13`.
- "We can neither omit the path nor set to undefined" + `@ts-ignore` su `path: null` — `feature-libs/product-configurator/rulebased/root/variant/variant-configurator-interactive.module.ts:L24-L26`.
- "//force early login" — `feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-root.module.ts:L18-L18`.
- `@ts-ignore TODO (#12620)` sul reducer entity — `feature-libs/product-configurator/rulebased/core/state/reducers/index.ts:L16-L20`.
- "TODO: remove the logic below when configurable products support "Saved Cart" and "Save For Later"" — `feature-libs/product-configurator/common/components/configurator-cart-entry-info/configurator-cart-entry-info.component.ts:L41-L41`, `feature-libs/product-configurator/common/components/configurator-issues-notification/configurator-issues-notification.component.ts:L56-L56`.
- "Requires default change detection strategy, as the disabled state of the quantity form control may change" — `feature-libs/product-configurator/common/components/configurator-cart-entry-info/configurator-cart-entry-info.component.ts:L16-L25`.

- Il commento su `concatMap` vs `switchMap` degli update — `feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L178-L181`.
- "we need a trinary state true, false, undefined!" — `feature-libs/product-configurator/rulebased/occ/variant/converters/occ-configurator-variant-normalizer.ts:L45-L45`.
- Il commento sul cleanup posticipato della configurazione product-bound ("existing observables would instantly trigger a re-create") — `feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L448-L458`.
- "We don't need this method but cannot remove it due to backward compatibility" (`ngOnInit` della direttiva) — `feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.directive.ts:L46-L52`.
- JSDoc di `ConfiguratorExpertModeService`: "This implementation is OCC specific" — `feature-libs/product-configurator/rulebased/core/services/configurator-expert-mode.service.ts:L10-L14`.
- JSDoc del toggle `productConfiguratorCPQContainer` con l'elenco delle classi coinvolte — `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L760-L788`.
- JSDoc di `getCpqContainerFragment` sulle assunzioni `anchorScrolling: 'enabled'` e assenza di id uguali al rowId — `feature-libs/product-configurator/common/components/configure-cart-entry/configure-cart-entry.component.ts:L195-L216`.
- Il JSDoc del normalizer textfield dice "Converts addToCart parameters into the generic format" ma converte una configurazione — `feature-libs/product-configurator/textfield/occ/converters/occ-configurator-textfield-normalizer.ts:L23-L28`.

8. Extension point

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`RulebasedConfiguratorConnector.CONFIGURATOR_ADAPTER_LIST`	aggiungere un adapter `multi: true` per un nuovo `configuratorType` (o sostituirne uno)	`feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L21-L23`	52 — OCC, adapter, converter
`RulebasedConfiguratorAdapter`	classe astratta da implementare; container row opzionali	`feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.adapter.ts:L12-L161`	52
Token converter VC (`VARIANT_CONFIGURATOR_NORMALIZER`, `VARIANT_CONFIGURATOR_SERIALIZER`, …) e CPQ (`CPQ_CONFIGURATOR_NORMALIZER`, …)	aggiungere normalizer/serializer `multi: true` per campi custom	`feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L12-L49`, `feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator.converters.ts:L12-L26`	52
Endpoint `backend.occ.endpoints.*`	override di URL/fields via `provideConfig`	§6	52
`productConfigurator.assignment`	mappare un `AttributeType_<uiType>` (anche custom con `___`) su un componente proprio	`feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.config.ts:L10-L24`	51 — componenti CMS, outlet, layout

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
`ConfiguratorAttributeCompositionContext` + classi base	il componente custom inietta il contesto ed estende `ConfiguratorAttributeSingleSelectionBaseComponent` ecc.	`feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.model.ts:L11-L21`, `feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-selection-base.component.ts:L23-L28`	51
`cmsComponents` (18 nomi rulebased, `TextfieldConfigurationForm`, `ConfigureProductComponent`)	sostituire un componente di pagina	`feature-libs/product-configurator/rulebased/root/rulebased-configurator-root-feature.module.ts:L11-L55`	51
Outlet `ProductListOutlets.ITEM_ACTIONS`, `CartOutlets.ITEM_DETAILS`, `ITEM_CONFIGURATOR_ISSUES`, `ITEM_BUNDLE_DETAILS`	aggiungere/sostituire contenuti nel carrello e nella PLP	§3.4	51
`layoutSlots` + `PAGE_LAYOUT_HANDLER`	cambiare slot dei template `VariantConfigurationTemplate`/`CpqConfigurationTemplate` o aggiungere un handler	`feature-libs/product-configurator/rulebased/root/variant/variant-configurator-overview-layout.module.ts:L22-L66`	51
`ConfiguratorUISettingsConfig`, `ConfiguratorCoreConfig`, `CommonConfiguratorUISettingsConfig`	override di debounce, soglie, variant search	§6	50 — config e DI
Facade e servizi (`ConfiguratorCommonsService`, `ConfiguratorCartService`, `ConfiguratorGroupsService`, `ConfiguratorExpertModeService`, `ConfiguratorQuantityService`)	sostituzione via DI (`useClass`/`useExisting`)	`feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L98-L119`, `feature-libs/product-configurator/rulebased/core/services/configurator-expert-mode.service.ts:L10-L22`	50
`ConfiguratorBasicEffectService`	helper "to enhance effects without new effects"	`feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic-effect.service.ts:L12-L17`	50
`HttpErrorHandler`	affiancare/sostituire `ConfiguratorBadRequestHandler`	`feature-libs/product-configurator/rulebased/root/http-interceptors/configurator-bad-request.handler.ts:L22-L82`	50
Augmentation `Configurator`, `ProductConfiguratorConfig`, `LAUNCH_CALLER`	estendere modello e config in modo type-safe	`feature-libs/product-configurator/common/core/model/product-configurator.config.ts:L9-L15`, `feature-libs/product-configurator/rulebased/components/config/augmented-config.model.ts:L14-L33`	54 — upgrade-safe

EXTENSION POINT	COME SI USA	RIFERIMENTO	RICETTA
Stili (`$configurator-rulebased-pages` e liste componenti `!default`) e chunk i18n `configurator`	escludere/riscrivere stili, tradurre	`feature-libs/product-configurator/rulebased/_index.scss:L38-L59`, `feature-libs/product-configurator/common/assets/translations/translations.ts:L9-L15`	53 — feature, cart, stili, i18n

Limiti dell'estendibilità:

- Il textfield **non** esporta adapter astratto, connector, facade, state né i token converter: `feature-libs/product-configurator/textfield/core/index.ts:L7-L7` esporta solo il core module (il public API, `feature-libs/product-configurator/textfield/public_api.ts:L7-L10`, espone componenti, core, occ e `TextfieldConfiguratorModule`). Esporta però l'adapter OCC concreto `OccConfiguratorTextfieldAdapter` e le classi converter OCC (normalizer e add-to-cart serializer) — `feature-libs/product-configurator/textfield/occ/index.ts:L7-L9`, `feature-libs/product-configurator/textfield/occ/converters/index.ts:L7-L8`. Usarlo come "template" significa copiarlo, non estenderlo.
- `CpqConfiguratorPageLayoutHandler` non è esportato: `feature-libs/product-configurator/rulebased/root/cpq/index.ts:L7-L10` esporta solo i quattro moduli CPQ, mentre `VariantConfiguratorPageLayoutHandler` è pubblico — `feature-libs/product-configurator/rulebased/root/variant/index.ts:L11-L11`. Per cambiare il comportamento CPQ si registra un proprio `PAGE_LAYOUT_HANDLER` invece di estendere la classe.
- Un nuovo configuratore rulebased richiede in pratica: un `configuratorType` nuovo, route `configure<TYPE>`/`configureOverview<TYPE>` con route Angular `path: null` + `cxRoute`, un adapter nel token multi, un layout per il template CMS. Tutti questi pezzi esistono come pattern nel codice VC/CPQ (§3.1–§3.10); una guida passo-passo non esiste nel repo.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `CpqConfiguratorRootModule` **protegge tutto lo storefront.** Fornisce `routing.protected: true` come **default** config (sovrascrivibile da `provideConfig` dell'app) — `feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-root.module.ts:L19-L19` — quindi l'intero storefront diventa protetto, non solo le pagine CPQ, salvo le route configurate con `protected: false` esplicito — `core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L72-L81`.
2. **Update scartati in silenzio.** Se il carrello attivo ha un `code` e non è stabile, `updateConfiguration` (e le container row) non dispatchano nulla; c'è solo un warning in dev mode — `feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L139-L151`. Poiché l'add-to-cart del configuratore incrementa i processi del carrello (`feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L91-L105`), un click su un attributo durante l'add-to-cart viene perso.
3. `loading` **resta true fino al finalize.** `UpdateConfigurationSuccess` non ha meta loader: il flag impostato da `setPendingLoader` resta true fino a `UpdateConfigurationFinalizeSuccess`, emessa solo a processi esauriti — `feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L203-L209`, `core-libs/core/src/state/utils/loader/loader.reducer.ts:L39-L60`.
4. **CPQ: errori sincroni negli adapter.** `updateConfigurationOverview` e `searchVariants` del CPQ lanciano in modo sincrono; in `updateOverview$` l'eccezione nasce nella project di `mergeMap`, fuori dal `catchError` interno, quindi l'effect va in errore e viene risottoscritto dal `defaultEffectsErrorHandler` di NgRx (max 10 volte) — `feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.adapter.`

ts:L106-L110, feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L377-L380, deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L174-L185. `searchVariants` è invece protetto dal filtro tipo VARIANT — feature-libs/product-configurator/rulebased/core/state/effects/configurator-variant.effect.ts:L25-L82.

5. `enableVariantSearch` **undefined.** Se si sovrascrive `productConfigurator` senza la chiave e il tipo è VARIANT, nessuno dei due effect variant emette (uno vuole `=== true`, l'altro gestisce il caso "non attivo"), e `SearchVariants` resta senza success — feature-libs/product-configurator/rulebased/core/state/effects/configurator-variant.effect.ts:L31-L35, :L68-L73.
6. **Placeholder letterale nell'URL CPQ.** `readCpqConfiguration` contiene `?tabId=${tabId}` nel template, ma `tabId` non è passato come urlParam: se è falsy il placeholder resta nell'URL — feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L25-L26, feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.service.ts:L292-L297, core-libs/core/src/occ/services/occ-endpoints.service.ts:L121-L146.
7. **Match per substring delle chiavi.** La pulizia riconosce le entità con `ownerKey.includes('product')` / `'cartEntry'`: un id contenente "product" farebbe match — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L668-L670, feature-libs/product-configurator/rulebased/core/state/effects/configurator-cart.effect.ts:L223-L232.
8. `displayOnly` **è una stringa.** L'extractor copia `params.displayOnly` (stringa) in un campo tipizzato boolean: `'false'` sarebbe truthy — feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L42-L42, feature-libs/product-configurator/common/components/service/configurator-router-data.ts:L17-L17. I link generati dal codice usano `displayOnly` solo con valore true.
9. **Due mapping per** `ConfiguratorForm`**.** `ConfiguratorGroupModule` registra `cmsComponents.ConfiguratorForm → ConfiguratorGroupComponent` e `ConfiguratorFormModule` lo rimappa su `ConfiguratorFormComponent` — feature-libs/product-configurator/rulebased/components/group/configurator-group.module.ts:L61-L68, feature-libs/product-configurator/rulebased/components/form/configurator-form.module.ts:L22-L29. Prevale il form: `ConfiguratorFormModule` importa `ConfiguratorGroupModule` (:L19-L19), Angular raccoglie i provider degli import prima dei propri (angular:packages/core/src/di/provider_collection.ts:L293-L346) e i `DefaultConfigChunk` sono fusi con `deepMerge` in ordine, l'ultimo vince (core-libs/core/src/config/utils/deep-merge.ts:L18-L49, core-libs/core/src/config/services/configuration.service.ts:L56-L89). Se si importa solo `ConfiguratorGroupModule` in un modulo custom, `ConfiguratorForm` diventa il gruppo.
10. **Nomi CMS senza mapping.** `ConfiguratorOverviewMenu` e `ConfiguratorOverviewFilter` stanno nei `cmsComponents` della root feature ma nessun modulo del repo registra un mapping con quelle chiavi — feature-libs/product-configurator/rulebased/root/rulebased-configurator-root-feature.module.ts:L14-L14, :L28-L28. I due componenti sono resi direttamente come elementi dentro la sidebar (`cx-configurator-overview-filter`, `cx-configurator-overview-menu`) e il filtro anche nel dialog filtro — feature-libs/product-configurator/rulebased/components/overview-sidebar/configurator-overview-sidebar.component.html:L35-L40, feature-libs/product-configurator/rulebased/components/overview-filter-dialog/configurator-overview-filter-dialog.component.html:L30-L30.
11. **Typo nelle stringhe action.** `'[Configurator] Create Configuration Sucess'`, `'… Read Configuration Sucess'`, doppio spazio in `SEARCH_VARIANTS_FAIL`, classi `DissmissConflictDialoge`/`CheckConflictDialoge` — feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L16-L21, feature-libs/product-configurator/rulebased/core/state/actions/configurator-variant.action.ts:L12-L12, feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L496-L504. Chi filtra le action per stringa (DevTools, meta-reducer) deve usare quelle esatte.
12. `UPDATE_CART_ENTRY_SUCCESS` **morta.** Definita ma mai usata — feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L29-L29.

13. **Configurazione senza attributi.** `getFirstGroupWithAttributes` lancia `Configuration does not have any attributes`, ed è chiamato in create, cart read, errore update e in `updateConfigurationSuccess$` (quando il gruppo corrente manca o è un conflitto) — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic-effect.service.ts:L112-L125, feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L440-L445. In create e cart read l'eccezione finisce nel `catchError` interno e diventa un'action `…Fail`; in `updateConfigurationSuccess$` il throw avviene nel `map` senza `catchError`, quindi fa fallire l'effect (che viene risottoscritto dal `defaultEffectsErrorHandler` di NgRx).
14. **Assignment mancante = niente render.** `assignment ?? []` usa un array come mappa; una chiave assente produce solo un warning in dev mode — feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.directive.ts:L37-L39, :L82-L88.
15. **Connector non in root.** `RulebasedConfiguratorConnector` e `ConfiguratorTextfieldConnector` non sono `providedIn: 'root'`: iniettarli fuori dal modulo lazy dà `NullInjectorError` — feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L18-L20, feature-libs/product-configurator/textfield/core/connectors/configurator-textfield.connector.ts:L14-L16.
16. **Handler 400 che inghiotte.** Con toggle CPQ container, `hasMatch` fa match sui 400 product-card ma `handleError` non mostra nulla — feature-libs/product-configurator/rulebased/root/http-interceptors/configurator-bad-request.handler.ts:L35-L48, :L84-L94. Il `BadRequestHandler` del core ha `Priority.LOW` — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L161-L163 — e `resolveApplicable` sceglie un solo handler, quello a priorità più alta — core-libs/core/src/util/applicable.ts:L50-L57 — quindi sul 400 product-card vince l'handler del configuratore (NORMAL) e il messaggio del core non compare.
17. **Page layout handler e sincronia.** Gli handler VC/CPQ riassegnano `slots$` dentro una `subscribe` con `take(1)`: la sostituzione funziona solo se `extractRouterData()` emette sincronamente — feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-page-layout-handler.ts:L57-L63, feature-libs/product-configurator/rulebased/root/variant/variant-configurator-page-layout-handler.ts:L41-L67. DA VERIFICARE: dipende dal replay dello stato router.
18. **Textfield: incoerenze.** `UpdateConfiguration` è un `LoaderLoadAction` senza success, quindi `loading` resta true dopo un update locale — feature-libs/product-configurator/textfield/core/state/actions/configurator-textfield.action.ts:L71-L76; `readConfigurationForCartEntry` dispatcha a ogni emissione di `requireLoadedCart()` — feature-libs/product-configurator/textfield/core/facade/configurator-textfield.service.ts:L180-L201; la lettura order entry usa sempre `OCC_USER_ID_CURRENT` e l'endpoint ordine anche per le quote — :L228-L234, feature-libs/product-configurator/textfield/occ/occ-configurator-textfield.adapter.ts:L108-L120; il bottone ha `routerLink` al carrello e `(click)`, quindi naviga senza attendere l'add-to-cart — feature-libs/product-configurator/textfield/components/add-to-cart-button/configurator-textfield-add-to-cart-button.component.html:L1-L5.
19. **Verbi HTTP diversi.** Update cart entry: VC = PUT, textfield = POST — feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.adapter.ts:L239-L241, feature-libs/product-configurator/textfield/occ/occ-configurator-textfield.adapter.ts:L154-L156.
20. **Codice CPQ non usato.** `getCpqAccessData`, `getConfigIdForCartEntry`, `getConfigIdForOrderEntry` non sono usati altrove nel repo — feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L14-L14, feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.service.ts:L82-L122.
21. **Test e `dist/`.** `feature-libs/product-configurator/vitest.config.ts` imposta `process.env['NX_TSCONFIG_PATH']` su `tsconfig.spec.json` prima di `defineConfig`: il commento spiega che altrimenti `nxViteTsPaths` userebbe `tsconfig.lib.json`, che mappa `@spartacus/*` su `dist/`, e dopo una build codice di libreria e helper di test avrebbero copie separate degli injection token (es. `FeatureToggles`) —

feature-libs/product-configurator/vitest.config.ts:L13-L16. La coverage esclude inoltre `textfield/root/**` — feature-libs/product-configurator/vitest.config.ts:L47-L59.

---

10. Come fare debug

- **Owner e tipo.** Quasi ogni bug parte da un `owner` sbagliato. Iscriversi a `ConfiguratorRouterExtractorService.extractRouterData()` e verificare `owner.key`, `configuratorType` e `pageType`; ricordare che non emette senza `entityKey` né durante navigazioni in corso — feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L28-L34. Un `Not able to determine configurator type` indica un `cxRoute` che non inizia con `configure` — :L101-L113.
- **"No adapter found for configurator type".** Il tipo dall'URL non ha un adapter nel token: tipicamente si è aperta una route CPQ senza aver caricato `RulebasedCpqConfiguratorModule` nel modulo lazy — feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L160-L190, projects/storefrontapp/src/app/spartacus/features/product-configurator/rulebased-configurator-wrapper.module.ts:L12-L21.
- **Redux DevTools.** Nello slice `productConfigurator.configurations.entities[<ownerKey>]` controllare `processesCount` (update in volo), `loading`, `error` e `value.interactionState`. Sequenza attesa per un update: `Update Configuration` → … `Success` → … `Finalize Success` + `Update Price Summary` (capitolo 19). Per un add-to-cart: `Add To Cart` → `Add Next Owner` + `CartAddEntrySuccess` → `Set Next Owner Cart Entry` + `Set Interaction State`.
- **Click ignorati.** Cercare in console (dev mode) il warning `CART_BUSY_NO_UPDATES_MSG` — feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L139-L151.
- **Attributo non renderizzato.** Warning dev mode "No attribute type component available for: ..." della direttiva — feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.directive.ts:L82-L88; controllare la chiave calcolata (`AttributeType_` + `uiType` o `uiTypeVariation`) — feature-libs/product-configurator/rulebased/components/group/configurator-group.component.ts:L121-L127 — e il valore normalizzato di `uiType` — feature-libs/product-configurator/rulebased/occ/variant/converters/occ-configurator-variant-normalizer.ts:L440-L461.
- **Network.** VC: `ccpconfigurator/…` (GET create, PATCH update, GET pricing); CPQ: `cpqconfigurator/…` con `tabId` e un PATCH per attributo. Un `${tabId}` letterale nell'URL è la trappola 6 di §9.
- **Configurazione "che si ricrea".** Se dopo un'azione si vede un nuovo `Create Configuration`, una `RemoveConfiguration` sulla chiave product-bound è avvenuta mentre `getOrCreateConfiguration` era ancora sottoscritto (§3.7 e commento in feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L448-L458).
- **Messaggio 400 mancante.** Verificare il toggle `productConfiguratorCPQContainer` e la URL della richiesta (product card) — trappola 16.

---

11. Domande di autoverifica

1. **Come decide il connector se usare l'adapter VC o quello CPQ?** Confronta `owner.configuratorType` (ricavato dal suffisso del semanticRoute) con `getConfiguratorType()` di ogni adapter del token multi `CONFIGURATOR_ADAPTER_LIST`; se nessuno combacia lancia errore. — feature-libs/product-configurator/

rulebased/core/connectors/rulebased-configurator.connector.ts:L160-L190, feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L101-L113.

2. **Perché** `RulebasedConfiguratorConnector` **non è** `providedIn: 'root'`**?** Perché gli adapter sono registrati nel modulo lazy: un connector in root verrebbe creato nell'injector root e non vedrebbe gli adapter lazy. Il core lo fornisce esplicitamente. — feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L18-L18, feature-libs/product-configurator/rulebased/core/rulebased-configurator-core.module.ts:L16-L34.
3. **Che chiave ha nello store la configurazione di un prodotto** `ABC` **dopo l'add-to-cart come entry 2?** Prima `product/ABC`; dopo l'add-to-cart il reducer imposta `nextOwner = cartEntry/2` e `addOwner$` copia l'`interactionState` sull'entità `cartEntry/2`. — feature-libs/product-configurator/common/shared/utils/configurator-model-utils.ts:L19-L27, feature-libs/product-configurator/rulebased/core/state/reducers/configurator.reducer.ts:L290-L305, feature-libs/product-configurator/rulebased/core/state/effects/configurator-cart.effect.ts:L250-L285.
4. **Perché gli update usano** `concatMap` **e non** `switchMap`**?** Ogni update invia un solo attributo cambiato; `switchMap` cancellerebbe richieste in volo perdendo modifiche. `concatMap` le mette in coda. — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L178-L181.
5. **Quando viene impostato** `isCartEntryUpdateRequired` **e a cosa serve?** A `true` nel finalize di un update, a `false` su `UPDATE_CART_ENTRY`; il bottone "update cart" chiama `updateCartEntry` solo se è true. — feature-libs/product-configurator/rulebased/core/state/reducers/configurator.reducer.ts:L122-L134, :L175-L181, feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L463-L482.
6. **Cosa succede se un update fallisce?** `UpdateConfigurationFail` decrementa i processi; a processi esauriti si emette `UpdateConfigurationFinalizeFail`, che `handleErrorOnUpdate$` trasforma in una `ReadConfiguration` del primo gruppo con attributi. — feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic.effect.ts:L516-L585.
7. **Come si aggiunge un componente per un nuovo tipo di attributo?** Registrando in un modulo `provideDefaultConfig({productConfigurator: {assignment: {AttributeType_<tipo>: MioComponente}}})`; il componente riceve `ConfiguratorAttributeCompositionContext` dall'injector creato dalla direttiva. — feature-libs/product-configurator/rulebased/components/attribute/types/radio-button/configurator-attribute-radio-button.module.ts:L33-L43, feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.directive.ts:L74-L101.
8. **Perché, passando all'overview dopo l'add-to-cart, il codice fa due navigazioni?** La prima, con `replaceUrl: true`, riscrive la voce di history della pagina di configurazione con owner `cartEntry`, così il tasto back torna alla configurazione della entry; la seconda va all'overview. — feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.component.ts:L172-L202.
9. **Che effetto collaterale globale ha importare** `CpqConfiguratorRootModule`**?** Fornisce `routing.protected: true` come default config ("force early login", sovrascrivibile da `provideConfig`): l'intera applicazione diventa protetta, salvo le route con `protected: false` esplicito. — feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-root.module.ts:L16-L21.
10. **In cosa differisce lo store del textfield da quello rulebased?** Il textfield ha un singolo `LoaderState` (una configurazione alla volta) e l'update è solo locale; il rulebased ha un `EntityProcessesLoaderState` indicizzato per owner key con update inviati al backend. — feature-libs/product-configurator/textfield/core/state/configuration-textfield-state.ts:L10-L20, feature-libs/product-configurator/textfield/core/facade/configurator-textfield.service.ts:L81-L101, feature-libs/product-configurator/rulebased/core/state/configurator-state.ts:L10-L19.

TAVOLA 30.1 **4.1 Moduli eager e lazy**

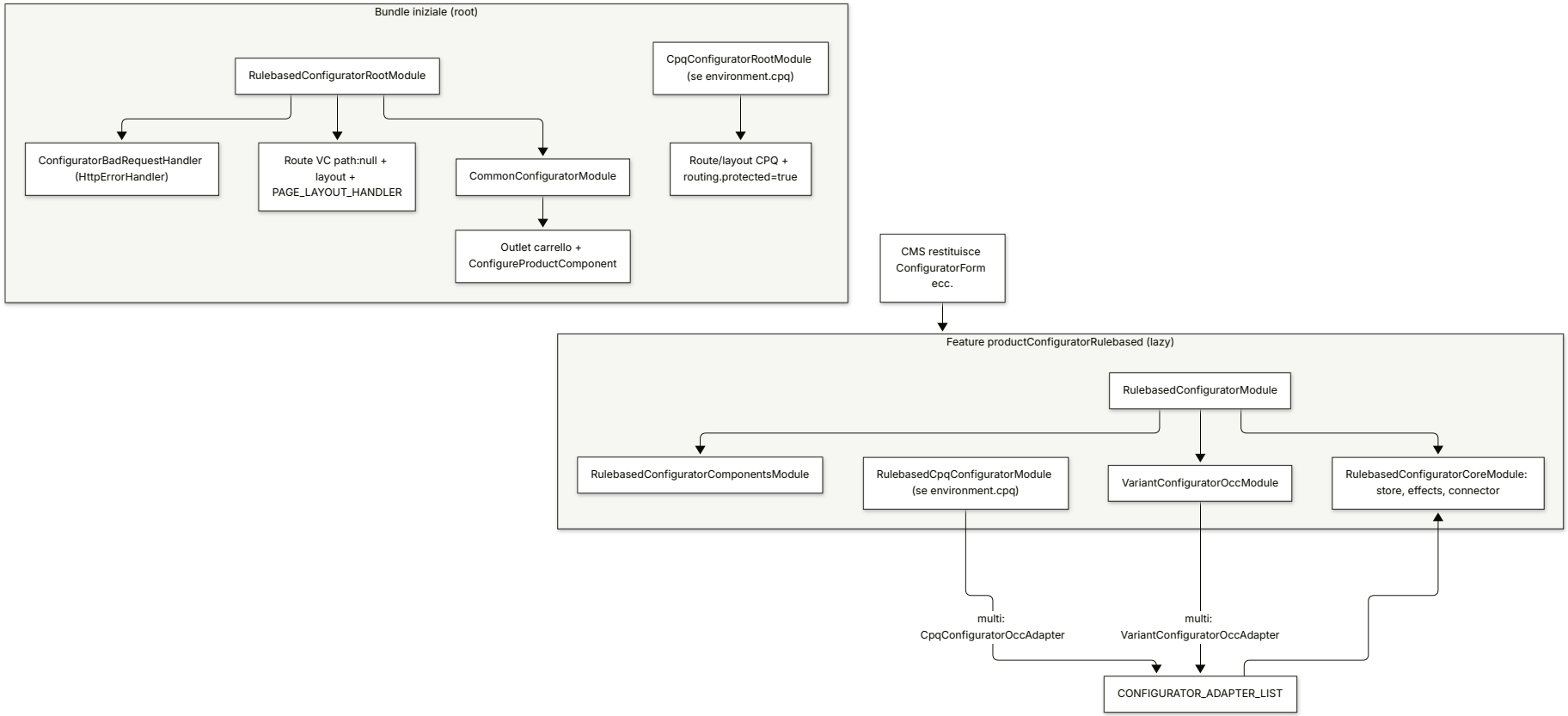


TAVOLA 30.2 **4.2 Dal PDP alla configurazione creata (VC)**

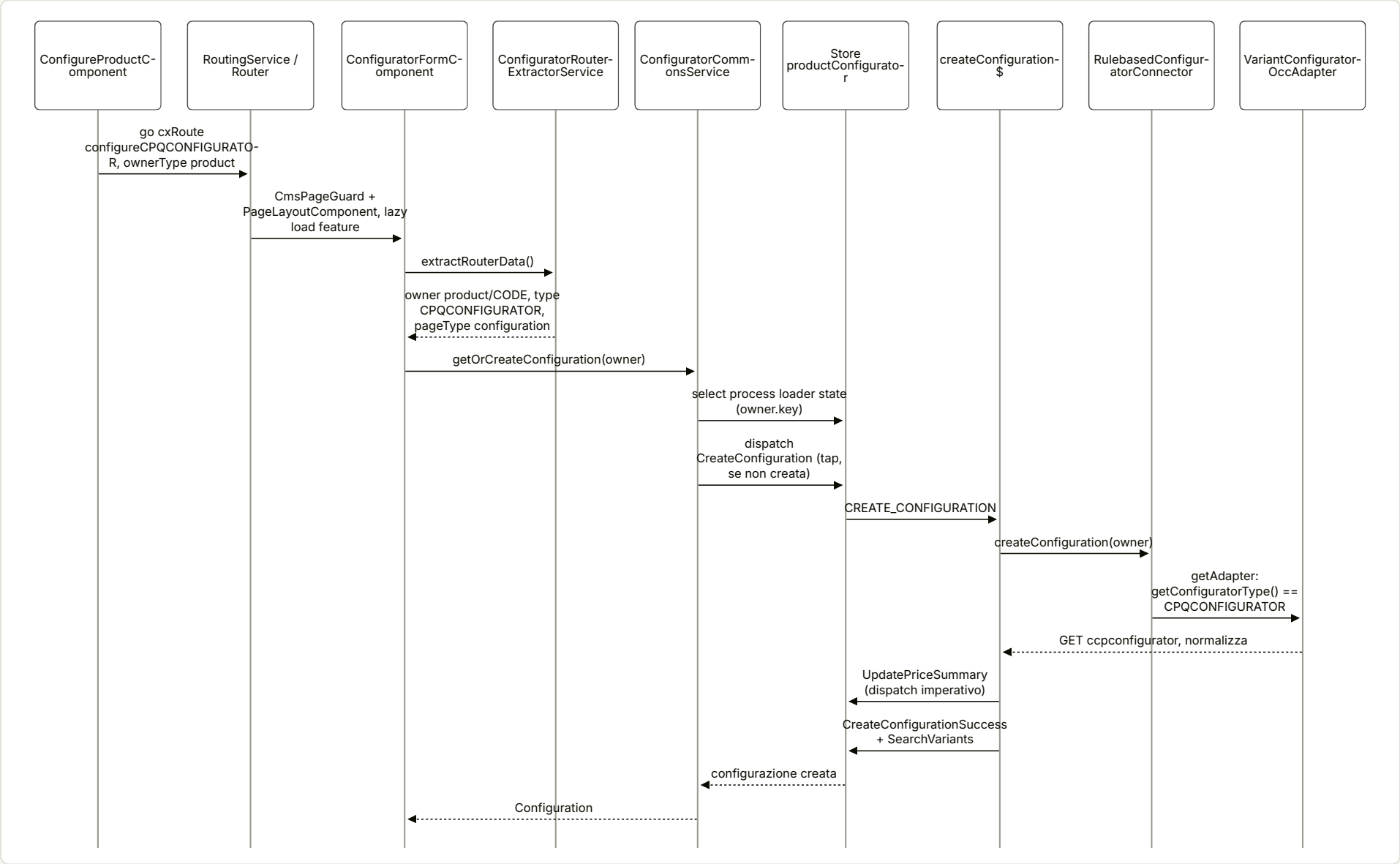


TAVOLA 30.3 **4.3 Modifica di un attributo**

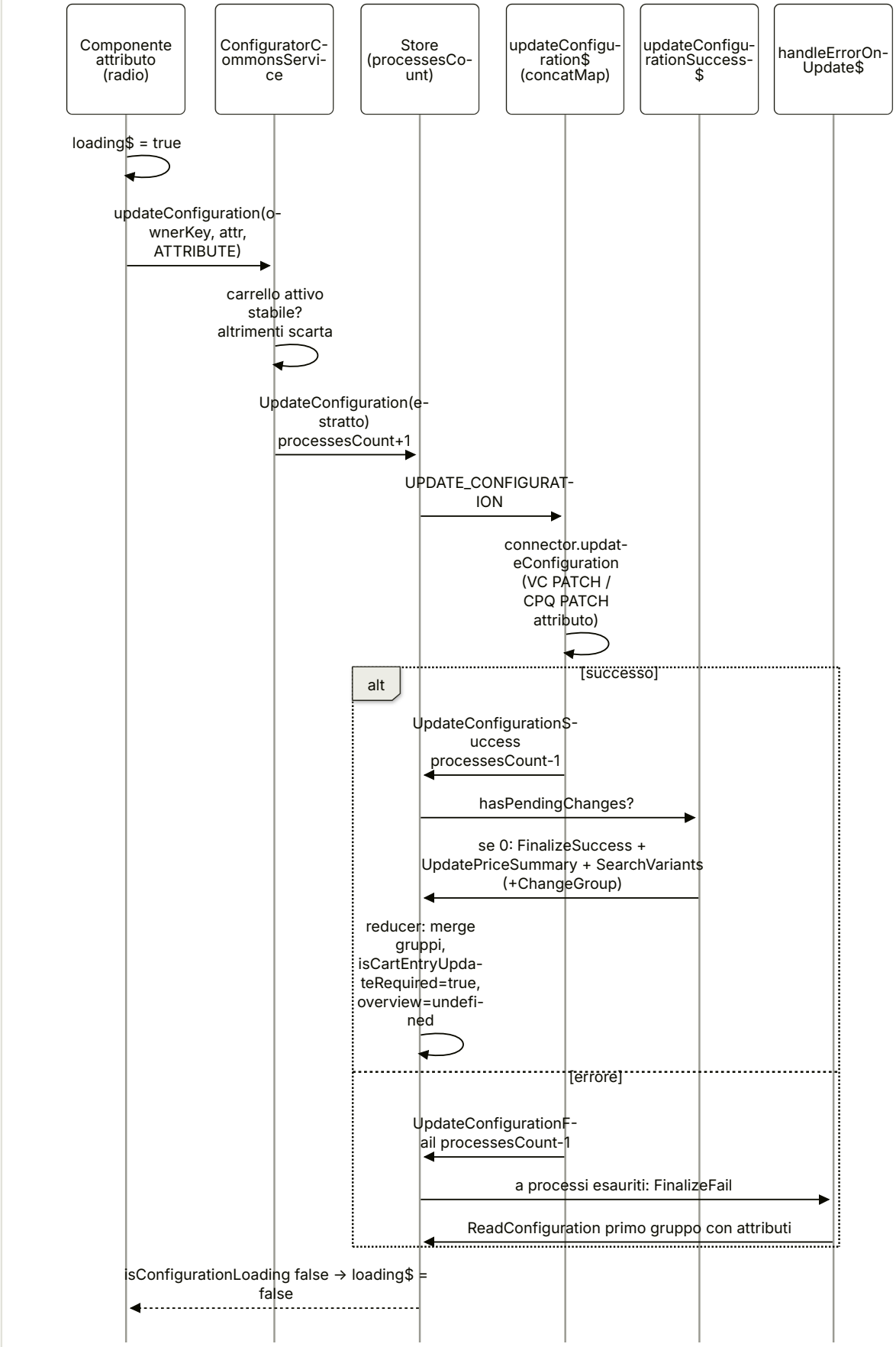
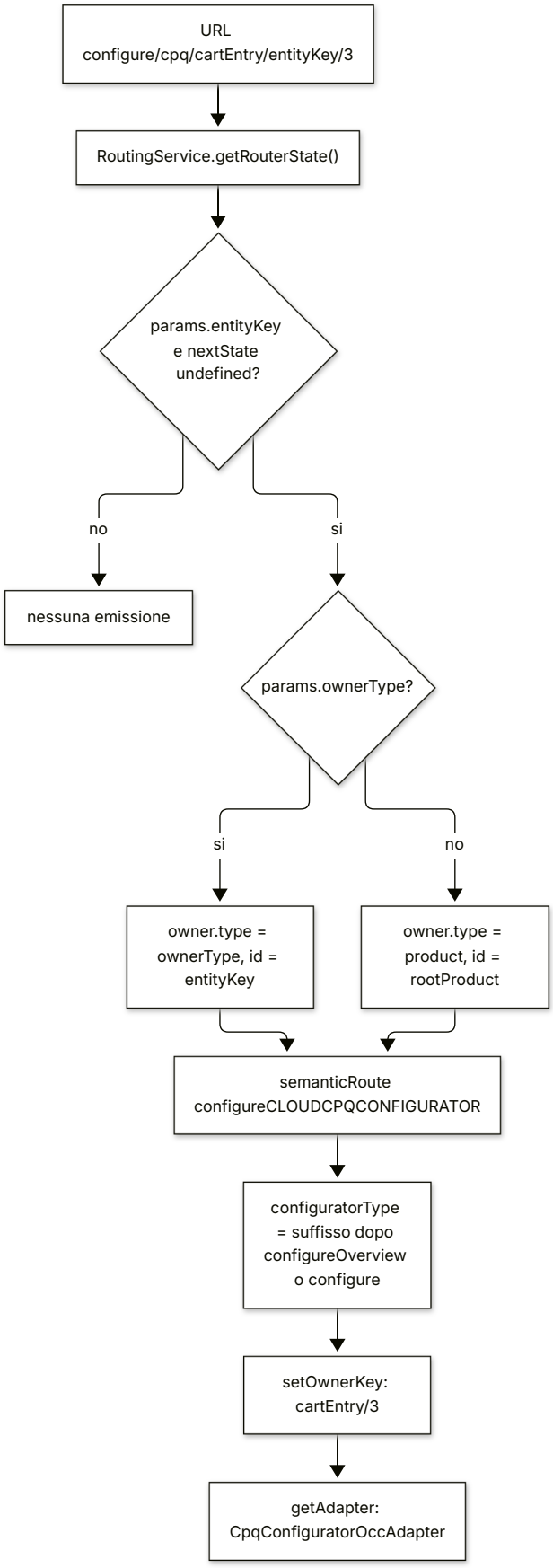


TAVOLA 30.4 **4.5 Risoluzione dell'owner e del tipo dall'URL**



31

Quote, customer ticketing, subscription billing

Tre feature lib di account con la stessa architettura a quattro strati, usata in modi diversi.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Questo capitolo copre tre feature library "di account" che vivono sotto `feature-libs/` e che condividono la stessa architettura a quattro strati (`root` eager, `core`, `occ`, `components` lazy), ma la usano in modi molto diversi:

LIBRERIA	PACCHETTO	COSA FA PER L'UTENTE	TRATTO ARCHITETTURALE DISTINTIVO
Quote	`@spartacus/quote`	negoziazione B2B di un preventivo tra buyer e seller (richiesta dal carrello, commenti, sconti, scadenza, checkout della quote)	macchina a stati guidata dal backend (`allowedActions`), "quote cart" che sostituisce il carrello attivo, guard CMS che blocca la navigazione, tre chunk lazy distinti
Customer ticketing	`@spartacus/customer-ticketing`	ticket di assistenza in My Account: lista, creazione, messaggi, allegati, chiusura e riapertura	`Command` con strategia `Queue`, orchestrazione tramite eventi (`CxEvent`) ascoltati da un listener eager
Subscription billing	`@spartacus/subscription-billing`	elenco e dettaglio degli abbonamenti, azioni (cancel/withdraw/resubscribe/extend), fatture, prezzi "a piano" in PDP e carrello	componenti standalone con signals, facade che restituiscono Observable freddi, outlet del carrello forniti dal modulo root eager

Tutte e tre le librerie seguono il pattern descritto in 12 — Feature module e lazy loading: il modulo `*RootModule` viene importato eagerly e dichiara `featureModules` (componenti CMS → nome feature), route e facade proxy; il resto (`*Module` = components + core + occ) viene caricato on demand. La facade lazy (`facadeFactory`) e il pattern Command/Query sono quelli di 20 — Facade, Command/Query, eventi; gli adapter OCC seguono 21 — OCC.

Quote è la più complessa e la più interessante: lo stato di una quote (17 valori di `QuoteState`, `feature-libs/quote/root/model/quote.model.ts:L75-L93`) e le azioni possibili (`allowedActions`) arrivano dal backend, ma Spartacus le **filtra, riordina e classifica** lato client con configurazione (`feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L30-L53`) e collega la quote al carrello: quando il buyer modifica o compra una quote, il carrello attivo diventa il "quote cart" e un guard CMS impedisce di uscire dal flusso (`feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L34-L57`). Per questo il capitolo dedica la maggior parte dello spazio a quote, e tratta ticketing e subscription billing per le differenze.

Prerequisiti utili: 25 — Cart (active cart, multi-cart, saved cart), 26 — Checkout, 18 — Layout, breakpoint, launch dialog (i dialog `LAUNCH_CALLER`), 17 — Outlet, 41 — Errori e global message (`HttpErrorHandler`).

2. Mappa dei file

2.1 `feature-libs/quote`

PATH	RUOLO
`feature-libs/quote/quote.module.ts`	`QuoteModule` = `QuoteComponentsModule` + `QuoteCoreModule` + `QuoteOccModule` (target del lazy load feature `quote`)
`feature-libs/quote/root/feature-name.ts`	tre nomi feature: `quote`, `quote_cart_guard`, `quote_request`

PATH	RUOLO
feature-libs/quote/root/quote-root.module.ts	eager: mapping CMS → feature, route quotes/quoteDetails, layout QuoteDetailsPageTemplate, ng-route con AuthGuard+CmsPageGuard, due HttpErrorHandler
feature-libs/quote/root/facade/quote.facade.ts	QuoteFacade astratta + proxy lazy (8 metodi proxati)
feature-libs/quote/root/model/quote.model.ts	Quote, QuoteAction, QuoteActionType, QuoteState, QuoteRoleType, QuoteDiscountType, QuoteMetadata, QuoteStarter
feature-libs/quote/root/model/augmented-types.ts	augment di OrderEntry.comments e di LAUNCH_CALLER.QUOTE_ACTION_CONFIRMATION
feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts	QuoteBadRequestHandler (400 legati alle quote)
feature-libs/quote/root/http-interceptors/quote-not-found.handler.ts	QuoteNotFoundHandler (404 Quote not found → redirect a quotes)
feature-libs/quote/root/config/quote.config.ts	QuoteConfig vuota agganciata a Config.quote
feature-libs/quote/core/quote-core.module.ts	facade provider, QuoteConnector, defaultQuoteCoreConfig, istanzia QuoteCartEventListener
feature-libs/quote/core/facade/quote.service.ts	QuoteService: tutti i command e le query
feature-libs/quote/core/facade/facade-providers.ts	QuoteFacade → useExisting: QuoteService
feature-libs/quote/core/config/quote-core.config.ts, default-quote.core.config.ts	QuoteCoreConfig.quote.actions (primarie e ordine per stato)
feature-libs/quote/core/connectors/quote.adapter.ts, quote.connector.ts, converters.ts	adapter astratto (9 operazioni), connector 1:1, 7 token di conversione
feature-libs/quote/core/event/quote.events.ts, quote-cart-event.listener.ts	QuoteDetailsReloadQueryEvent e traduzione eventi carrello → reload
feature-libs/quote/core/services/quote-cart.service.ts	QuoteCartService: "il carrello attivo è un quote cart?", checkoutAllowed
feature-libs/quote/core/services/cart-utils.service.ts	CartUtilsService.handleCartAndGoToQuoteList
feature-libs/quote/core/services/quote-storefront-utils.service.ts	accesso DOM SSR-safe
feature-libs/quote/occ/quote-occ.module.ts	QuoteAdapter → OccQuoteAdapter, normalizer multi su QUOTE_NORMALIZER
feature-libs/quote/occ/adapters/occ-quote.adapter.ts	chiamate HTTP verso users/${userId}/quotes...
feature-libs/quote/occ/config/default-occ-quote-config.ts	10 endpoint di default
feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts	filtro/ordine/primarietà delle azioni, calcolo isEditable
feature-libs/quote/occ/converters/occ-quote-entry-normalizer.ts	quoteCode sulle entry, PRODUCT_NORMALIZER sui prodotti

PATH	RUOLO
feature-libs/quote/components/quote-components.module.ts	aggregatore di 12 moduli UI + defaultQuoteUIConfig
feature-libs/quote/components/config/quote-ui.config.ts, default-quote-ui.config.ts	QuoteUIConfig (commenti, troncamento, dialog di conferma, sconto, debounce)
feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts	bottoni azione, logica dialog di conferma
feature-libs/quote/components/summary/seller-edit/*	sconto e scadenza (lato seller)
feature-libs/quote/components/header/*	testata, form buyer (nome, descrizione)
feature-libs/quote/components/items/*	entry: da quote, saved cart read-only o active cart
feature-libs/quote/components/list/*	lista quote (AccountMyQuotesComponent)
feature-libs/quote/components/links/*	link: nuovo carrello, download PDF proposta, ordine
feature-libs/quote/components/comments/*	messaggistica buyer/seller
feature-libs/quote/components/confirm-dialog/*	QuoteConfirmDialogComponent + layout QUOTE_ACTION_CONFIRMATION
feature-libs/quote/components/cart-guard/*	entry point secondario: QuoteCartGuardComponent marker + QuoteCartGuard
feature-libs/quote/components/request-button/*	entry point secondario: QuoteRequestButtonComponent (componente CMS QuoteRequestComponent)
projects/storefrontapp/src/app/spartacus/features/quote-feature.module.ts	wiring demo: tre import() distinti
core-libs/schematics/src/shared/lib-configs/quote-schematics-config.ts	configurazione schematics (b2b, dipendenza da cart base)

2.2 feature-libs/customer-ticketing

PATH	RUOLO
feature-libs/customer-ticketing/customer-ticketing.module.ts	CustomerTicketingModule = components + core + occ
feature-libs/customer-ticketing/root/customer-ticketing-root.module.ts	6 componenti CMS, alias feature core, due ng-route, CustomerTicketingEventModule
feature-libs/customer-ticketing/root/customer-ticketing-constants.ts	limiti (5000/255 caratteri, 10 MB, 1 allegato, page size 5)
feature-libs/customer-ticketing/root/config/*	CustomerTicketingConfig, default, routing
feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts	eventi reset/reload query ed eventi di dominio
feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts	listener eager: messaggi e reload/reset
feature-libs/customer-ticketing/root/facade/customer-ticketing.facade.ts	facade lazy (12 metodi)

PATH	RUOLO
feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts	STATUS, STATUS_NAME, TicketDetails, TicketEvent
feature-libs/customer-ticketing/root/model/augmented-core.model.ts	LAUNCH_CALLER.CUSTOMER_TICKETING_*
feature-libs/customer-ticketing/core/customer-ticketing-core.module.ts	facade, connector, NotFoundTicketRequestHandler, CustomerTicketingPageMetaResolver
feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts	CustomerTicketingService (command Queue, query)
feature-libs/customer-ticketing/core/http-interceptors/handlers/not-found-ticket-request.handler.ts	404 sul dettaglio → lista
feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts	heading e breadcrumb del dettaglio
feature-libs/customer-ticketing/occ/adapters/occ-customer-ticketing.adapter.ts	HTTP, FormData per upload
feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts	endpoint
feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/*	CustomerTicketingDialogComponent astratta + layout dei tre dialog
feature-libs/customer-ticketing/components/details/*	dettagli, messaggi, close, reopen
feature-libs/customer-ticketing/components/list/*	lista e create
feature-libs/customer-ticketing/components/my-account-v2/*	widget "ultimo ticket"
projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts	wrapper demo che aggiunge CDP se environment.cdp

2.3 feature-libs/subscription-billing

PATH	RUOLO
feature-libs/subscription-billing/subscription-billing.module.ts	SubscriptionBillingModule = components + core + occ
feature-libs/subscription-billing/root/subscription-billing-root.module.ts	5 componenti CMS, alias core, routing, **due outlet del carrello** (eager)
feature-libs/subscription-billing/root/components/cart/*	SubscriptionCartPriceHeadingComponent, SubscriptionCartPriceBodyComponent (standalone, eager)
feature-libs/subscription-billing/root/services/subscription-product.service.ts	SubscriptionProductService: riconosce i prodotti subscription
feature-libs/subscription-billing/root/model/subscription-product.model.ts	augment di Product e ProductScope.SUBSCRIPTION
feature-libs/subscription-billing/root/model/subscription.model.ts	SubscriptionDetail, SubscriptionActionMode, LAUNCH_CALLER aggiuntivi
feature-libs/subscription-billing/root/facade/*	SubscriptionFacade, SubscriptionBillingFacade, SubscriptionActionsFacade
feature-libs/subscription-billing/root/config/extend-subscription-dropdown-options-config.ts	ExtendSubscriptionFrequencyDropdownOptionsConfig

PATH	RUOLO
feature-libs/subscription-billing/root/events/subscription-billing-event.listener.ts	reload su cambio lingua/valuta
feature-libs/subscription-billing/core/facade/*	SubscriptionService, SubscriptionBillingService, SubscriptionActionsService, facade provider
feature-libs/subscription-billing/core/connector/*	tre coppie adapter/connector
feature-libs/subscription-billing/core/services/*	due page meta resolver
feature-libs/subscription-billing/occ/adapters/*	tre adapter OCC (senza converter)
feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts	endpoint + scope prodotto subscription
feature-libs/subscription-billing/components/actions-modal/*	dialog azioni (signals, effect)
feature-libs/subscription-billing/components/list/*	lista subscription, lista bill, DateRangeModalComponent
feature-libs/subscription-billing/components/product/*	prezzo e usage charge in PDP

File aggiuntivi

PATH	RUOLO
feature-libs/quote/public_api.ts	Entry @spartacus/quote: riferimento triple-slash a types.d.ts (per @HostListener con eventi tipo keydown.Escape) + riesportazione di quote.module (:L14-L19).
feature-libs/quote/schematics/add-quote/index.ts	addQuoteFeatures(): factory degli schematics ng-add/add di @spartacus/quote (vedi 3.12).
feature-libs/quote/schematics/add-quote/index_spec.ts	Test dello schematic, incluso il controllo che la configurazione b2b venga aggiunta.
feature-libs/quote/schematics/constants.ts	Costanti schematics (QUOTE_FOLDER_NAME, QUOTE_TRANSLATIONS, ...), duplicate di quelle in core-libs schematics (vedi 3.12).
feature-libs/quote/vitest.config.ts	Config Vitest: jsdom, pool forks, alias per mock feature directive e url-testing.module, coverage 85/65.
feature-libs/quote/_index.scss	Entry SCSS: core + app Spartacus, mixin locale visible-focus, poi styles/index.
feature-libs/quote/styles/_index.scss	Importa mixin, 12 file componente e layout; $quote-components-allowlist con @extend.
feature-libs/quote/styles/_quote-comments.scss	Placeholder %cx-quote-comments.
feature-libs/quote/styles/_quote-confirm-dialog.scss	Placeholder %cx-quote-confirm-dialog.
feature-libs/quote/styles/_quote-header-buyer-edit.scss	Placeholder %cx-quote-header-buyer-edit (importa lo stile card).

PATH	RUOLO
feature-libs/quote/styles/_quote-header-overview.scss	Placeholder %cx-quote-header-overview + regole RTL.
feature-libs/quote/styles/_quote-items.scss	Placeholder %cx-quote-items.
feature-libs/quote/styles/_quote-links.scss	Placeholder %cx-quote-links + regole LTR/RTL.
feature-libs/quote/styles/_quote-list.scss	Placeholder %cx-quote-list + regole LTR/RTL.
feature-libs/quote/styles/_quote-request-button.scss	Placeholder %cx-quote-request-button.
feature-libs/quote/styles/_quote-summary-actions.scss	Placeholder %cx-quote-summary-actions.
feature-libs/quote/styles/_quote-summary-prices.scss	Placeholder %cx-quote-summary-prices.
feature-libs/quote/styles/_quote-summary-seller-edit.scss	Placeholder %cx-quote-summary-seller-edit.
feature-libs/quote/styles/_quote-summary.scss	Placeholder %cx-quote-summary.
feature-libs/quote/styles/layout/_index.scss	Allowlist/blocklist del page template QuoteDetailsPageTemplate (variabili con suffisso -store-finder, vedi 3.12).
feature-libs/quote/styles/layout/_quote-details-page.scss	Placeholder %QuoteDetailsPageTemplate: flex 70/30 tra BodyContent e CenterRightContent.
feature-libs/quote/styles/mixins/_mixins.scss	Importa i due mixin sotto.
feature-libs/quote/styles/mixins/_ghost-bar.scss	Mixin cx-ghost-bar.
feature-libs/quote/styles/mixins/_quote-toggle.scss	Mixin cx-quote-toggle.
feature-libs/customer-ticketing/_index.scss	Entry SCSS: core + blocco modal di cxbase + styles/index.
feature-libs/customer-ticketing/vitest.config.ts	Config Vitest: alias @spartacus/core/testing/mock-feature-toggles, coverage 85/65.
feature-libs/subscription-billing/public_api.ts	Entry @spartacus/subscription-billing: riesporta subscription-billing.module (il commento dice "segment", residuo di copia).
feature-libs/subscription-billing/assets/public_api.ts	Entry assets: riesporta translations/translations.
feature-libs/subscription-billing/vitest.config.ts	Config Vitest senza alias, coverage 85/65.
feature-libs/subscription-billing/_index.scss	Entry SCSS: importa solo styles/index.
feature-libs/subscription-billing/styles/_index.scss	Importa core + 10 file componente; $subscription-billing-components-allowlist con @extend.
feature-libs/subscription-billing/styles/billing-details/_subscription-billing-details.scss	Placeholder %cx-subscription-billing-details.
feature-libs/subscription-billing/styles/cart/_subscription-cart-item-list.scss	Placeholder %cx-subscription-cart-item-list.

PATH	RUOLO
feature-libs/subscription-billing/styles/cart/_subscription-cart-price-body.scss	Placeholder %cx-subscription-cart-price-body (outlet carrello eager).
feature-libs/subscription-billing/styles/details/_subscription-details.scss	Placeholder %cx-subscription-details.
feature-libs/subscription-billing/styles/list/_billing-list.scss	Placeholder %cx-billing-list.
feature-libs/subscription-billing/styles/list/_date-range-modal.scss	Placeholder %cx-date-range-modal.
feature-libs/subscription-billing/styles/list/_subscription-list.scss	Placeholder %cx-subscription-list (e regole condivise con %cx-subscription-details).
feature-libs/subscription-billing/styles/modal/_subscription-modal.scss	Placeholder %cx-subscription-actions-modal.
feature-libs/subscription-billing/styles/product/_subscription-product-price.scss	Placeholder %cx-subscription-product-price.
feature-libs/subscription-billing/styles/product/_subscription-product-usage-charge.scss	Placeholder %cx-subscription-product-usage-charge.

3. Come funziona passo per passo

3.1 Quote: packaging in tre chunk

La lib quote ha **tre** nomi feature (feature-libs/quote/root/feature-name.ts:L7-L9). defaultQuoteComponentsConfig assegna sei componenti CMS (lista, link, commenti, testata, entry, summary) alla feature quote (feature-libs/quote/root/quote-root.module.ts:L29-L44), mentre QuoteCartGuardComponent e QuoteRequestComponent vanno su feature separate (feature-libs/quote/root/quote-root.module.ts:L46-L64). Il motivo è il peso: il bottone "Request quote" sta nella pagina carrello e il marker del guard sta (presumibilmente) in carrello e checkout; caricarli non deve trascinare tutto il chunk quote.

Nell'app demo le tre feature corrispondono a tre import() distinti: @spartacus/quote, @spartacus/quote/components/cart-guard, @spartacus/quote/components/request-button (projects/storefrontapp/src/app/spartacus/features/quote-feature.module.ts:L26-L63). I due chunk piccoli sono entry point ng-packagr secondari e infatti components/public_api.ts non li riesporta (feature-libs/quote/components/public_api.ts:L7-L15).

QuoteModule importa solo components, core e occ (feature-libs/quote/quote.module.ts:L12-L15). Lo schematics marca la lib come b2b e dipendente da cart base (core-libs/schematics/src/shared/lib-configs/quote-schematics-config.ts:L36-L84).

3.2 Quote: root eager (route, layout, error handler)

- Route semantiche: quotes → my-account/quotes e quoteDetails → my-account/quote/:quoteId (feature-libs/quote/root/quote-root.module.ts:L66-L78).
- Layout: QuoteDetailsPageTemplate con slot BodyContent e CenterRightContent (feature-libs/quote/root/quote-root.module.ts:L80-L86).

- Solo `quoteDetails` ha una ng-route esplicita (`path: null`, risolta dalla config di routing come spiegato in 14 — Routing), protetta da `AuthGuard` e `CmsPageGuard`; `quotes` è una content page CMS (`feature-libs/quote/root/quote-root.module.ts:L88-L106`).
- Due `HttpErrorHandler` multi: `QuoteBadRequestHandler` e `QuoteNotFoundHandler` (`feature-libs/quote/root/quote-root.module.ts:L108-L118`).

Gli handler sono registrati nel root perché devono esistere anche prima del caricamento lazy. Comunque `HttpErrorInterceptor` li raccoglie con `UnifiedInjector.getMulti(HttpErrorHandler)`, quindi vedrebbe anche provider di moduli lazy (`core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L26-L28`). Il `hasMatch` base confronta solo lo status (`core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L42-L44`); i due handler quote lo restringono:

- `QuoteBadRequestHandler` (400) fa match solo se l'errore è "legato alle quote" (`feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L28-L32`): tipi OCC `QuoteUnderThresholdError`, `CartValidationError`, `CartQuoteAccessError` (`feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L83-L101`), oppure `IllegalArgumentError` con messaggio che corrisponde a `/Discount type is absolute/` o `/Invalid quote expiration time/` (`feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L121-L133`). `handleError` aggiunge un global message per ogni categoria presente (`feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L34-L52`). Priorità `Priority.NORMAL` (`feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L54-L56`).
- `QuoteNotFoundHandler` (404) fa match solo per `NotFoundError` con messaggio esatto `'Quote not found'` e naviga a `quotes` (`feature-libs/quote/root/http-interceptors/quote-not-found.handler.ts:L29-L60`). Il JSDoc spiega lo scenario: fine di un'emulazione ASM mentre si guarda una quote, poi emulazione di un altro utente (`feature-libs/quote/root/http-interceptors/quote-not-found.handler.ts:L44-L50`).

3.3 Quote: la facade e il suo proxy

`QuoteFacade` è `providedIn: 'root'` con `useFactory: () => facadeFactory({...})`, feature `quote`, otto metodi proxati (`feature-libs/quote/root/facade/quote.facade.ts:L21-L38`). La classe astratta dichiara però anche `requote` e `downloadAttachment` (`feature-libs/quote/root/facade/quote.facade.ts:L99-L134`), che **non** sono nella lista `methods`. `FacadeFactoryService.create` costruisce il proxy come istanza di una sottoclasse vuota e definisce solo i metodi elencati (`core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L110-L131`): sul proxy `requote` è quindi il metodo astratto, cioè `undefined` a runtime (vedi §9).

Nel chunk lazy, `QuoteCoreModule` fornisce `QuoteFacade` con `useExisting: QuoteService` (`feature-libs/quote/core/facade/facade-providers.ts:L11-L17`), registra `QuoteConnector` e `defaultQuoteCoreConfig`, e istanzia nel costruttore `QuoteCartEventListener` (`feature-libs/quote/core/quote-core.module.ts:L14-L25`). I componenti dichiarati nel chunk `quote` ricevono quindi `QuoteService` direttamente; quelli dei chunk `cart-guard` e `request-button` passano dal proxy.

3.4 Quote: modello di dominio

`Quote` è derivato da `OccQuote`: `allowedActions` passa da array di tipi a `QuoteAction[]` (`{type, isPrimary}`) e si aggiunge `isEditable` (`feature-libs/quote/root/model/quote.model.ts:L41-L49`). Le azioni sono 14 (`QuoteActionType`, `feature-libs/quote/root/model/quote.model.ts:L58-L73`); gli stati 17 con prefisso di ruolo (BUYER_, SELLER_, SELLERAPPROVER_) più `CREATED`, `CANCELLED`, `EXPIRED` (`feature-libs/quote/root/model/quote.model.ts:L75-L93`). `QuoteRoleType` ha `ALL`, `BUYER`, `SELLER`, `SELLERAPPROVER`; `QuoteDiscountType` ha `PERCENT` e `ABSOLUTE` (`feature-libs/quote/root/model/quote.model.ts:L95-L105`).

Dettagli notevoli:

- `QuoteMetadata.expirationTime` è `string` di proposito, per inviare l'offset locale (`+02:00`) invece di UTC (feature-libs/quote/root/model/quote.model.ts:L120-L128).
- `QuoteStarter` contiene `cartId` (creazione dal carrello) o `quoteCode` (requote) (feature-libs/quote/root/model/quote.model.ts:L130-L133).
- `QuotesStateParams` passa pagina e sort come Observable (feature-libs/quote/root/model/quote.model.ts:L137-L140).
- `augmented-types.ts` aggiunge `comments` a `OrderEntry` e il valore `QUOTE_ACTION_CONFIRMATION` a `LAUNCH_CALLER`, sia nel tipo sia a runtime (feature-libs/quote/root/model/augmented-types.ts:L10-L23).

3.5 Quote: normalizzazione delle azioni (il cuore della "macchina a stati")

Il backend decide quali azioni sono possibili; il client decide **quali mostrare, in che ordine e con che enfasi**. `QuoteOccModule` registra due normalizer multi su `QUOTE_NORMALIZER`: `OccQuoteActionNormalizer` e `OccQuoteEntryNormalizer` (feature-libs/quote/occ/quote-occ.module.ts:L17-L37).

`OccQuoteActionNormalizer.convert` (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L30-L53):

1. Se esiste un ordine configurato per lo stato (`quote.actions.actionsOrderByState[state]`), clona la lista; se lo stato non è `BUYER_OFFER` e il carrello attivo è il quote cart **di questa quote**, toglie `EDIT` dalla lista clonata (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L60-L81).
2. Filtra le azioni OCC tenendo solo quelle presenti nell'ordine e le ordina di conseguenza; se non c'è ordine per lo stato restituisce la lista OCC invariata (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L82-L89).
3. Marca `isPrimary` se il tipo è in `quote.actions.primaryActions` (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L46-L53).
4. Calcola `isEditable = OCC permette EDIT && EDIT non è più tra le azioni visualizzate` (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L37-L43). Tradotto: `isEditable` è vero quando OCC permette `EDIT` ma `EDIT` non è tra le azioni normalizzate. Il caso tipico è la quote già nel quote cart attivo (EDIT rimosso dal normalizer); vale però anche per gli stati il cui ordine configurato non contiene `EDIT` (con i default `EXPIRED` e `CANCELLED`), dove `EDIT` viene filtrato via (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L82-L83, feature-libs/quote/core/config/default-quote.core.config.ts:L30-L31). Che OCC mandi `EDIT` per `EXPIRED` è previsto dalla UI config, che ha un dialog `EXPIRED.EDIT` (feature-libs/quote/components/config/default-quote-ui.config.ts:L41-L43).

I default (feature-libs/quote/core/config/default-quote.core.config.ts:L10-L37): primarie `APPROVE`, `CHECKOUT`, `SUBMIT`, `REQUOTE`; ordine per sei stati: `BUYER_DRAFT` [SUBMIT, EDIT, CANCEL], `BUYER_OFFER` [CHECKOUT, EDIT, CANCEL], `CANCELLED` [REQUOTE], `EXPIRED` [REQUOTE, CANCEL], `SELLER_REQUEST` e `SELLER_DRAFT` [SUBMIT, EDIT].

`OccQuoteEntryNormalizer` aggiunge `quoteCode` a ogni entry e normalizza il prodotto con `PRODUCT_NORMALIZER` (feature-libs/quote/occ/converters/occ-quote-entry-normalizer.ts:L24-L33).

3.6 Quote: `QuoteService`, command e query

Tutti i command usano `CommandStrategy.CancelPrevious` (feature-libs/quote/core/facade/quote.service.ts:L127-L129), che processa via `switchMap` e completa il notifier del comando cancellato (core-libs/core/src/util/command-query/command.service.ts:L73-L92). `Command.execute` è eager: accoda il comando e ritorna un `ReplaySubject`, quindi la richiesta parte anche se il chiamante non si sottoscrive (core-libs/core/src/util/command-query/command.service.ts:L120-L131).

Un `BehaviorSubject` `isActionPerforming$` (feature-libs/quote/core/facade/quote.service.ts:L82-L85) viene OR-ato con `loading` della query dettagli in `getQuoteDetailsQueryState`, così la UI mostra lo spinner durante le azioni; `getQuoteDetails` emette solo quote caricate e definite (feature-libs/quote/core/facade/quote.service.ts:L535-L554).

Creazione (`createQuoteCommand`, feature-libs/quote/core/facade/quote.service.ts:L90-L125): prende `userId` e l'id del carrello attivo, POST `createQuote` con `{ cartId }`, poi PATCH `editQuote` con i metadata, carica il quote cart come attivo (`multiCartFacade.loadCart` con `extraData: { active: true }`) e dispatcha `QuoteDetailsReloadQueryEvent`.

Azione (`performQuoteActionCommand`, feature-libs/quote/core/facade/quote.service.ts:L230-L298): imposta `isActionPerforming$`, POST `/action`, poi dirama:

- `SUBMIT`/`CANCEL` → `cartUtilsService.handleCartAndGoToQuoteList(isEditable)` (feature-libs/quote/core/facade/quote.service.ts:L244-L251), che se la quote era in edit crea un nuovo carrello prima di navigare a `quotes` (feature-libs/quote/core/services/cart-utils.service.ts:L52-L62);
- `REJECT`/`APPROVE` → `go({ cxRoute: 'quotes' })` (feature-libs/quote/core/facade/quote.service.ts:L252-L259);
- `EDIT`/`CHECKOUT` → `saveActiveCart()`, poi `loadQuoteCartAndProceed`; se la quote non ha ancora `cartId` la rilegge dal backend (feature-libs/quote/core/facade/quote.service.ts:L260-L292);
- errore → `triggerReloadAndCompleteAction` + `handleError` (feature-libs/quote/core/facade/quote.service.ts:L295-L298). `handleError` trasforma `CommerceQuoteExpirationTimeError` in messaggio `quote.httpHandlers.expired` + `EMPTY`, altrimenti rilancia (feature-libs/quote/core/facade/quote.service.ts:L306-L315).

`saveActiveCart` salva come saved cart il carrello attivo solo se non vuoto e non già quote cart, avvisando l'utente (feature-libs/quote/core/facade/quote.service.ts:L461-L487): entrare in una quote non distrugge il carrello normale.

`loadQuoteCartAndProceed` (feature-libs/quote/core/facade/quote.service.ts:L330-L354) carica il cart come attivo, attende che `activeCartFacade.getActive()` emetta quel codice, poi: per `CHECKOUT` chiama `quoteCartService.setCheckoutAllowed(true)` e va a `checkout`; per `REQUOTE` va a `quoteDetails` con il nuovo id.

Requote (`requoteCommand`, feature-libs/quote/core/facade/quote.service.ts:L362-L386): POST `createQuote` con `{ quoteCode }`, salva il carrello attivo e procede con il flusso `REQUOTE`.

Commenti e sconti: `addQuoteCommentCommand` sceglie commento di entry o di testata in base a `entryNumber` e non ricarica (lo fa il componente) (feature-libs/quote/core/facade/quote.service.ts:L171-L186); `addDiscountCommand` ricarica in successo e in errore (feature-libs/quote/core/facade/quote.service.ts:L203-L216). `downloadAttachment` non è un command ma una pipe `takeUserId` + `mergeMap` (feature-libs/quote/core/facade/quote.service.ts:L556-L569).

Query dettagli (feature-libs/quote/core/facade/quote.service.ts:L388-L421): attende una prima emissione del carrello attivo (serve al normalizer per sapere se la quote è collegata al quote cart), poi legge `quoteId` dai parametri router solo quando `nextState === undefined` e l'utente è loggato; si ricarica su `QuoteDetailsReloadQueryEvent` e `LoginEvent`.

Query lista (feature-libs/quote/core/facade/quote.service.ts:L423-L449): reagisce a `currentPage$`/`sort$` dentro il loader, usa `viewConfig.view.defaultPageSize`, `reloadOn: []`, `resetOn: [LoginEvent]`.

Reload da carrello: `QuoteCartEventListener` traduce `CartAddEntrySuccessEvent`, `CartUpdateEntrySuccessEvent`, `CartRemoveEntrySuccessEvent` in `QuoteDetailsReloadQueryEvent` (feature-libs/quote/core/event/quote-cart-event.listener.ts:L29-L49), l'unico evento della lib (feature-libs/quote/core/event/quote.events.ts:L9-L14): modificare il quote cart aggiorna subito i totali della quote.

3.7 Quote: quote cart e guard

`QuoteCartService` deriva "il carrello attivo è un quote cart" da `cart.quoteCode !== undefined` e tiene `checkoutAllowed` in un `ReplaySubject(1)` inizializzato a `false` (feature-libs/quote/core/services/quote-cart.service.ts:L18-L45).

`QuoteCartGuardComponent` è un **marker** con template vuoto (feature-libs/quote/components/cart-guard/quote-cart-guard.component.ts:L10-L17); il suo mapping CMS dichiara `guards: [QuoteCartGuard]` (feature-libs/quote/components/cart-guard/quote-cart-guard.component.module.ts:L16-L23). Così ogni pagina CMS che contiene il marker eredita il guard tramite `CmsPageGuard` (meccanismo dei guard di componente in 15 — CMS core). `QuoteCartGuard.canActivate` restituisce un `UrlTree` verso `quoteDetails` se il quote cart è attivo e la navigazione non è un checkout consentito (feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L34-L57). "Checkout consentito" = (prossima route semantica che inizia con `checkout` OPPURE route corrente di checkout senza prossima route) E `isCheckoutAllowed` (feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L67-L80). In quali pagine sia posizionato il marker dipende dai dati CMS del backend: `DA VERIFICARE` (non presente nel repo).

3.8 Quote: componenti principali

Request button — `QuoteRequestButtonComponent` (componente CMS `QuoteRequestComponent`, feature-libs/quote/components/request-button/quote-request-button.module.ts:L16-L22) mostra il bottone solo agli utenti loggati: non usa `AuthGuard` per non proteggere l'intera pagina carrello (feature-libs/quote/components/request-button/quote-request-button.component.ts:L32-L37). Il click chiama `quoteFacade.createQuote({})` e naviga a `quoteDetails` con il codice ottenuto (feature-libs/quote/components/request-button/quote-request-button.component.ts:L43-L57).

Summary actions — `QuoteSummaryActionsComponent`:

- filtra via `DOWNLOAD_QUOTE_PROPOSAL_DOCUMENT` (diventa un link) (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L421-L425);
- rende un bottone per azione con label `quote.summary.actions.<TYPE>` e `[disabled]` (feature-libs/quote/components/summary/actions/quote-summary-actions.component.html:L8-L14);
- disabilita `SUBMIT` (e `REQUOTE` se il toggle `showWarningMessageOnRequoteButtonClick` è attivo) quando `totalPrice.value` è sotto la soglia (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L219-L232); all'init mostra il warning `quote.commons.minRequestInitiationNote` se un'azione va disabilitata (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L99-L117);
- stile: `btn-primary` se `isPrimary`; con più di due bottoni `CANCEL` diventa `btn-tertiary` (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L330-L343);
- `onClick` esegue subito se non serve conferma, altrimenti apre il dialog e attende la chiusura (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L241-L257). Il config del dialog si cerca per stato → ruolo (derivato dal prefisso dello stato) → `ALL` (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L349-L363, feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L411-L419). Il dialog è `LAUNCH_CALLER.QUOTE_ACTION_CONFIRMATION`; l'azione parte solo se il reason di chiusura è `'yes'` (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L267-L299). `REQUOTE` passa da `quoteFacade.requote`, tutte le altre da `performQuoteAction` (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L259-L265).

Il layout del dialog è `inline`, `DIALOG_TYPE.DIALOG`, con `QuoteConfirmDialogComponent` (feature-libs/quote/components/confirm-dialog/default-quote-confirm-dialog.config.ts:L10-L18), che legge il context da `launchDialogService.data$` e chiude con un reason (feature-libs/quote/components/confirm-dialog/quote-

confirm-dialog.component.ts:L60-L74). Le chiavi i18n del dialog sono costruite dal prefisso configurato: .title, .confirmNote, .a11y.close, opzionali .warningNote e .successMessage (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L365-L390).

Seller edit — attivo solo se stato SELLER_REQUEST/SELLER_DRAFT e isEditable (feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.service.ts:L133-L140). Lo sconto viene sempre inviato come PERCENT; la scadenza passa da un Subject con debounceTime(updateDebounceTime.expiryDate ?? 500) prima di editQuote (feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.ts:L129-L184). La data viene composta con ora locale e offset di fuso (feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.service.ts:L108-L115). La validazione rifiuta >100 o troppi decimali (feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.service.ts:L234-L256).

Header — il buyer modifica nome/descrizione solo se isEditable e stato BUYER_DRAFT/BUYER_OFFER (feature-libs/quote/components/header/overview/quote-header-overview.component.ts:L89-L95); la label totale è "Total" se tra le azioni c'è CHECKOUT, altrimenti "Estimated total" (feature-libs/quote/components/header/overview/quote-header-overview.component.ts:L304-L311).

Items — QuoteItemsComponentService.retrieveQuoteEntries sceglie la fonte: le entry della quote (nessun cartId), il carrello read-only (con cartId, non editabile) o il carrello attivo editabile (feature-libs/quote/components/items/quote-items.component.service.ts:L80-L91); nel secondo caso il cart viene caricato con extraData: { active: false } (feature-libs/quote/components/items/quote-items.component.service.ts:L130-L138). Il rendering riusa l'outlet CART_ITEM_LIST di cart base con readonly nel contesto (feature-libs/quote/components/items/quote-items.component.html:L27-L30).

Lista — QuoteListModule mappa AccountMyQuotesComponent e fornisce view.defaultPageSize: 5 come default config (feature-libs/quote/components/list/quote-list.module.ts:L35-L49); sort di default byCode con opzioni hard-coded sostituite da data.sorts quando presenti (feature-libs/quote/components/list/quote-list-component.service.ts:L17-L44).

Link e commenti — QuoteLinksComponent forza il reload all'init perché la pagina dettagli è in cache (feature-libs/quote/components/links/quote-links.component.ts:L47-L50); scarica il primo allegato come <filename>.pdf (feature-libs/quote/components/links/quote-links.component.ts:L68-L83); mostra il link all'ordine solo con orderConfig.showOrderQuoteLink e sapOrderCode (feature-libs/quote/components/links/quote-links.component.html:L38-L46). QuoteCommentsComponent.onSend invia, resetta il form comunque e ricarica in successo (feature-libs/quote/components/comments/quote-comments.component.ts:L62-L86).

3.9 Quote: OCC

Dieci endpoint di default; getQuote chiede FULL,expirationTime, i prezzi formattati e entries(FULL) (feature-libs/quote/occ/config/default-occ-quote-config.ts:L9-L31), fusi in OccEndpoints via module augmentation (feature-libs/quote/occ/model/occ-quote-endpoints.model.ts:L61-L63). Con orderConfig.showOrderQuoteLink (default false, feature-libs/order/root/config/default-order-config.ts:L27-L27) l'adapter fonde i field di getQuote e getOrderCode con OccFieldsService.getOptimalUrlGroups (feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L107-L124). editQuote è PATCH, performQuoteAction POST con { action } (feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L136-L167); il download usa responseType: 'blob' as 'json' (feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L253-L264); tutte le chiamate normalizzano gli errori con tryNormalizeHttpError (feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L276-L284).

3.10 Customer ticketing

Root. Sei componenti CMS sulla feature `customerTicketing`; `customerTicketingCore` è un alias della stessa feature ("by default core is bundled together with components") (feature-libs/customer-ticketing/root/customer-ticketing-root.module.ts:L24-L43). Il root importa `CustomerTicketingEventModule` e definisce due ng-route `path: null` per `supportTicketDetails` e `supportTickets` (feature-libs/customer-ticketing/root/customer-ticketing-root.module.ts:L44-L74), mappate su `my-account/support-tickets` e `my-account/support-ticket/:ticketCode` (feature-libs/customer-ticketing/root/config/default-customer-ticketing-routing-config.ts:L9-L21). Nell'app demo il chunk lazy è un wrapper che aggiunge `CdpCustomerTicketingModule` se `environment.cdp` (projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts:L13-L22).

Modello. Stati `OPEN`, `CLOSED`, `INPROCESS` (feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts:L14-L24); `TicketDetails` porta `availableStatusTransitions` e `ticketEvents`, `TicketEvent` porta `toStatus`, `addedByAgent`, `ticketEventAttachments` (feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts:L26-L69). Tre `LAUNCH_CALLER` aggiunti a runtime (feature-libs/customer-ticketing/root/model/augmented-core.model.ts:L9-L21).

Facade e servizio. `CustomerTicketingFacade` è lazy con 12 metodi (feature-libs/customer-ticketing/root/facade/customer-ticketing.facade.ts:L19-L40). I command di `CustomerTicketingService` usano `CommandStrategy.Queue` (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L100-L102), cioè `concatMap`: i comandi si accodano (core-libs/core/src/util/command-query/command.service.ts:L105-L116). Le precondizioni prendono `userId` e `ticketCode` dalla route con `take(1)` e lanciano un errore se manca l'utente (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L254-L270).

Orchestrazione a eventi. Il servizio non ricarica direttamente: dispatcha eventi di dominio e un listener eager li traduce.

- `createTicketCommand` → POST → `TicketCreatedEvent` (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L86-L103).
- `createTicketEventCommand`: `toStatus` CLOSED → `TicketClosedEvent`; altrimenti, se non c'è allegato, OPEN/INPROCESS → `TicketReopenedEvent`, altro → `NewMessageEvent` (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L118-L130).
- `uploadAttachmentCommand` usa `payload.ticketId ?? ticketId` e dispatcha `UploadAttachmentSuccessEvent` (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L151-L163).
- Listener: `TicketCreatedEvent` → messaggio + reload lista (feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L50-L62); lingua/valuta → reload lista e dettaglio; login/logout → reset delle quattro query (feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L64-L91); Closed → reset ticket + `requestClosed`; Reopened → reload + `requestReopened`; NewMessage/UploadAttachmentSuccess → reload ticket (feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L93-L135).

Error handler e SEO. Il core lazy registra `NotFoundTicketRequestHandler` e `CustomerTicketingPageMetaResolver` (feature-libs/customer-ticketing/core/customer-ticketing-core.module.ts:L14-L31). L'handler fa match solo per 404 sulla route `supportTicketDetails` con errori notFound e reindirizza a `supportTickets` (feature-libs/customer-ticketing/core/http-interceptors/handlers/not-found-ticket-request.handler.ts:L38-L70); `isNotFoundError` richiede sia `reason` sia `type` NOT_FOUND (feature-libs/customer-ticketing/core/utils/utils.ts:L16-L21). Il resolver (pageUid `support-ticket-details`) usa il `subject` come heading e aggiunge un breadcrumb "customer service" (feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts:L53-L101). Per i page meta resolver in generale vedi 40 — SEO e PWA.

OCC. `getTicket` inverte in place `ticketEvents` prima della normalizzazione (`feature-libs/customer-ticketing/occ/adapters/occ-customer-ticketing.adapter.ts:L92-L101`); l'upload è un `FormData` con campo `ticketEventAttachment` (`feature-libs/customer-ticketing/occ/adapters/occ-customer-ticketing.adapter.ts:L212-L219`); associated objects e categorie vengono estratti dai wrapper e normalizzati con `pipeableMany` (`feature-libs/customer-ticketing/occ/adapters/occ-customer-ticketing.adapter.ts:L58-L85`). Alcuni endpoint hanno lo slash iniziale (`/ticketCategories`, `/users/...`) (`feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L13-L23`): è innocuo perché `buildUrl` compone con `urlPathJoin`, che ripulisce gli slash di ogni parte (`core-libs/core/src/occ/services/occ-endpoints.service.ts:L233-L238`, `core-libs/core/src/occ/utils/occ-url-util.ts:L12-L33`).

Componenti. I tre dialog (REOPEN, CLOSE, CREATE) sono `inline` di tipo DIALOG (`feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/default-customer-ticketing-form-layout.config.ts:L12-L32`) e derivano da `CustomerTicketingDialogComponent`, una `@Directive()` astratta che chiude al click sull'host e costruisce il form con validatori dell'allegato (`feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/customer-ticketing-dialog.component.ts:L82-L118`). "Close" è abilitato con stato OPEN/INPROCESS e transizione verso CLOSED (`feature-libs/customer-ticketing/components/details/customer-ticketing-close/customer-ticketing-close-component.service.ts:L17-L30`); "Reopen" con stato CLOSED e transizione verso INPROCESS/OPEN (`feature-libs/customer-ticketing/components/details/customer-ticketing-reopen/customer-ticketing-reopen-component.service.ts:L17-L31`). Il reopen invia `toStatus INPROCESS` e, se c'è un file, passa `containsAttachment=true` e carica l'allegato sull'evento creato (`feature-libs/customer-ticketing/components/details/customer-ticketing-reopen/customer-ticketing-reopen-dialog/customer-ticketing-reopen-dialog.component.ts:L58-L98`). Il create dialog carica l'allegato sul primo evento del ticket, passando `response.id` esplicitamente (`feature-libs/customer-ticketing/components/list/customer-ticketing-create/customer-ticketing-create-dialog/customer-ticketing-create-dialog.component.ts:L161-L172`). Il dettaglio dispatcha `GetTicketQueryReloadEvent` se il ticket in cache non corrisponde al `ticketCode` della route (`feature-libs/customer-ticketing/components/details/customer-ticketing-details/customer-ticketing-details.component.ts:L83-L97`).

3.11 Subscription billing

Root eager con outlet. Cinque componenti CMS sulla feature `subscriptionBilling` e alias `subscriptionBillingCore` (`feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L24-L40`). Il root fornisce inoltre due outlet del carrello in posizione `AFTER`: `SUBSCRIPTION_PRICE_HEADING` e `SUBSCRIPTION_PRICE_BODY` (`feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L42-L59`), i cui id sono definiti in cart base (`feature-libs/cart/base/root/models/cart-outlets.model.ts:L28-L29`) e resi dalla cart item list (`feature-libs/cart/base/components/cart-shared/cart-item-list/cart-item-list.component.html:L14-L14`). I componenti outlet sono standalone e leggono il contesto con `toSignal(OutletContextData.context$)` (`feature-libs/subscription-billing/root/components/cart/price-heading/subscription-cart-price-heading.component.ts:L22-L31`). I componenti outlet esistono quindi anche senza il chunk lazy. Il prezzo compare però solo se le entry del carrello contengono `sapSubscriptionTerm` e `sapPricePlan` (`feature-libs/subscription-billing/root/services/subscription-product.service.ts:L18-L20`), che i field di default dell'endpoint `cart` non richiedono esplicitamente (`feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L17-L17`); la config OCC di subscription-billing arriva solo dal modulo OCC lazy (`feature-libs/subscription-billing/occ/subscription-billing-occ.module.ts:L23-L23`). `DA VERIFICARE`: dipende dal livello `DEFAULT` del backend.

Prodotti subscription. Un prodotto è subscription se ha sia `sapSubscriptionTerm` sia `sapPricePlan`; i dati si caricano con lo scope `SUBSCRIPTION` (`feature-libs/subscription-billing/root/services/subscription-product.service.ts:L18-L28`). Il model augmenta `Product` e aggiunge `ProductScope.SUBSCRIPTION = 'subscription'` anche a runtime (`feature-libs/subscription-billing/root/model/subscription-product.`

model.ts:L98-L110); l'endpoint product.subscription chiede fields=sapPricePlan,sapSubscriptionTerm (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L10-L43). Gli scope prodotto sono trattati in 24 — Product e search.

Facade. Tre facade lazy: SubscriptionFacade, SubscriptionBillingFacade, SubscriptionActionsFacade (6 metodi) (feature-libs/subscription-billing/root/facade/subscription-actions.facade.ts:L17-L32). Il core registra tre coppie facade/servizio e due PageMetaResolver (feature-libs/subscription-billing/core/facade/facade-provider.ts:L20-L46).

Codice dall'URL. Il codice della subscription è estratto con la regex /subscription/<id> dall'URL, non dai parametri di route (feature-libs/subscription-billing/core/facade/subscription.service.ts:L126-L134); idem per i bill con /subscription-bill/<id> (feature-libs/subscription-billing/core/facade/subscription-billing.service.ts:L157-L165). Le precondizioni del dettaglio non hanno take(1) e lanciano se manca codice o utente (feature-libs/subscription-billing/core/facade/subscription.service.ts:L111-L123); la query ha solo reloadOn (il resetOn è commentato) (feature-libs/subscription-billing/core/facade/subscription.service.ts:L53-L56).

Azioni senza Command. SubscriptionActionsService restituisce Observable freddi (combineLatest + take(1) + switchMap) (feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L55-L77): senza subscribe non parte nulla. getExtensionEffectiveDate è l'unico senza take(1) (feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L123-L138). Né lui né extendSubscription controllano userId/codice (feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L145-L157); gli altri quattro metodi lanciano un errore se mancano (feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L35-L121).

OCC. Gli adapter non usano converter: restituiscono il JSON tipizzato normalizzando solo gli errori (feature-libs/subscription-billing/occ/adapters/occ-subscription.adapter.ts:L36-L40). L'estensione invia { numberOfBillingCycles, unlimited } (feature-libs/subscription-billing/occ/adapters/occ-subscription-actions.adapter.ts:L120-L153); reverseCancellation fa POST con body null e Content-Type JSON (feature-libs/subscription-billing/occ/adapters/occ-subscription-actions.adapter.ts:L98-L101); il filtro bill va nel query param filters (feature-libs/subscription-billing/occ/adapters/occ-subscription-billing.adapter.ts:L37-L47).

Componenti. SubscriptionDetailsComponent forza il reload all'init e apre il dialog con openDialogAndSubscribe(SUBSCRIPTION_ACTION_CONFIRMATION, ...) passando subscription, code e mode (feature-libs/subscription-billing/components/details/subscription-details.component.ts:L51-L73); il dialog è inlineRoot (feature-libs/subscription-billing/components/actions-modal/subscription-actions-popup.config.ts:L12-L20). SubscriptionActionsModalComponent usa signals (mode default 'cancel', feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L67-L92), un effect che in modalità cancel richiede subito la data effettiva (feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L120-L140) e una mappa di handler per mode in onConfirm (feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L174-L254). Successo: close + GetSubscriptionByCodeReloadEvent + conferma; errore: close('error') + subscriptionActions.unknownError (feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal-component.service.ts:L23-L59). La lista bill riceve il range date dal dialogClose del DateRangeModalComponent (feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L77-L90, feature-libs/subscription-billing/components/list/billing/date-range-modal/date-range-modal.component.ts:L85-L94) e lo serializza come startAt:%s:endAt:%s (feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L64-L64).

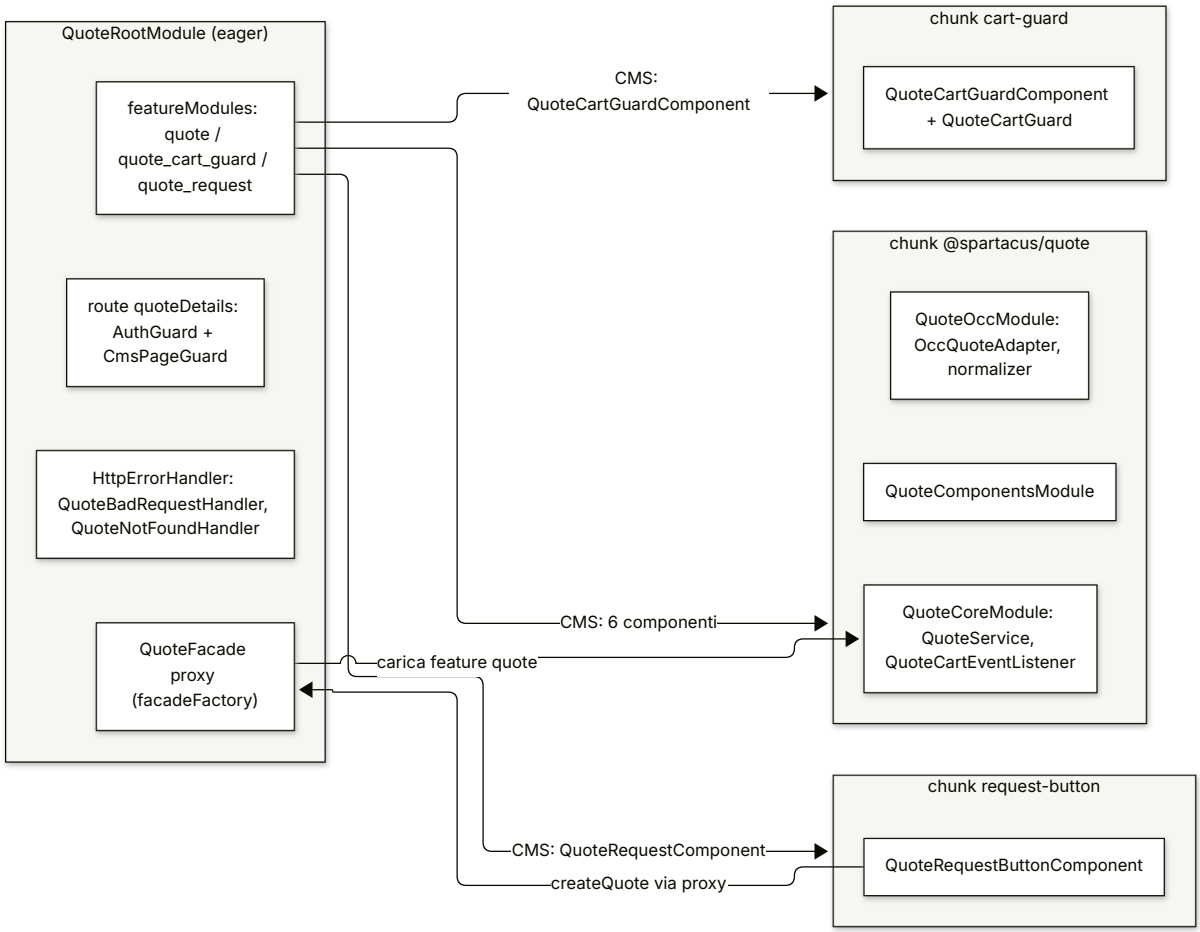
3.12 Installazione via schematics, costanti e stili

`addQuoteFeatures()` valida l'installazione, risolve le dipendenze tra feature ed esegue in catena `analyzeApplication`, `addFeatures`, `addPackageJsonDependenciesForLibrary(peerDependencies, ...)` e `finalizeInstallation` (feature-libs/quote/schematics/add-quote/index.ts:L25-L43), registrata come `ng-add` e `add` (feature-libs/quote/schematics/collection.json:L4-L16). Lo spec controlla con snapshot che il modulo di configurazione riceva le feature b2b (feature-libs/quote/schematics/add-quote/index_spec.ts:L115-L122), coerente con la config schematics della quote. Le costanti di feature-libs/quote/schematics/constants.ts:L9-L17 replicano nomi già definiti in core-libs (core-libs/schematics/src/shared/lib-configs/quote-schematics-config.ts:L21-L33, core-libs/schematics/src/shared/libs-constants.ts:L237-L238), e nessun file di `feature-libs/quote` le importa: la fonte effettiva è quella di core-libs.

Negli stili quote, il page template usa variabili chiamate `$page-template-allowlist-store-finder` / `$page-template-blocklist-store-finder` pur riferendosi a `QuoteDetailsPageTemplate` (feature-libs/quote/styles/layout/_index.scss:L5-L17): per escludere il template va sovrascritta la variabile con suffisso `-store-finder`. Il mixin `visible-focus` è definito localmente nell'entry SCSS prima di importare gli stili (feature-libs/quote/_index.scss:L4-L12). Le tre `vitest.config.ts` hanno la stessa struttura (jsdom, pool `forks`, coverage v8 con soglie 85/85/65/85) e differiscono solo negli alias (feature-libs/quote/vitest.config.ts:L16-L21, feature-libs/customer-ticketing/vitest.config.ts:L16-L20, feature-libs/subscription-billing/vitest.config.ts:L11-L23).

4. Diagrammi

4.1 Quote: cosa è eager e cosa è lazy



Riferimenti: `feature-libs/quote/root/quote-root.module.ts:L29-L64`, `feature-libs/quote/root/quote-root.module.ts:L88-L118`, `feature-libs/quote/root/facade/quote.facade.ts:L21-L38`, `projects/storefrontapp/src/app/spartacus/features/quote-feature.module.ts:L26-L63`, `feature-libs/quote/core/quote-core.module.ts:L14-L25`.

4.2 Quote: dalla risposta OCC ai bottoni

■ **TAVOLA 31.1** diagramma a tutta pagina, in fondo al capitolo

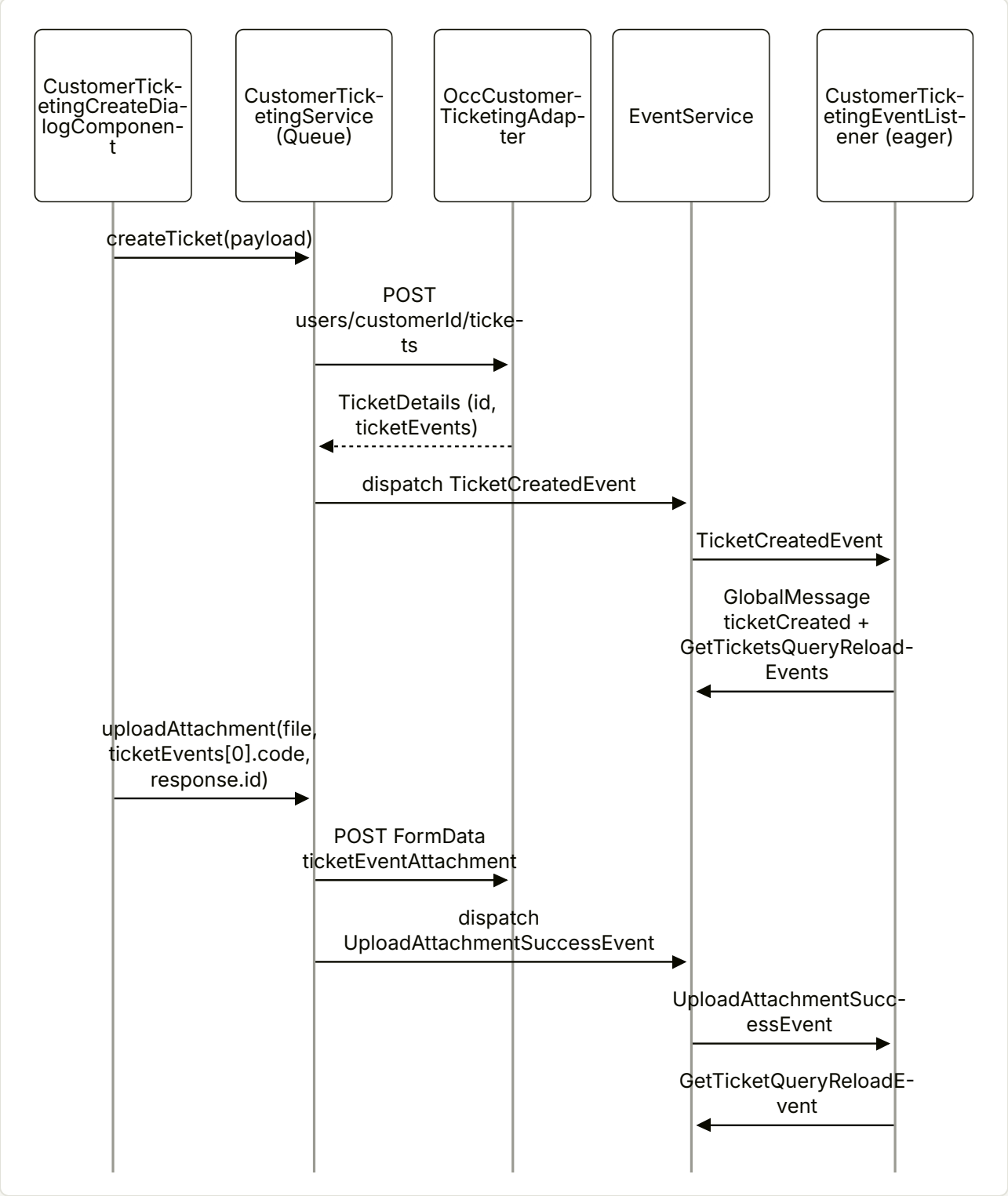
Riferimenti: `feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L30-L89`, `feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L241-L299`, `feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L349-L363`, `feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L421-L425`.

4.3 Quote: CHECKOUT di una quote in BUYER_OFFER

■ **TAVOLA 31.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/quote/components/config/default-quote-ui.config.ts:L22-L102, feature-libs/quote/core/facade/quote.service.ts:L230-L292, feature-libs/quote/core/facade/quote.service.ts:L330-L354, feature-libs/quote/core/facade/quote.service.ts:L461-L487, feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L34-L80.

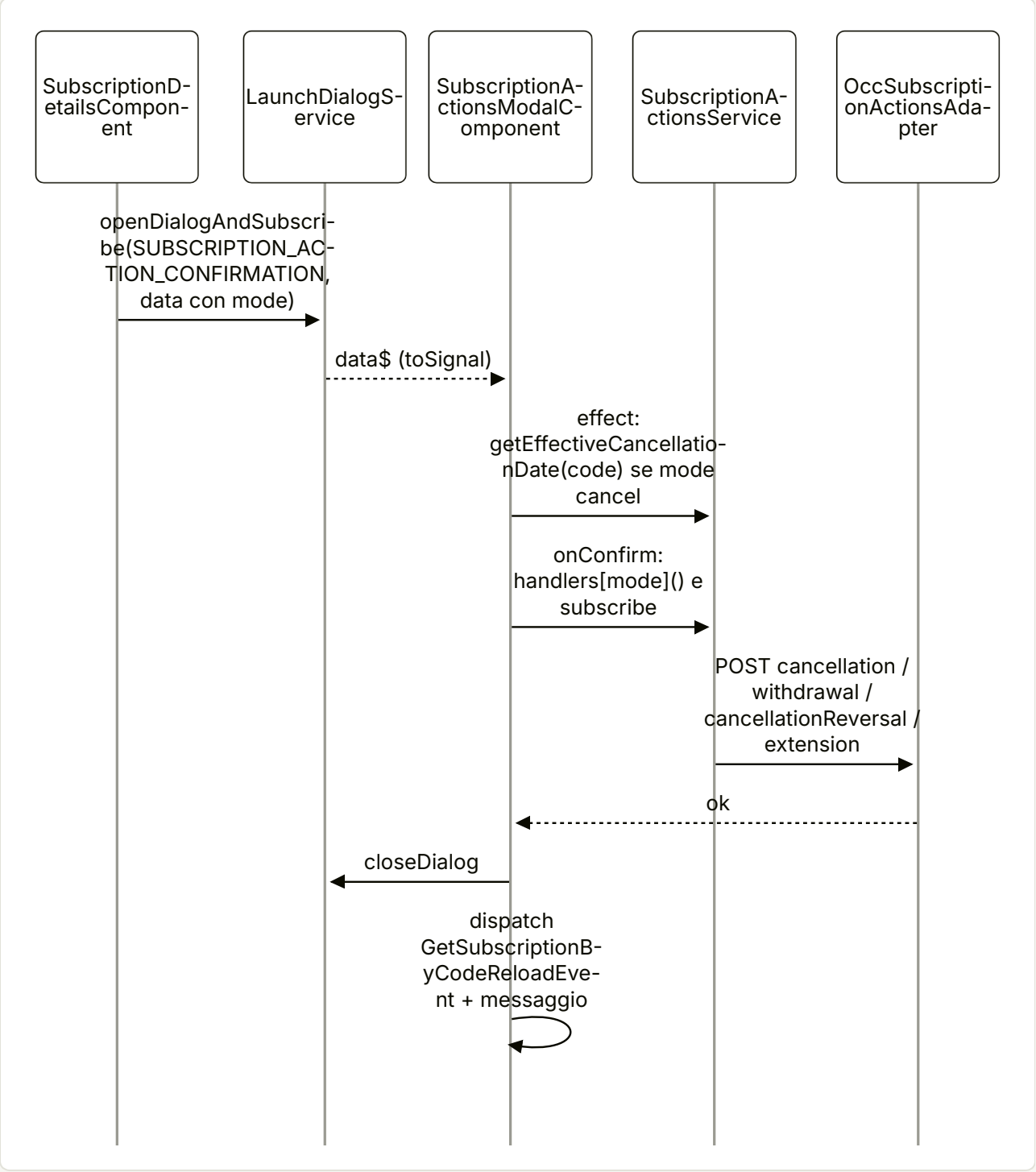
4.4 Customer ticketing: creazione con allegato



Riferimenti: `feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L86-L103`, `feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L151-L163`, `feature-libs/customer-ticketing/components/list/customer-ticketing-create/customer-ticketing-create-dialog/customer-ticketing-create-dialog.component.ts:L161-L172`, `feature-libs/customer-ticketing/occ/adapters/occ-customer-ticketing.adapter.ts:L212-L219`, `feature-libs/customer-ticketing/root/events/`

customer-ticketing-event.listener.ts:L50-L62, feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L93-L135.

4.5 Subscription billing: azione sul dettaglio



Riferimenti: feature-libs/subscription-billing/components/details/subscription-details.component.ts:L51-L73, feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L67-L92, feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L120-L140, feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L174-L254, feature-libs/subscription-billing/core/facade/

subscription-actions.service.ts:L55-L77, feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal-component.service.ts:L23-L59.

---

5. Estratti di codice reale

5.1 Facade proxy con metodi mancanti

feature-libs/quote/root/facade/quote.facade.ts:L21-L38

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: QuoteFacade,
      feature: QUOTE_FEATURE,
      methods: [
        'getQuotesState',
        'getQuoteDetailsQueryState',
        'getQuoteDetails',
        'createQuote',
        'editQuote',
        'performQuoteAction',
        'addQuoteComment',
        'addDiscount',
      ],
    }),
})
```

core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L119-L123

```ts
    const result: any = new (class extends (facade as any) {})();
    (methods ?? []).forEach((method) => {
      result[method] = (...args: any[]) =>
        this.call(resolver$, method as string, args);
    });
```

5.2 Normalizer delle azioni

feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L30-L44

```ts
    if (source.allowedActions && source.state) {
      target.allowedActions = this.getOrderedActions(
        source.state,
        source.allowedActions,
        source.code
      ).map((action) => this.getActionCategory(action));
```

```
}
const switchToEditModeRequired = target.allowedActions?.find(
  (quoteAction) => quoteAction.type === QuoteActionType.EDIT
);

target.isEditable =
  !!source.allowedActions?.includes(QuoteActionType.EDIT) &&
  !switchToEditModeRequired;
```

feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L60-L89

```ts
const order =
  this.quoteCoreConfig.quote?.actions?.actionsOrderByState?.[state];
if (order) {
  //deep copy order list
  const clonedActionList = structuredClone(order);
  combineLatest([
    this.quoteCartService.isQuoteCartActive(),
    this.quoteCartService.getQuoteId(),
  ])
    .pipe(take(1))
    .subscribe(([isQuoteCartActive, cartQuoteId]) => {
      if (
        state !== QuoteState.BUYER_OFFER &&
        isQuoteCartActive &&
        cartQuoteId === quoteId
      ) {
        const editIndex = clonedActionList.indexOf(QuoteActionType.EDIT);
        if (editIndex > -1) {
          clonedActionList.splice(editIndex, 1);
        }
      }
    });
  return list
    .filter((item) => clonedActionList.includes(item))
    .sort(
      (a, b) => clonedActionList.indexOf(a) - clonedActionList.indexOf(b)
    );
} else {
  return list;
}
```

5.3 Diramazione delle azioni in `QuoteService`

feature-libs/quote/core/facade/quote.service.ts:L245-L266

```ts
      if (
        payload.quoteAction === QuoteActionType.SUBMIT ||
        payload.quoteAction === QuoteActionType.CANCEL
      ) {
        this.cartUtilsService.handleCartAndGoToQuoteList(
          payload.quote.isEditable
        );
        this.isActionPerforming$.next(false);
      }
      if (
        payload.quoteAction === QuoteActionType.REJECT ||
        payload.quoteAction === QuoteActionType.APPROVE
      ) {
        this.routingService.go({ cxRoute: 'quotes' });
      }
      if (
        payload.quoteAction === QuoteActionType.EDIT ||
        payload.quoteAction === QuoteActionType.CHECKOUT
      ) {
        this.saveActiveCart();
        //no cartId present: ensure that we re-fetch quote cart id from quote
        const cartId = payload.quote.cartId;
```

feature-libs/quote/core/facade/quote.service.ts:L337-L354

```ts
  this.activeCartFacade
    .getActive()
    .pipe(
      filter((cart) => cart.code === cartId),
      take(1)
    )
    .subscribe(() => {
      this.triggerReloadAndCompleteAction();
      if (actionType === QuoteActionType.CHECKOUT) {
        this.quoteCartService.setCheckoutAllowed(true);
        this.routingService.go({ cxRoute: 'checkout' });
      } else if (actionType === QuoteActionType.REQUOTE) {
        this.routingService.go({
          cxRoute: 'quoteDetails',
          params: { quoteId: quoteId },
        });
      }
    });
```

5.4 Query dei dettagli guidata dal router

feature-libs/quote/core/facade/quote.service.ts:L396-L420

```ts
            combineLatest([
              this.routingService.getRouterState(),
              this.authService.isUserLoggedIn(),
            ]).pipe(
              //we don't need to cover the intermediate router states where a future route is already known.
              //only changes to the URL are relevant. Otherwise we get unneeded hits when e.g. navigating back from quotes.
              //In addition we check is the user is logged in as otherwise we would get (failing) calls after logout
              filter(
                ([routerState, isLoggedIn]) =>
                  routerState.nextState === undefined && isLoggedIn
              ),
              withLatestFrom(this.userIdService.takeUserId()),
              switchMap((([routerState, _isLoggedIn], userId]) => {
                return this.quoteConnector.getQuote(
                  userId,
                  routerState.state.params.quoteId
                );
              })
            )
          )
        ),

      {
        reloadOn: [QuoteDetailsReloadQueryEvent, LoginEvent],
      }
```

5.5 Guard del quote cart

feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L67-L80

```ts
  protected checkAllowedCheckoutNavigation(
    routerState: RouterState,
    isCheckoutAllowed: boolean
  ) {
    const nextStateIsCheckout =
      routerState.nextState?.semanticRoute?.startsWith('checkout');
    const currentStateIsCheckout =
      routerState.state.semanticRoute?.startsWith('checkout');
    const noNextState = routerState.nextState === undefined;
    return (
      (nextStateIsCheckout || (currentStateIsCheckout && noNextState)) &&
      isCheckoutAllowed
    );
  }
```

5.6 Ricerca del dialog di conferma

feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L349-L363

```ts
protected isConfirmationDialogRequired(
  action: QuoteActionType,
  state: QuoteState,
  cartIsEmptyOrQuoteCart: boolean
): boolean {
  const mappingConfig = this.quoteUIConfig.quote?.confirmActionDialogMapping;
  const dialogConfig =
    mappingConfig?.[state]?.[action] ??
    mappingConfig?.[this.stateToRoleTypeForDialogConfig(state)]?.[action] ??
    mappingConfig?.[QuoteRoleType.ALL]?.[action];
  return (
    !!dialogConfig &&
    (!cartIsEmptyOrQuoteCart || !dialogConfig.showOnlyWhenCartIsNotEmpty)
  );
}
```

5.7 Ticketing: evento scelto in base a `toStatus`

feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L115-L132

```ts
        this.customerTicketingConnector
          .createTicketEvent(customerId, ticketId, payload.ticketEvent)
          .pipe(
            tap(() => {
              if (payload.ticketEvent.toStatus?.id === STATUS.CLOSED) {
                this.eventService.dispatch({}, TicketClosedEvent);
              } else if (!payload.containsAttachment) {
                if (
                  payload.ticketEvent.toStatus?.id === STATUS.OPEN ||
                  payload.ticketEvent.toStatus?.id === STATUS.INPROCESS
                ) {
                  this.eventService.dispatch({}, TicketReopenedEvent);
                } else {
                  this.eventService.dispatch({}, NewMessageEvent);
                }
              }
            })
          )
```

5.8 Subscription: outlet eager e azioni fredde

feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L47-L56

```ts
    provideOutlet({
      id: CartOutlets.SUBSCRIPTION_PRICE_HEADING,
      position: OutletPosition.AFTER,
      component: SubscriptionCartPriceHeadingComponent,
    }),
    provideOutlet({
      id: CartOutlets.SUBSCRIPTION_PRICE_BODY,
      position: OutletPosition.AFTER,
      component: SubscriptionCartPriceBodyComponent,
    }),
```

feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L55-L77

```ts
  cancelSubscription(
    cancellationDetails: SubscriptionCancellationDetails,
    subscriptionCode?: string
  ): Observable<unknown> {
    return combineLatest([
      this.userIdService.getUserId(),
      of(subscriptionCode),
    ]).pipe(
      take(1),
      switchMap(([userId, code]) => {
        if (!userId || !code) {
          throw new Error(
            'Cannot cancel subscription: missing user ID or subscription code.'
          );
        }
        return this.subscriptionActionsConnector.cancelSubscription(
          userId,
          code,
          cancellationDetails
        );
      })
    );
  }
```

6. Configurazione

6.1 Quote

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
quote.actions.primaryActions	QuoteActionType[]	APPROVE, CHECKOUT, SUBMIT, REQUOTE	feature-libs/quote/core/config/default-quote.core.config.ts:L13-L18	feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L49-L50
quote.actions.actionsOrderByState	Partial<QuoteActionsByState>	6 stati (v. §3.5)	feature-libs/quote/core/config/default-quote.core.config.ts:L19-L34	feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L60-L61
quote.maxCharsForComments	number	1000	feature-libs/quote/components/config/default-quote-ui.config.ts:L106-L106	feature-libs/quote/components/comments/quote-comments.component.ts:L131-L133
quote.truncateCardTileContentAfterNumChars	number	100	feature-libs/quote/components/config/default-quote-ui.config.ts:L107-L107	feature-libs/quote/components/header/overview/quote-header-overview.component.ts:L279-L284
quote.updateDebounceTime.expiryDate	number (ms)	500	feature-libs/quote/components/config/default-quote-ui.config.ts:L108-L110	feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.ts:L133-L135
quote.confirmActionDialogMapping	ConfirmActionDialogMappingConfig	defaultDialogMappings	feature-libs/quote/components/config/default-quote-ui.config.ts:L22-L102	feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L354-L358

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`quote.maximumDecimalsForPercentageDiscount`	number	8	`feature-libs/quote/components/config/default-quote-ui.config.ts:L112-L112`	`feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.service.ts:L227-L232`
`quote.showSubtotalBeforeDiscounts`	boolean	true	`feature-libs/quote/components/config/default-quote-ui.config.ts:L113-L113`	`feature-libs/quote/components/summary/prices/quote-summary-prices.component.ts:L24-L26`
`view.defaultPageSize`	number	5 (default di `QuoteListModule`)	`feature-libs/quote/components/list/quote-list.module.ts:L36-L40`	`feature-libs/quote/core/facade/quote.service.ts:L438-L438`
`showOrderQuoteLink` (`OrderConfig`)	boolean	false	`feature-libs/order/root/config/default-order-config.ts:L27-L27`	`feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L108-L108`, `feature-libs/quote/components/links/quote-links.component.html:L38-L38`
feature toggle `showWarningMessageOnRequoteButtonClick`	boolean	`false` (core); `true` nell'app demo	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1110-L1110`	`feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L220-L220`
`backend.occ.endpoints.getQuote` ecc.	string	v. §3.9	`feature-libs/quote/occ/config/default-occ-quote-config.ts:L12-L34`	`feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L75-L78`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes.quotes / quoteDetails	RoutingConfig	my-account/quotes, my-account/quote/:quoteId	feature-libs/quote/root/quote-root.module.ts:L66-L78	navigazioni cxRoute (es. feature-libs/quote/core/facade/quote.service.ts:L252-L259)

QuoteConfig (in root) è un'interfaccia vuota agganciata a Config.quote (feature-libs/quote/root/config/quote.config.ts:L9-L15); le chiavi concrete arrivano da QuoteCoreConfig (feature-libs/quote/core/config/quote-core.config.ts:L11-L29) e QuoteUIConfig (feature-libs/quote/components/config/quote-ui.config.ts:L15-L24). Il mapping dei dialog è indicizzato per stato **o** per ruolo; vince sempre quello per stato (feature-libs/quote/components/config/quote-ui.config.ts:L34-L45).

Nota sui default forniti dai chunk lazy (es. view.defaultPageSize o defaultQuoteCoreConfig): ConfigurationService raccoglie via UnifiedInjector i DefaultConfigChunk/ConfigChunk dei moduli caricati, li fonde con priorità default < ambient default < ambient < root e, salvo il toggle disableConfigUpdates, li fonde anche nel Config globale (core-libs/core/src/config/services/configuration.service.ts:L76-L106). Una provideConfig dell'app vince quindi sempre sui default lazy. Dettagli in 11 — Configurazione.

6.2 Customer ticketing

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
customerTicketing.attachmentRestrictions.maxSize	number (MB)	10	feature-libs/customer-ticketing/root/config/default-customer-ticketing-config.ts:L18-L18	feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/customer-ticketing-dialog.component.ts:L75-L80
customerTicketing.attachmentRestrictions.allowedTypes	string[]	11 estensioni (pdf, txt, png, jpg, jpeg, gif, doc, docx, html, htm, zip)	feature-libs/customer-ticketing/root/config/default-customer-ticketing-config.ts:L19-L31	feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/customer-ticketing-dialog.component.ts:L56-L59
customerTicketing.inputCharactersLimit	number	5000	feature-libs/customer-ticketing/root/config/default-customer-ticketing-config.ts:L33-L33	feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/customer-ticketing-dialog.component.ts:L61-L66, feature-libs/customer-ticketing/components/details/customer-ticketing-messages/customer-ticketing-messages.component.ts:L110-L111
customerTicketing.inputCharactersLimitForSubject	number	255	feature-libs/customer-ticketing/root/config/default-customer-ticketing-config.ts:L34-L34	feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/customer-ticketing-dialog.component.ts:L68-L73
customerTicketing.listViewPageSize	number	5	feature-libs/customer-ticketing/root/config/default-customer-ticketing-config.ts:L35-L35	feature-libs/customer-ticketing/components/list/customer-ticketing-list/customer-ticketing-list.component.ts:L62-L63

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`customerTicketing.agentSessionTimer.startingDelayInSeconds`	number	nessuno	`feature-libs/customer-ticketing/root/config/customer-ticketing-config.ts:L16-L18`	mai letta nella lib
`backend.occ.endpoints.getTicket` ecc.	string	v. §3.10	`feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L9-L27`	`feature-libs/customer-ticketing/occ/adapters/occ-customer-ticketing.adapter.ts:L104-L111`
`routing.routes.supportTickets` / `supportTicketDetails`	`RoutingConfig`	`my-account/support-tickets`, `my-account/support-ticket/:ticketCode`	`feature-libs/customer-ticketing/root/config/default-customer-ticketing-routing-config.ts:L9-L21`	handler e dialog

`CustomerTicketingConfig` è una classe astratta con `useExisting: Config` (`feature-libs/customer-ticketing/root/config/customer-ticketing-config.ts:L10-L27`); le costanti di fallback sono in `feature-libs/customer-ticketing/root/customer-ticketing-constants.ts:L7-L12`.

6.3 Subscription billing

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`Months` / `Years` / `Days` / `Weeks` (top-level `Config`)	number	12 / 5 / 30 / 4 (costante)	`feature-libs/subscription-billing/root/config/default-extend-subscription-dropdown-options-config.ts:L9-L15`	costante importata direttamente: `feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L84-L85`
`backend.occ.endpoints.product.subscription`	string	`products/${productCode}?fields=sapPricePlan,sapSubscriptionTerm`	`feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L36-L39`	`ProductService.get(..., [ProductScope.SUBSCRIPTION])` in `feature-libs/subscription-billing/root/services/subscription-product.service.ts:L26-L27`
`backend.occ.endpoints.subscription*`	string	10 endpoint	`feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L10-L29`	adapter OCC
`routing.routes.subscriptions` ecc.	`RoutingConfig`	`my-account/subscriptions`, …	`feature-libs/subscription-billing/root/config/default-subscription-billing-routing-config.ts:L11-L26`	componenti

Nell'app demo la feature subscription carica solo l'inglese con `fallbackLang: 'en'` (`projects/storefrontapp/src/app/spartacus/features/subscription-billing/subscription-billing-feature.module.ts:L30-L36`).

7. Commenti e note notevoli dal sorgente

- **Perché** `expirationTime` **è una stringa**: "we cannot use date here, as this would result in a format 2023-08-26T07:13:00.000Z sent to commerce while it expects (for the same time) 2023-08-26T09:13:00+02:00" (`feature-libs/quote/root/model/quote.model.ts:L120-L128`).
- **Filtro del router state** nella query dettagli: "we don't need to cover the intermediate router states where a future route is already known. [...] In addition we check is the user is logged in as otherwise we would get (failing) calls after logout" (`feature-libs/quote/core/facade/quote.service.ts:L400-L402`).
- `reloadOn: []` **nella lista**: "Changes to currentPage$/sort$ are now handled reactively within the factory itself, so reloadOn is no longer needed." (`feature-libs/quote/core/facade/quote.service.ts:L446-L446`).
- **Route** `quotes` **dal CMS**: "The ng route corresponding to the cxRoute 'quotes' is not defined here as it is provided via CMS." (`feature-libs/quote/root/quote-root.module.ts:L88-L93`).
- **Marker component**: "This component is used as a marker. In case it is present in the CMS composition of a page, this page can be navigated to only if the QuoteCartGuard [...] allows it." (`feature-libs/quote/components/cart-guard/quote-cart-guard.component.ts:L10-L17`) e il JSDoc del guard sui due casi di redirect (`feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L19-L24`).
- **Niente** `AuthGuard` **sul request button**: "We cannot enforce that via an authGuard here because otheriwise the entire cart page would need an authenticated user." (`feature-libs/quote/components/request-button/quote-request-button.component.ts:L32-L37`).
- **cartBase sempre caricato** perché `getQuoteDetails()` interroga sempre `activeCartFacade` (`feature-libs/quote/components/items/quote-items.component.ts:L29-L33`).
- **Scenario ASM del 404** (`feature-libs/quote/root/http-interceptors/quote-not-found.handler.ts:L44-L50`).
- **TODO sort hard-coded** per gap API, CXEC-31715 (`feature-libs/quote/components/list/quote-list-component.service.ts:L22-L23`).
- **Deprecazioni**: `isShowDownloadProposalButtonFeatureEnabled` obsoleto dal 2211.41, ritorna sempre `true` (`feature-libs/quote/components/links/quote-links.component.ts:L95-L103`); tre metodi della lista bill spostati in `DateRangeModalComponent` (`feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L117-L158`).
- **Altezza fissa 226 px** dei bottoni azione, da tenere allineata con `_quote-summary-actions.scss` (`feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L76-L80`).
- `resetOn` **commentato** nella query subscription, residuo copiato dal ticketing (`feature-libs/subscription-billing/core/facade/subscription.service.ts:L55-L55`).
- **Core bundled con components** per ticketing (`feature-libs/customer-ticketing/root/customer-ticketing-root.module.ts:L38-L39`).
- **Triple-slash reference** ai tipi globali nel `public_api` del ticketing (`feature-libs/customer-ticketing/public_api.ts:L7-L14`).

8. Extension point

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
Ordine e primarietà delle azioni quote	`provideConfig({ quote: { actions: { primaryActions, actionsOrderByState } } })`	`feature-libs/quote/core/config/quote-core.config.ts:L11-L29`	50 — Ricette config/DI

EXTENSION POINT	COME	RIFERIMENTO	RICETTA
Dialog di conferma per stato/ruolo	`quote.confirmActionDialogMapping` (`i18nKeyPrefix`, `showOnlyWhenCartIsNotEmpty`, …)	`feature-libs/quote/components/config/quote-ui.config.ts:L34-L57`	50 — Ricette config/DI
Sostituire la logica della facade	`{ provide: QuoteFacade, useExisting: MyQuoteService }` nel chunk, oppure estendere `QuoteService`	`feature-libs/quote/core/facade/facade-providers.ts:L11-L17`	53 — Ricette feature lazy
Adapter/converter OCC	`QuoteAdapter`, `QUOTE_NORMALIZER` (multi), 5 serializer; ticketing ha 7 token; subscription ha solo adapter (nessun converter)	`feature-libs/quote/core/connectors/converters.ts:L20-L46`, `feature-libs/customer-ticketing/core/connectors/converters.ts:L18-L45`, `feature-libs/subscription-billing/occ/adapters/occ-subscription.adapter.ts:L36-L40`	52 — Ricette OCC
Endpoint	`backend.occ.endpoints.*` (augment di `OccEndpoints`)	`feature-libs/quote/occ/model/occ-quote-endpoints.model.ts:L61-L63`	52 — Ricette OCC
Componenti CMS	sovrascrivere `cmsComponents` (es. `QuoteSummaryComponent`, `SupportTicketDetailsComponent`, `SubscriptionDetailsComponent`)	`feature-libs/quote/components/summary/quote-summary.module.ts:L30-L37`	51 — Ricette CMS/outlet/layout
Guard del quote cart	estendere `QuoteCartGuard` (`checkAllowedCheckoutNavigation` è `protected`) e rimappare `QuoteCartGuardComponent.guards`	`feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L67-L80`	51 — Ricette CMS/outlet/layout
Outlet del carrello per subscription	`provideOutlet` su `CartOutlets.SUBSCRIPTION_PRICE_*` (REPLACE/BEFORE/AFTER)	`feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L47-L56`	51 — Ricette CMS/outlet/layout
Error handler	estendere `QuoteBadRequestHandler`/`QuoteNotFoundHandler`/`NotFoundTicketRequestHandler` (metodi protected) e registrarli come `HttpErrorHandler` multi	`feature-libs/quote/root/quote-root.module.ts:L108-L118`	50 — Ricette config/DI
Eventi di reload	dispatch di `QuoteDetailsReloadQueryEvent`, `GetTicketQueryReloadEvent`, `GetSubscriptionByCodeReloadEvent`; override dei metodi protected `get*QueryReloadEvents` e `get*QueryResetEvents` di `CustomerTicketingService`	`feature-libs/quote/core/event/quote.events.ts:L9-L14`, `feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L57-L84`	50 — Ricette config/DI
Dialog	`LAUNCH_CALLER.QUOTE_ACTION_CONFIRMATION`, `CUSTOMER_TICKETING_*`, `SUBSCRIPTION_ACTION_CONFIRMATION`, `DATE_RANGE_MODAL` con `launch` config	`feature-libs/quote/components/confirm-dialog/default-quote-confirm-dialog.config.ts:L10-L18`	51 — Ricette CMS/outlet/layout
Page meta	`PageMetaResolver` multi (ticket, subscription)	`feature-libs/customer-ticketing/core/customer-ticketing-core.module.ts:L14-L31`	52 — Ricette OCC/meta resolver
Traduzioni	chunk `quote`, `customerTicketing`, `myAccountV2CustomerTicketing`	`feature-libs/quote/assets/translations/translations.ts:L9-L11`, `feature-libs/customer-ticketing/assets/translations/translations.ts:L9-L18`	53 — Ricette traduzioni

Attenzione all'API pubblica: `OccQuoteActionNormalizer` e `OccQuoteEntryNormalizer` non sono esportati da `occ/public_api.ts` (feature-libs/quote/occ/public_api.ts:L7-L9); per estenderli bisogna importare da path interni. `SubscriptionDetailsPageMetaResolver` e `SubscriptionBillingDetailsPageMetaResolver` invece sono pubblici in `@spartacus/subscription-billing/core` (feature-libs/subscription-billing/core/services/index.ts:L6-L7, feature-libs/subscription-billing/core/public_api.ts:L10-L10). Vedi 54 — API pubblica e upgrade.

---

9. Trappole, casi limite, comportamenti controintuitivi

Quote

1. `requote` **e** `downloadAttachment` **non sono proxati.** Fuori dal chunk `quote` (dove `QuoteFacade` è ancora il proxy) chiamarli dà un metodo non definito (feature-libs/quote/root/facade/quote.facade.ts:L27-L36, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L119-L123). Il chunk `request-button` usa sempre il proxy, ma chiama solo `createQuote`, che è proxato (feature-libs/quote/components/request-button/quote-request-button.component.ts:L43-L57).
2. **Il componente ignora l'Observable di** `requote`: funziona solo perché `Command.execute` è eager (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L309-L311, core-libs/core/src/util/command-query/command.service.ts:L126-L131).
3. `isEditable` **significa "OCC permette EDIT ma EDIT non è un bottone"**: è `false` finché `EDIT` è offerta come bottone; tipicamente indica "già in edit", ma è `true` anche per gli stati il cui ordine configurato filtra via `EDIT` (con i default `EXPIRED`, `CANCELLED`) (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L37-L43, feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L82-L83, feature-libs/quote/core/config/default-quote.core.config.ts:L30-L31).
4. **Azioni OCC scartate silenziosamente**: per gli stati con ordine configurato, le azioni non elencate spariscono; per gli altri passano tutte senza ordine (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L82-L89). Se aggiungi un'azione lato backend, aggiungila anche in `actionsOrderByState`.
5. **Lettura sincrona dello stato carrello nel converter**: `take(1)` dentro `subscribe` in `convert`; se il cart non ha ancora emesso, `EDIT` non viene rimosso (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L65-L81). La query dettagli mitiga attendendo una prima emissione del carrello (feature-libs/quote/core/facade/quote.service.ts:L388-L397).
6. `currentPage = 0` **non inviato** (test di truthiness) (feature-libs/quote/occ/adapters/occ-quote.adapter.ts:L59-L61).
7. `view.defaultPageSize: 5` **globale**: fornito come default config da `QuoteListModule` (feature-libs/quote/components/list/quote-list.module.ts:L36-L40); dopo il caricamento del chunk viene fuso nel `Config` globale (salvo `disableConfigUpdates`) e può influenzare altri consumer di `ViewConfig` che non impostano il valore (core-libs/core/src/config/services/configuration.service.ts:L92-L106).
8. **Campo PO incoerente**: `defineQuoteMetaData` invia `sapPurchaseOrderNumber` mentre `QuoteMetadata` dichiara `purchaseOrderNumber` (feature-libs/quote/components/header/overview/quote-header-overview.component.ts:L71-L78, feature-libs/quote/root/model/quote.model.ts:L120-L128).
9. **Proprietà non enumerabili**: il buyer edit usa `Object.defineProperty` senza `enumerable`, per questo il consumer usa `Object.getOwnPropertyNames` (feature-libs/quote/components/header/buyer-edit/quote-header-buyer-edit.component.ts:L65-L78, feature-libs/quote/components/header/overview/quote-header-overview.component.ts:L68-L68).
10. **Validatori che si accumulano**: `fillFormAttributes` chiama `addValidators` a ogni emissione della quote (feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.ts:L105-

L110).

11. **Sconti ABSOLUTE impostati dal backend** non sono mostrati nel campo; il client invia sempre `PERCENT` (feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.ts:L112-L121, feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.ts:L129-L184).
12. **Eccezione con locale senza** `%`: `getLocalizationElements` lancia `No percentage sign present` (feature-libs/quote/components/summary/seller-edit/quote-summary-seller-edit.component.service.ts:L142-L152).
13. **Listener di chiusura dialog**: `handleConfirmationDialogClose` ricrea la subscription a ogni click; una chiusura senza `'yes'` lascia il listener attivo fino al click successivo o al destroy (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L283-L299).
14. **Download proposta**: solo `attachments[0]`, estensione `.pdf` forzata (feature-libs/quote/components/links/quote-links.component.ts:L69-L74).
15. **Dipendenza dal testo backend** `'Quote not found'` (feature-libs/quote/root/http-interceptors/quote-not-found.handler.ts:L54-L54) e dalle regex dei messaggi `IllegalArgumentError` (feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L121-L133): una localizzazione dei messaggi lato backend rompe il matching.
16. **Toggle requote**: con il default core (`false`) `REQUOTE` non viene disabilitato sotto soglia; l'app demo lo attiva (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1110-L1110, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L394-L394). Vedi 42 — Feature toggle.

Customer ticketing

17. `ticketCode` **letto una volta**: le precondizioni usano `take(1)`; cambiare ticket richiede un reload esplicito, che il dettaglio fa quando il ticket in cache non corrisponde alla route (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L254-L270, feature-libs/customer-ticketing/components/details/customer-ticketing-details/customer-ticketing-details.component.ts:L83-L97).
18. **Con allegato niente Reopened/NewMessage**: non compare `requestReopened`; il reload arriva solo da `UploadAttachmentSuccessEvent` (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L121-L130).
19. `reverse()` **in place** sulla risposta di `getTicket` (feature-libs/customer-ticketing/occ/adapters/occ-customer-ticketing.adapter.ts:L99-L99).
20. **Tipo di ritorno di** `createTicket`: tipato `Observable<TicketStarter>` nell'adapter ma consumato come `TicketDetails` (feature-libs/customer-ticketing/core/connectors/customer-ticketing.adapter.ts:L36-L39, feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L325-L329).
21. **Config morta**: `agentSessionTimer` non è letto da nessun file della lib (feature-libs/customer-ticketing/root/config/customer-ticketing-config.ts:L16-L18).
22. `window`/`document` **globali** nel download allegato (non `WindowRef`): eseguibile solo nel browser, dove il click avviene (feature-libs/customer-ticketing/components/details/customer-ticketing-messages/customer-ticketing-messages.component.ts:L77-L86).
23. **Messaggi backend raw**: gli errori `HttpErrorModel` della creazione sono mostrati con `{ raw: err.message }` (feature-libs/customer-ticketing/components/list/customer-ticketing-create/customer-ticketing-create-dialog/customer-ticketing-create-dialog.component.ts:L184-L206).

Subscription billing

24. `productCode` **letto troppo presto**: `SubscriptionProductPriceComponent` usa l'`@Input()` in un field initializer, prima dell'assegnazione degli input, quindi in pratica usa sempre `currentProductService` (feature-libs/

subscription-billing/components/product/price/subscription-product-price.component.ts:L30-L35, feature-libs/subscription-billing/root/services/subscription-product.service.ts:L18-L28).

25. **Config di estensione non personalizzabile**: `ExtendSubscriptionFrequencyDropdownOptionsConfig` augmenta `Config` ma il modal usa la costante `defaultExtendDropdownOptions` (feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L84-L85, feature-libs/subscription-billing/root/config/extend-subscription-dropdown-options-config.ts:L10-L23).
26. **Precondizioni senza** `take(1)`: navigare fuori da `/subscription/` fa lanciare l'errore delle precondizioni → stato `error` della query (feature-libs/subscription-billing/core/facade/subscription.service.ts:L111-L123).
27. **Azioni fredde**: senza subscribe nessuna richiesta parte (diverso da quote/ticketing con `Command`) (feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L55-L77); `getExtensionEffectiveDate` senza `take(1)` rifà la POST a ogni cambio di userId (feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L123-L138).
28. `dialogClose` **globale**: la lista bill si sottoscrive nel costruttore senza unsubscribe e reagisce a qualunque dialog chiuso con un oggetto (feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L77-L90).
29. `?fields=FULL` **+ queryParams**: l'endpoint `subscriptionBillsList` contiene già `?fields=FULL`, ma `buildUrl` separa la query dell'endpoint e la unisce ai `queryParams` con `fromString`, quindi non si producono due `?` (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L10-L43, core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L146).
30. **Nomi schematics asimmetrici**: `addCustomerTicketingFeatures` (plurale) vs `addSubscriptionBillingFeature` (singolare) (feature-libs/customer-ticketing/schematics/add-customer-ticketing/index.ts:L25-L25, feature-libs/subscription-billing/schematics/add-subscription-billing/index.ts:L25-L25).

---

10. Come fare debug

- **"Il bottone X non compare"**: confronta `allowedActions` nella risposta di rete di `getQuote` con `actionsOrderByState[state]` effettivo (`ConfigurationService.config.quote`); ricorda che EDIT sparisce se il quote cart attivo è proprio quella quote (feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L60-L89). Un breakpoint in `OccQuoteActionNormalizer.convert` mostra input e output.
- **"Il bottone è disabilitato"**: controlla `quote.threshold` e `totalPrice.value` e il toggle per REQUOTE (feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L219-L232).
- **"Vengo rimbalzato sulla pagina della quote"**: è `QuoteCartGuard`. Verifica se il carrello attivo ha `quoteCode` (feature-libs/quote/core/services/quote-cart.service.ts:L18-L45) e il valore di `checkoutAllowed` (diventa `true` solo da `loadQuoteCartAndProceed` per CHECKOUT, feature-libs/quote/core/facade/quote.service.ts:L345-L347). Nello store NgRx del router controlla `nextState.semanticRoute` (feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L67-L80).
- **"La quote non si aggiorna"**: la query dettagli si ricarica solo su `QuoteDetailsReloadQueryEvent` e `LoginEvent` (feature-libs/quote/core/facade/quote.service.ts:L418-L420); osserva `EventService.get(QuoteDetailsReloadQueryEvent)` per capire se l'evento parte (cart listener, link, commenti).
- **Spinner infinito**: `isActionPerforming$` torna a `false` solo nei rami previsti (feature-libs/quote/core/facade/quote.service.ts:L244-L292, feature-libs/quote/core/facade/quote.service.ts:L356-L360); per EDIT/CHECKOUT dipende dall'arrivo del cart atteso in `getActive()`.

- **Messaggi d'errore inattesi**: verifica quale `HttpErrorHandler` vince (priorità `NORMAL` degli handler quote, feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L54-L56) e se il payload OCC corrisponde ai tipi/regex attesi.
- **Ticketing**: per lista o dettaglio "vecchi", segui la catena evento → listener (feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L64-L135); un errore "Customer ticketing pre conditions not met" indica utente anonimo (feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L254-L270).
- **Subscription**: se il dettaglio va in errore controlla che l'URL contenga `/subscription/<id>` (regex, feature-libs/subscription-billing/core/facade/subscription.service.ts:L126-L134); se un'azione "non fa nulla", verifica che qualcuno sottoscriva l'Observable restituito dalla facade.
- **Chunk caricati**: nel pannello Network deve comparire il chunk di `@spartacus/quote/components/cart-guard` sulle pagine con il marker, quello di `request-button` sul carrello e quello principale solo quando serve `QuoteService` (projects/storefrontapp/src/app/spartacus/features/quote-feature.module.ts:L26-L63). Vedi anche 12 — Feature module e lazy loading.

---

11. Domande di autoverifica

1. **Perché la lib quote definisce tre nomi feature invece di uno?** Per caricare marker del guard e bottone "Request quote" senza il chunk completo: sono mappati su `quote_cart_guard` e `quote_request`, con `import()` separati. — feature-libs/quote/root/quote-root.module.ts:L46-L64, projects/storefrontapp/src/app/spartacus/features/quote-feature.module.ts:L26-L63
2. **Cosa succede se un componente fuori dal chunk `quote` chiama `quoteFacade.requote(code)`?** Sul proxy `requote` non è definito, perché non è in `methods`; il proxy definisce solo i metodi elencati. — feature-libs/quote/root/facade/quote.facade.ts:L27-L36, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L119-L123
3. **Quando `Quote.isEditable` è `true`?** Quando OCC permette EDIT ma EDIT è stato rimosso dalle azioni visualizzate, tipicamente quando la quote (non in `BUYER_OFFER`) è già nel quote cart attivo, ma anche negli stati il cui ordine configurato non contiene `EDIT` (con i default `EXPIRED` e `CANCELLED`), dove il filtro lo scarta. — feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L37-L43, feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L70-L80, feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L82-L83, feature-libs/quote/core/config/default-quote.core.config.ts:L30-L31
4. **Come fa `QuoteCartGuard` a lasciar passare il checkout di una quote?** `loadQuoteCartAndProceed` imposta `setCheckoutAllowed(true)` prima di navigare; il guard consente se la prossima route semantica inizia con `checkout` (o si è già in checkout senza nextState) e il checkout è consentito. — feature-libs/quote/core/facade/quote.service.ts:L345-L347, feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L67-L80
5. **In che ordine viene cercato il config del dialog di conferma?** Stato → ruolo derivato dal prefisso dello stato → `ALL`; serve il dialog se il config esiste e il carrello non è vuoto/quote cart oppure `showOnlyWhenCartIsNotEmpty` è falso. — feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L349-L363, feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L411-L419
6. **Perché entrare in modalità EDIT non fa perdere il carrello normale?** `saveActiveCart` salva come saved cart il carrello attivo non vuoto e non quote cart prima di caricare il quote cart. — feature-libs/quote/core/facade/quote.service.ts:L461-L487

7. **Perché dopo aver aggiunto un prodotto al quote cart i totali della quote si aggiornano?** `QuoteCartEventListener` trasforma gli eventi di successo del carrello in `QuoteDetailsReloadQueryEvent`, che è nel `reloadOn` della query dettagli. — `feature-libs/quote/core/event/quote-cart-event.listener.ts:L29-L49`, `feature-libs/quote/core/facade/quote.service.ts:L418-L420`
8. **Che differenza c'è tra le strategie dei command quote e ticketing?** Quote usa `CancelPrevious` (`switchMap`, il comando nuovo cancella il precedente); ticketing usa `Queue` (`concatMap`, i comandi si accodano). — `feature-libs/quote/core/facade/quote.service.ts:L127-L129`, `feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L100-L102`, `core-libs/core/src/util/command-query/command.service.ts:L73-L116`
9. **Quale evento parte quando si riapre un ticket con un allegato?** Nessun `TicketReopenedEvent`: con `containsAttachment` il servizio non dispatcha Reopened/NewMessage; il reload arriva da `UploadAttachmentSuccessEvent`. — `feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L118-L130`, `feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L151-L163`
10. **Perché i componenti del prezzo di abbonamento in carrello esistono anche se il chunk `subscriptionBilling` non è stato caricato, e perché `cancelSubscription()` senza subscribe non fa nulla?** Gli outlet `SUBSCRIPTION_PRICE_*` sono forniti dal root module eager con componenti standalone (il prezzo appare però solo se le entry hanno `sapSubscriptionTerm`/`sapPricePlan`: `DA VERIFICARE`, dipende dal livello `DEFAULT` del backend); le azioni di `SubscriptionActionsService` sono Observable freddi, non `Command`. — `feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L42-L59`, `feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L55-L77`

TAVOLA 31.1 **4.2 Quote: dalla risposta OCC ai bottoni**

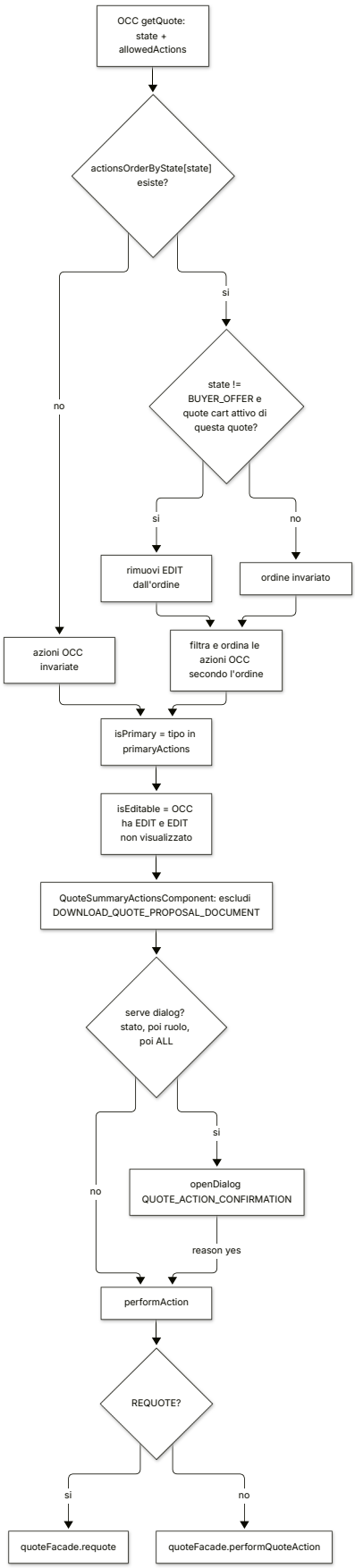
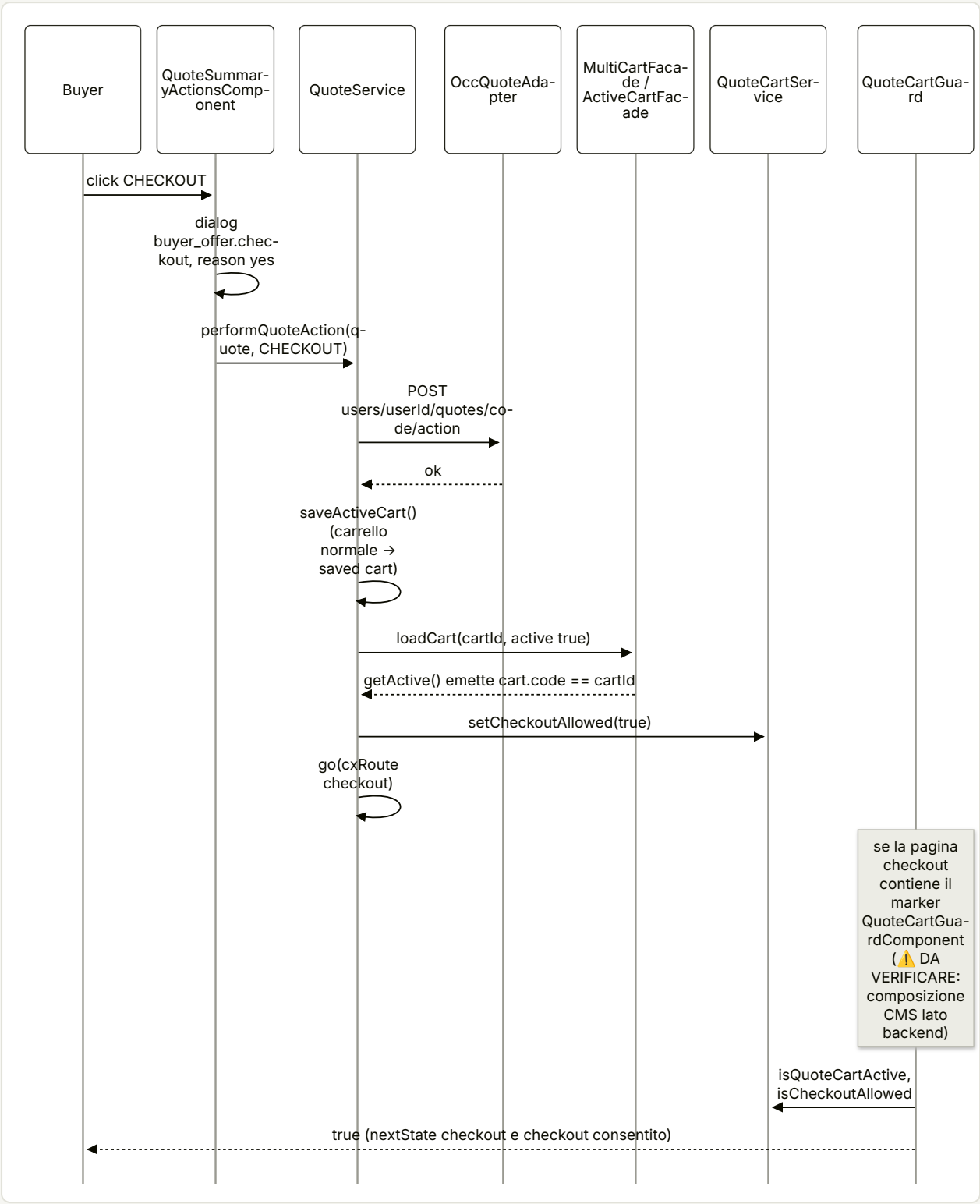


TAVOLA 31.2 **4.3 Quote: CHECKOUT di una quote in BUYER_OFFER**



32

Pickup in store, store finder, delivery date

"Dove e quando arriva la merce": data di consegna richiesta e stimata, store finder, ritiro in negozio.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 12 — Feature module e lazy loading, 17 — Outlet, 19 — State management, 20 — Facade, command & query, eventi, 21 — Livello OCC. Collegati: 25 — Carrello, 26 — Checkout, 27 — Ordini, 11 — Configurazione, 41 — Errori e global message.

1. A cosa serve e perché esiste

Questo capitolo raccoglie quattro feature library che hanno in comune il tema "dove e quando arriva la merce":

LIBRERIA	PACCHETTO	COSA FA	STATO
`requested-delivery-date` (RDD)	`@spartacus/requested-delivery-date`	Permette di scegliere una data di consegna richiesta durante il checkout (step delivery mode) e la mostra nel riepilogo ordine	nessun NgRx, facade lazy + chiamata OCC diretta
`estimated-delivery-date` (EDD)	`@spartacus/estimated-delivery-date`	Mostra la data stimata di arrivo (arrival slots) sulle righe del carrello / ordine	nessun NgRx, solo un componente + un override degli endpoint carrello
`storefinder`	`@spartacus/storefinder`	Pagina "trova negozio": ricerca testuale o per geolocalizzazione, conteggi per paese/regione, dettaglio negozio, mappa Google	NgRx (`stores`) + facade lazy
`pickup-in-store`	`@spartacus/pickup-in-store`	"Ritiro in negozio" (BOPIS): scelta del negozio dal PDP e dal carrello, stock per negozio, negozio preferito, riepiloghi in checkout/conferma	NgRx (3 feature) + 4 facade lazy

Le quattro librerie seguono lo schema classico delle feature lib descritto nel capitolo 12: un entry point `root` caricato eager (nomi feature, config CMS lazy, outlet, facade astratte, modelli augmentati) e un modulo "pesante" (core + occ + componenti) caricato on demand.

Il tratto comune più interessante è che **nessuna delle quattro modifica il codice del carrello o del checkout**: si agganciano tramite *outlet* esposti da cart/checkout/order (`CartOutlets.DELIVERY_MODE`, `CartOutlets.ORDER_OVERVIEW`, `CartOutlets.ITEM_DETAILS`, `CartOutlets.ITEM_DELIVERY_DETAILS`, `CartOutlets.ADD_TO_CART_PICKUP_OPTION`, …) (feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L31), tramite *module augmentation* dei modelli (`Cart`, `Order`, `OrderEntry`, `Consignment`) e tramite *override di configurazione* (endpoint OCC, dipendenze tra feature). Sono quindi un ottimo catalogo di tecniche di estensione "non invasive", e allo stesso tempo contengono parecchi casi limite che vale la pena conoscere (sezione 9).

In sintesi:

- **RDD**: un componente in outlet AFTER sul delivery mode del checkout chiama `RequestedDeliveryDateFacade.setRequestedDeliveryDate` → `PUT users/${userId}/carts/${cartId}/requestedretrievaldate?requestedRetrievalAt=…` (feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L35-L50, feature-libs/requested-delivery-date/occ/config/default-occ-requested-delivery-date-config.ts:L9-L18).
- **EDD**: un componente in outlet AFTER su `CartOutlets.ITEM_DETAILS` legge `arrivalSlots` dall'entry; perché il backend li restituisca, la lib ridefinisce gli endpoint `carts`/`cart`/`createCart` aggiungendo `arrivalSlots` ai `fields` (feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L28-L35, feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L10-L28).

- **Store finder**: un solo componente CMS (`StoreFinderComponent`) con *child routes* che montano i sotto-componenti; stato NgRx `stores` con due slice loader; rendering mappa via `GoogleMapRendererService` (feature-libs/storefinder/components/store-finder-components.module.ts:L66-L99, feature-libs/storefinder/core/store/store-finder-state.ts:L10-L29).
- **Pickup in store**: si dichiara dipendenza di `addToCart` e `cartBase` così da essere caricato prima di loro, sostituisce (REPLACE) gli outlet di "opzione di consegna" del PDP e del carrello, e apre un dialog di selezione negozio basato su stock (feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L39-L60).

2. Mappa dei file

PATH	RUOLO
feature-libs/requested-delivery-date/requested-delivery-date.module.ts	`RequestedDeliveryDateModule` = Core + Occ (:L11-L14); i componenti stanno in `root`.
feature-libs/requested-delivery-date/root/feature-name.ts	`requestedDeliveryDate`, `requestedDeliveryDateCore` (:L7-L8).
feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts	CMS config lazy + due `provideOutlet` AFTER (:L18-L50).
feature-libs/requested-delivery-date/root/facade/requested-delivery-date.facade.ts	Facade lazy con un solo metodo (:L12-L32).
feature-libs/requested-delivery-date/root/model/augmented-core.model.ts	Augmentation `Cart`/`Order` (:L10-L21).
feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts	Date picker nel delivery mode (:L38-L180).
feature-libs/requested-delivery-date/root/components/order-overview-delivery-date/order-overview-delivery-date.component.ts	Card read-only nell'order overview (:L31-L61).
feature-libs/requested-delivery-date/root/components/shared/date-validation.service.ts	Parsing/validazione date (:L9-L52).
feature-libs/requested-delivery-date/core/requested-delivery-date-core.module.ts	Service, alias facade, `HttpErrorHandler` multi, connector (:L14-L29).
feature-libs/requested-delivery-date/core/http-interceptors/bad-request/requested-delivery-date-badrequest.handler.ts	Handler 400 `ValidationError` (:L17-L49).
feature-libs/requested-delivery-date/occ/adapters/occ-requested-delivery-date.adapter.ts	`PUT` OCC (:L25-L44).
feature-libs/estimated-delivery-date/estimated-delivery-date.module.ts	Importa solo `ShowEstimatedDeliveryDateModule` (:L10-L13).
feature-libs/estimated-delivery-date/root/estimated-delivery-date-root.module.ts	Root module vuoto (:L9-L10).
feature-libs/estimated-delivery-date/root/model/augmented-core.model.ts	`OrderEntry.arrivalSlots`, `Consignment.arrivalSlot` (:L11-L21).

PATH	RUOLO
feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts	Override endpoint + outlet ITEM_DETAILS (:L28-L35).
feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts	Endpoint carrello con arrivalSlots (:L10-L28).
feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts	Componente EDD (:L19-L74).
feature-libs/storefinder/root/store-finder-root.module.ts	Layout, routing, CMS config lazy (:L17-L38).
feature-libs/storefinder/root/config/default-store-finder-routing-config.ts	Rotte semantiche storeFinder* (:L9-L35).
feature-libs/storefinder/root/facade/store-finder.facade.ts	Facade lazy async: true (:L21-L73).
feature-libs/storefinder/root/model/store-entities.model.ts	StoreFinderOutlets.PREFERRED_STORE, StoreEntities (:L9-L15).
feature-libs/storefinder/core/config/default-store-finder-config.ts	Default googleMaps (:L9-L21).
feature-libs/storefinder/core/facade/store-finder.service.ts	Service concreto, geolocalizzazione, reload su cambio contesto (:L26-L259).
feature-libs/storefinder/core/service/google-map-renderer.service.ts	Caricamento script Google Maps e marker (:L44-L275).
feature-libs/storefinder/core/service/store-location.service.ts	Lat/lng e link direzioni (:L10-L41).
feature-libs/storefinder/core/store/**	Azioni, effetti, reducer, selettori, stato stores.
feature-libs/storefinder/occ/adapters/occ-store-finder.adapter.ts	Adapter OCC (:L36-L103).
feature-libs/storefinder/components/store-finder-components.module.ts	Mapping CMS con childRoutes (:L38-L99).
feature-libs/storefinder/components/**	13 componenti (search, result/list, grid, map, store, stores-count, ...).
feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts	CMS lazy + dipendenze di addToCart/cartBase (:L23-L60).
feature-libs/pickup-in-store/root/facade/*.facade.ts	4 facade lazy (IntendedPickupLocationFacade, PickupLocationsSearchFacade, PickupOptionFacade, PreferredStoreFacade).
feature-libs/pickup-in-store/root/model/augmented-core.model.ts	LAUNCH_CALLER.PICKUP_IN_STORE (:L9-L15).
feature-libs/pickup-in-store/core/pickup-in-store-core.module.ts	Config, connector, facade providers, store (:L14-L23).
feature-libs/pickup-in-store/core/store/**	3 feature NgRx: pickup-locations, pickup-option, stock.
feature-libs/pickup-in-store/core/services/preferred-store.service.ts	Negozio preferito (:L32-L97).

PATH	RUOLO
feature-libs/pickup-in-store/occ/adapters/occ-stock.adapter.ts	Stock per prodotto / per negozio (:L35-L51).
feature-libs/pickup-in-store/components/container/**	Container: PDP, carrello, dialog, store list/search, preferred store, riepiloghi.
feature-libs/pickup-in-store/components/services/delivery-points.service.ts	Raggruppa le entry pickup per negozio (:L66-L160).
feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts	Legge la scelta pickup al momento dell'add (:L267-L287).
feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts	PUT vs PATCH per pickup ↔ delivery (:L86-L104).

File aggiuntivi

PATH	RUOLO
feature-libs/estimated-delivery-date/public_api.ts	Entry point principale: riesporta EstimatedDeliveryDateModule e la public API di show-estimated-delivery-date (:L10-L11).
feature-libs/estimated-delivery-date/assets/public_api.ts	Entry point assets: riesporta le traduzioni (:L7-L7).
feature-libs/estimated-delivery-date/_index.scss	Entry Sass della lib: importa il core styles e styles/index (:L1-L2).
feature-libs/estimated-delivery-date/styles/_index.scss	Importa _estimated-delivery-date.scss (:L1-L1).
feature-libs/estimated-delivery-date/styles/_estimated-delivery-date.scss	Regole globali (non placeholder) per cx-estimated-delivery-date e .cx-estimated-delivery-date-info (:L1-L23).
feature-libs/estimated-delivery-date/schematics/add-estimated-delivery-date/index.ts	Factory ng-add addEstimatedDeliveryDateFeature (:L25-L45).
feature-libs/estimated-delivery-date/schematics/add-estimated-delivery-date/index_spec.ts	Test Jest dello schematic: lazy loading nel wrapper cart-base, nessuna feature user/order installata (:L110-L126).
feature-libs/estimated-delivery-date/vitest.config.ts	Config Vitest (jsdom, coverage v8, soglie 85/85/65/85) (:L14-L47).
feature-libs/pickup-in-store/pickup-in-store.module.ts	PickupInStoreModule = Components + Core + Occ (:L12-L19).
feature-libs/pickup-in-store/public_api.ts	Entry point principale: riferimento a types.d.ts e riesportazione del module (:L14-L19).
feature-libs/pickup-in-store/_index.scss	Entry Sass: definisce $cx-color-link-primary e importa core, modal, table e styles/index (:L1-L7).
feature-libs/pickup-in-store/schematics/add-pickup-in-store/index.ts	Factory ng-add addPickupInStoreFeatures (:L25-L45).
feature-libs/pickup-in-store/schematics/add-pickup-in-store/index_spec.ts	Test Jest dello schematic (senza feature, lazy, styling, eager) (:L25-L144).
feature-libs/requested-delivery-date/public_api.ts	Entry point principale: riesporta solo RequestedDeliveryDateModule (:L10-L10).

PATH	RUOLO
feature-libs/requested-delivery-date/_index.scss	Entry Sass: core + styles/index (:L1-L2).
feature-libs/requested-delivery-date/styles/_index.scss	Importa _requested-delivery-date.scss (:L1-L1).
feature-libs/requested-delivery-date/styles/_requested-delivery-date.scss	Regole globali per cx-request-delivery-date e per .cx-review-card-shipping del checkout (:L1-L25).
feature-libs/requested-delivery-date/schematics/add-requested-delivery-date/index.ts	Factory ng-add addRequestedDeliveryDateFeature (:L25-L45).
feature-libs/requested-delivery-date/schematics/add-requested-delivery-date/index_spec.ts	Test Jest dello schematic (:L21-L131).
feature-libs/requested-delivery-date/vitest.config.ts	Config Vitest (:L14-L58).
feature-libs/storefinder/public_api.ts	Entry point principale: riesporta store-finder.module (:L10-L10).
feature-libs/storefinder/_index.scss	Entry Sass: core + styles/index (:L1-L2).
feature-libs/storefinder/schematics/add-storefinder/index.ts	Factory ng-add addStorefinderFeatures (:L25-L45).
feature-libs/storefinder/schematics/add-storefinder/index_spec.ts	Test Jest dello schematic (:L21-L125).
feature-libs/storefinder/vitest.config.ts	Config Vitest con resolve.alias verso helper di test di core/storefront (:L16-L22).

3. Come funziona passo per passo

3.1 Requested Delivery Date

Wiring. Il root module fa due cose: dichiara la config CMS lazy della feature requestedDeliveryDate (componenti DeliveryModeDatePickerComponent e OrderOverviewDeliveryDateComponent) mappando la feature core sulla principale — "by default core is bundled together with components" — e registra **eagerly** due outlet AFTER (feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L18-L50). Gli outlet hanno id cx-delivery-mode e cx-order-overview (feature-libs/cart/base/root/models/cart-outlets.model.ts:L23-L24): il primo è renderizzato dal componente delivery mode del checkout (feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.html:L55-L55), il secondo dall'order overview (feature-libs/order/components/order-details/order-overview/order-overview.component.html:L43-L43).

Particolarità: i componenti vivono nell'entry point root ("Make components available without being CMS driven") (feature-libs/requested-delivery-date/root/public_api.ts:L7-L11). Questo è necessario perché provideOutlet riceve direttamente la classe del componente, quindi il componente deve essere disponibile già nel bundle eager. Il modulo "pesante" RequestedDeliveryDateModule contiene solo core + OCC (feature-libs/requested-delivery-date/requested-delivery-date.module.ts:L11-L14).

Facade. RequestedDeliveryDateFacade è un facade lazy (facadeFactory) con un solo metodo, setRequestedDeliveryDate(userId, cartId, requestedDate): Observable<{}>, senza async (feature-libs/requested-delivery-date/root/facade/requested-delivery-date.facade.ts:L12-L32). Al primo uso il facade

risolve la feature `requestedDeliveryDate`, il cui modulo core lega `RequestedDeliveryDateFacade → useExisting: RequestedDeliveryDateService` e registra un `HttpErrorHandler` multi (feature-libs/requested-delivery-date/core/requested-delivery-date-core.module.ts:L14-L29). Il service delega direttamente al connector, il connector all'adapter astratto (feature-libs/requested-delivery-date/core/services/requested-delivery-date.service.ts:L19-L29, feature-libs/requested-delivery-date/core/connectors/requested-delivery-date.connector.ts:L17-L23). Nessun NgRx, nessuna `Command`: la chiamata HTTP parte alla subscribe.

Modello. La lib augmenta `Cart` con `earliestRetrievalAt` e `requestedRetrievalAt` e `Order` con `requestedRetrievalAt` (stringhe) (feature-libs/requested-delivery-date/root/model/augmented-core.model.ts:L10-L21). La data minima arriva dal backend dentro il carrello.

Componente picker. `DeliveryModeDatePickerComponent` (selector `cx-request-delivery-date`) inietta facade, `DateValidationService`, `EventService`, `GlobalMessageService` e, opzionalmente, `OutletContextData` (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L38-L62). Passi di `ngOnInit`:

1. si sottoscrive a `context$` dell'outlet e legge `context.item` (il cart) e `context.readonly` (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L74-L82);
2. se c'è `earliestRetrievalAt` valido lo memorizza; se il cart ha già `requestedRetrievalAt` lo usa, altrimenti imposta `requestedRetrievalAt = earliestRetrievalAt` e chiama **subito** `setRequestedDeliveryDate()` — cioè apre la pagina e scrive sul backend (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L84-L99).

`setRequestedDeliveryDate()` esce senza fare nulla se mancano userId, cartId o data, se la data non è valida o è precedente a `earliestRetrievalAt` (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L129-L145). Altrimenti chiama il facade; al successo dispatcha `CheckoutSupportedDeliveryModesQueryReloadEvent` (la lista dei delivery mode può dipendere dalla data) e un global message INFO; in errore mostra un messaggio solo se nella risposta ci sono errori `UnknownResourceError` (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L146-L176).

Il template mostra il picker solo se esiste `earliestRetrievalAt`; in modalità readonly mostra una `cx-card`, altrimenti `cx-date-picker` con `[min]="earliestRetrievalAt"` e `(update)="setRequestedDeliveryDate()"` (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.html:L1-L23).

Adapter OCC. L'endpoint di default è `requestedDeliveryDate: 'users/${userId}/carts/${cartId}/requestedretrievaldate'` (feature-libs/requested-delivery-date/occ/config/default-occ-requested-delivery-date-config.ts:L9-L18), aggiunto a `OccEndpoints` via augmentation (feature-libs/requested-delivery-date/occ/model/occ-requested-delivery-date-endpoints.model.ts:L9-L17). `OccRequestedDeliveryDateAdapter` costruisce header form-urlencoded con `USE_CLIENT_TOKEN`, mette la data in query string (`requestedRetrievalAt`) e chiama `this.http.put(url, { headers })` (feature-libs/requested-delivery-date/occ/adapters/occ-requested-delivery-date.adapter.ts:L25-L44). Attenzione: in `HttpClient.put` il secondo argomento è il **body** (angular:packages/common/http/src/client.ts:L4451-L4454) — vedi trappola T1 in sezione 9.

Errori. Due percorsi distinti: `RequestedDeliveryDateBadRequestHandler` (globale, registrato come `HttpErrorHandler` multi) intercetta solo i 400 con errore `ValidationError` + messaggio `checkout.multi.requestedretrievaldatevalid.error`, priorità `NORMAL` (feature-libs/requested-delivery-date/core/http-interceptors/bad-request/requested-delivery-date-badrequest.handler.ts:L20-L48); il componente gestisce gli `UnknownResourceError`. Il meccanismo degli `HttpErrorHandler` è descritto nel capitolo 41.

Order overview. `OrderOverviewDeliveryDateComponent` legge l'ordine da `context.item` dell'outlet e mostra una card intitolata `requestedDeliveryDate.readOnlyTextLabel` (feature-libs/requested-delivery-date/root/components/order-overview-delivery-date/order-overview-delivery-date.component.ts:L31-L61).

Date. `DateValidationService` presume il formato `dd-mm-yyyy` e lo converte in `mm/dd/yyyy` con una regex prima di `new Date` (feature-libs/requested-delivery-date/root/components/shared/date-validation.service.ts:L33-L35); `isDateGreaterOrEqual` restituisce `false` se una delle due stringhe è vuota (feature-libs/requested-delivery-date/root/components/shared/date-validation.service.ts:L43-L51).

3.2 Estimated Delivery Date

Wiring. `EstimatedDeliveryDateRootModule` è un `@NgModule({})` vuoto (feature-libs/estimated-delivery-date/root/estimated-delivery-date-root.module.ts:L9-L10). La costante `estimatedDeliveryDate` esiste (feature-libs/estimated-delivery-date/root/feature-name.ts:L7-L7) ma la demo non la usa come feature lazy. Se la variabile `CX_ESTIMATED_DELIVERY_DATE` è attiva (`environment.estimatedDeliveryDate`, default `false`: projects/storefrontapp/src/environments/environment.ts:L39-L39), la demo aggiunge `EstimatedDeliveryDateModule` alle `extensions` di `CartBaseWrapperModule`, quindi l'EDD **viaggia nello stesso chunk del carrello** (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L16-L18); altrimenti non viene caricato. La stessa condizione governa `EstimatedDeliveryDateFeatureModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L167-L169). Il feature module EDD della demo importa solo il root module vuoto e la config i18n (projects/storefrontapp/src/app/spartacus/features/estimated-delivery-date/estimated-delivery-date-feature.module.ts:L18-L35).

`EstimatedDeliveryDateModule` importa solo `ShowEstimatedDeliveryDateModule` (feature-libs/estimated-delivery-date/estimated-delivery-date.module.ts:L10-L13), che fa due cose (feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L28-L35):

1. `provideDefaultConfig(defaultOccCartWithEddConfig)`: ridefinisce gli endpoint `carts`, `cart`, `createCart` aggiungendo `arrivalSlots` dentro `entries(...)` (feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L10-L28). L'endpoint `cart` di default della cart lib è identico tranne per l'assenza di `arrivalSlots` (feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L17-L17).
2. `provideOutlet` AFTER su `CartOutlets.ITEM_DETAILS` (`cx-cart-item.details`), renderizzato da `cart-item` e `cart-item-list-row` (feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L54-L54, feature-libs/cart/base/root/models/cart-outlets.model.ts:L13-L13).

Chi vince sugli endpoint? Nel chunk lazy le `DefaultConfigChunk` vengono fuse in ordine con `deepMerge` dentro `ambientDefaultConfig`; la config unificata è poi `defaultConfig` < `ambientDefaultConfig` < `ambientConfig` < `rootConfig` (core-libs/core/src/config/services/configuration.service.ts:L76-L100). `deepMerge` sostituisce le stringhe (core-libs/core/src/config/utils/deep-merge.ts:L31-L48). Nella demo `CartBaseModule` (che tramite `CartBaseOccModule` fornisce `defaultOccCartConfigFactory`) è importato prima dell'EDD (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L22-L25, feature-libs/cart/base/cart-base.module.ts:L12-L15, feature-libs/cart/base/occ/cart-base-occ.module.ts:L35-L35), quindi la chunk EDD è più avanti nella lista e vince. `DA VERIFICARE`: che l'ordine dei provider `multi` nel module injector segua sempre l'ordine degli `imports` è un comportamento di Angular non riverificato qui per questo caso; e una `provideConfig` applicativa sugli stessi endpoint (rootConfig/ambientConfig) prevale comunque sull'EDD.

Modello. `ArrivalSlots {at?: Date; quantity?: number}`, `ArrivalSlot {at?: Date}` (feature-libs/estimated-delivery-date/root/model/estimated-delivery-date.model.ts:L7-L13); augmentation `OrderEntry`.

`arrivalSlots?: ArrivalSlots[]` e `Consignment.arrivalSlot?: ArrivalSlot` (feature-libs/estimated-delivery-date/root/model/augmented-core.model.ts:L11-L21).

Componente. `EstimatedDeliveryDateComponent` usa `inject()`: `orderEntry$` è `cartItemContext?.item$ ?? EMPTY` (feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L25-L32); `consignments$` appiattisce le consignment dell'ordine corrente (da `OrderHistoryFacade.getOrderDetails()`) con `switchMap` su array, emettendo una consignment alla volta (feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L34-L37).

Il template (feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.html:L1-L56):

- per ogni arrival slot dell'entry crea un testo `cx-visually-hidden` collegato con `aria-describedby` e i valori visibili `aria-hidden` (`:L4-L29`);
- se lo slot 0 non ha `quantity` mostra un riepilogo ETA con la data del primo slot (`:L30-L37`);
- in fallback mostra l'ETA della consignment solo se la consignment ha uno slot **e** l'entry non ne ha (`:L42-L56`);
- le date sono formattate con `cxDate: 'M/d/yyyy'`, formato fisso (`:L19-L22`).

3.3 Store finder — wiring e routing

`StoreFinderRootModule` fornisce layout config, routing config e la CMS config lazy con il solo `StoreFinderComponent` per la feature `storeFinder` (feature-libs/storefinder/root/store-finder-root.module.ts:L17-L38, feature-libs/storefinder/root/feature-name.ts:L7-L7). Il layout `StoreFinderPageTemplate` usa gli slot `MiddleContent`, `SideContent` (feature-libs/storefinder/root/config/default-store-finder-layout-config.ts:L9-L15). Le rotte semantiche sono `storeFinder`, `storeFinderFind`, `storeFinderViewAll`, `storeFinderCountry`, `storeFinderRegion`, `storeFinderStore` (quest'ultima con due path) (feature-libs/storefinder/root/config/default-store-finder-routing-config.ts:L9-L35).

La navigazione interna non passa da pagine CMS diverse: il componente CMS `StoreFinderComponent` dichiara `childRoutes` che montano i sotto-componenti (`find` → `StoreFinderSearchResultComponent`, `view-all` → `StoreFinderStoresCountComponent`, `country/:country` e `country/:country/region/:region` → `StoreFinderGridComponent`, i due path con `:store` → `StoreFinderStoreComponent`) (feature-libs/storefinder/components/store-finder-components.module.ts:L66-L99). Il componente stesso è solo un guscio con header e `<router-outlet />` (feature-libs/storefinder/components/store-finder/store-finder.component.html:L1-L4). Il meccanismo `childRoutes` è spiegato nel capitolo 16 e nel capitolo 14.

`StoreFinderComponentsModule` importa **di nuovo** `StoreFinderCoreModule` (feature-libs/storefinder/components/store-finder-components.module.ts:L38-L65), oltre a `StoreFinderModule` che importa Core, Occ e Components (feature-libs/storefinder/store-finder.module.ts:L12-L19): nessun danno (NgModule deduplica), ma è utile saperlo se si compone il modulo a mano.

3.4 Store finder — facade, service, NgRx

Facade. `StoreFinderFacade` è `async: true` e proxa 10 metodi; i tre deprecati (`getStoreLatitude`, `getStoreLongitude`, `getDirections`) sono dichiarati astratti ma **non** compaiono nella lista `methods` (feature-libs/storefinder/root/facade/store-finder.facade.ts:L21-L72). `FacadeFactoryService` crea proprietà solo per i metodi elencati, e con `async` aggiunge un `delay(0)` alla risoluzione (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L53-L55, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L109-L125). Il core lega `StoreFinderFacade → useExisting: StoreFinderService` (feature-libs/storefinder/core/facade/facade-providers.ts:L11-L14).

Stato. Chiave NgRx `stores`, entity type loader `'[StoreFinder] Store Finder Data'`, stato `{findStores, viewAllStores}` entrambi `LoaderState` (feature-libs/storefinder/core/store/store-finder-state.ts:L10-L29). `StoreFinderStoreModule` registra `StoreModule.forFeature('stores', reducerToken, {metaReducers})` ed `EffectsModule.forFeature(effects)` (feature-libs/storefinder/core/store/store-finder-store.module.ts:L17-L27). Entrambi i sotto-reducer sono avvolti in `loaderReducer` con **lo stesso** entity type (feature-libs/storefinder/core/store/reducers/index.ts:L21-L29); `loaderReducer` reagisce a ogni azione con `meta.entityType` uguale (core-libs/core/src/state/utils/loader/loader.reducer.ts:L32-L60) — vedi T9.

Ricerca. `StoreFinderService.findStoresAction(...)` (feature-libs/storefinder/core/facade/store-finder.service.ts:L106-L153):

- senza `useMyLocation` dispatcha direttamente `FindStores` (azione loader "load");
- con `useMyLocation` dispatcha `FindStoresOnHold` (stato loading), avvia `navigator.geolocation.watchPosition` e alla prima posizione dispatcha `FindStores` con `longitudeLatitude`; in errore mostra il global message `storeFinder.geolocationNotEnabled` e naviga a `/store-finder`.

Tutte le dispatch passano da `clearWatchGeolocation`, che chiude l'eventuale watch attivo (feature-libs/storefinder/core/facade/store-finder.service.ts:L172-L180).

`FindStoresEffect.findStores$` usa `mergeMap`, chiama il connector e, se c'è `countryIsoCode`, filtra gli store per paese **lato client** e li ordina per `name` (feature-libs/storefinder/core/store/effects/find-stores.effect.ts:L24-L62); `findStoreById$` usa `switchMap` (feature-libs/storefinder/core/store/effects/find-stores.effect.ts:L64-L84). Il reducer salva il payload di `FIND_STORES_SUCCESS` in `findStoresEntities` e quello di `FIND_STORE_BY_ID_SUCCESS` in `findStoreEntityById` (feature-libs/storefinder/core/store/reducers/find-stores.reducer.ts:L10-L33).

Conteggi. `viewAllStores$` reagisce a `VIEW_ALL_STORES` **e** a `CLEAR_STORE_FINDER_DATA`, carica i conteggi e li ordina per nome (feature-libs/storefinder/core/store/effects/view-all-stores.effect.ts:L28-L55).

Cambio contesto. `clearStoreFinderData$` mappa `LANGUAGE_CHANGE` e `CURRENCY_CHANGE` in `ClearStoreFinderData` (feature-libs/storefinder/core/store/effects/view-all-stores.effect.ts:L57-L68); il meta reducer `clearStoreFinderState` azzera lo stato su `LANGUAGE_CHANGE` e `CLEAR_STORE_FINDER_DATA` (feature-libs/storefinder/core/store/reducers/index.ts:L39-L53). A quel punto `StoreFinderService.reloadStoreEntitiesOnContextChange()` — avviato nel costruttore, solo in browser o senza `platformId` — vede entità vuote e non in caricamento, e ricarica per `country` o per `store` leggendo i route params (feature-libs/storefinder/core/facade/store-finder.service.ts:L191-L213). `callFindStoresAction` cerca con query vuota, `pageSize: -1` e il paese dai route params (feature-libs/storefinder/core/facade/store-finder.service.ts:L215-L224).

OCC. Endpoint: `store: 'stores/${storeId}?fields=FULL'`, `stores: 'stores?fields=stores(...),pagination(DEFAULT),sorts(DEFAULT)'`, `storescounts: 'stores/storescounts'` (feature-libs/storefinder/occ/adapters/default-occ-store-finder-config.ts:L9-L20). `callOccFindStores` invia `longitude`/`latitude` **al posto** di `query` se c'è una posizione, e aggiunge `radius`, `pageSize`, `currentPage`, `sort` solo se truthy (feature-libs/storefinder/occ/adapters/occ-store-finder.adapter.ts:L71-L103). `search` converte con `STORE_FINDER_SEARCH_PAGE_NORMALIZER`, `loadCounts` estrae `countriesAndRegionsStoreCount` e applica `STORE_COUNT_NORMALIZER`, `load` usa `POINT_OF_SERVICE_NORMALIZER` del core (feature-libs/storefinder/occ/adapters/occ-store-finder.adapter.ts:L36-L69). Il connector rinomina `getCounts()` → `loadCounts()` e `get(id)` → `load(id)` (feature-libs/storefinder/core/connectors/store-finder.connector.ts:L26-L32).

3.5 Store finder — componenti e Google Maps

- `StoreFinderSearchComponent` naviga a `store-finder/find` con query param `query` o `useMyLocation: true` (feature-libs/storefinder/components/store-finder-search/store-finder-search.component.ts:L47-

L61); il template ha due varianti governate da `*cxFeature="'a11yStoreFinderLabel'"` (feature-libs/storefinder/components/store-finder-search/store-finder-search.component.html:L2-L2).

- `StoreFinderSearchResultComponent` a ogni cambio di `queryParams` ricalcola la ricerca con `currentPage: 0` e `radius = googleMaps.radius`, chiamando `findStoresAction` (feature-libs/storefinder/components/store-finder-search-result/store-finder-search-result.component.ts:L52-L92); `viewPage` riesegue con la nuova pagina (feature-libs/storefinder/components/store-finder-search-result/store-finder-search-result.component.ts:L64-L74).
- `StoreFinderListComponent` alterna lista/mappa, centra la mappa sullo store cliccato, scrolla all'item selezionato dalla mappa (feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L85-L122); con toggle `a11yStoreFinderFocusOnBackButton` un `effect()` sul signal `viewChild` sposta il focus sul back button (feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L68-L83).
- `StoreFinderMapComponent` chiama il renderer a ogni `ngOnChanges` di `locations` e riemette l'indice del marker cliccato (feature-libs/storefinder/components/store-finder-map/store-finder-map.component.ts:L33-L60).
- `StoreFinderGridComponent` (OnPush) su init chiama `callFindStoresAction(route.snapshot.params)` se c'è `country` (feature-libs/storefinder/components/store-finder-grid/store-finder-grid.component.ts:L41-L52); `StoreFinderStoreComponent` carica lo store da `route.snapshot.params.store` se non riceve `@Input() location` (feature-libs/storefinder/components/store-finder-store/store-finder-store.component.ts:L46-L56); `StoreFinderStoresCountComponent` dispatcha `viewAllStores()` (feature-libs/storefinder/components/store-finder-stores-count/store-finder-stores-count.component.ts:L41-L50).
- `StoreFinderListItemComponent` espone l'outlet `StoreFinderOutlets.PREFERRED_STORE` (`cx-pick-up-in-store-make-my-store`) con contesto `{displayName, name}` (feature-libs/storefinder/components/store-finder-list-item/store-finder-list-item.component.html:L41-L47, feature-libs/storefinder/root/model/store-entities.model.ts:L9-L15): è il punto in cui pickup-in-store inietta il bottone "imposta come negozio preferito".

`GoogleMapRendererService.renderMap` (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L44-L68):

1. senza `googleMaps.apiKey` esce (con `warn` solo in dev mode);
2. esce se `locations[0]` è un oggetto vuoto;
3. se la mappa non esiste ancora carica lo script (`embedMapScript`), altrimenti ridisegna.

`embedMapScript` invia la chiave vuota se è il valore speciale `'cx-development'` (`GOOGLE_MAPS_DEVELOPMENT_KEY_CONFIG`, feature-libs/storefinder/root/config/constants.ts:L7-L7), e carica lo script con `ScriptLoader.embedScript` (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L82-L116). Con il toggle `useGoogleMapsAsyncLoading` aggiunge `loading: 'async'` + un `callback` globale; con `useAdvancedGoogleMarkers` aggiunge `libraries: 'marker'` (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L87-L108). Il callback globale ha nome univoco `__spartacusGoogleMapsInit_<n>`, si auto-cancella e non viene registrato se `document.defaultView` è null (SSR) (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L125-L142). `initMap` usa `gestureHandling: 'greedy'`, `zoom = googleMaps.scale` e `mapId` solo con advanced markers (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L175-L193). Advanced markers: `PinElement` con `glyphText` numerico, evento `gmp-click`, store senza geolocazione saltati (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L204-L250); il ramo legacy usa `google.maps.Marker` e i metodi deprecati di `StoreFinderService` (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L251-L275).

La demo usa la chiave speciale per mostrare la mappa senza chiave ("for development or demo purposes") (projects/storefrontapp/src/app/private/private.providers.ts:L46-L49).

3.6 Pickup in store — wiring

PickupInStoreRootModule (feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L23-L60):

1. CMS lazy per la feature pickupInStore: CheckoutReviewPickup, MyPreferredStoreComponent, OrderConfirmationPickUpComponent, PickupInStoreDeliveryModeComponent; pickupInStoreCore → pickupInStore;
2. aggiunge pickupInStore alle dependencies delle feature ADD_TO_CART_FEATURE e CART_BASE_FEATURE ("make pickup lib loaded before add-to-cart" / "before cart base"). Così, quando la feature add-to-cart o cart-base viene caricata, il chunk pickup è già caricato e i suoi provideOutlet (REPLACE) sono registrati.

Inoltre il root augmenta **a runtime** l'enum LAUNCH_CALLER con PICKUP_IN_STORE (dichiarazione TypeScript + assegnazione) (feature-libs/pickup-in-store/root/model/augmented-core.model.ts:L9-L15). I modelli principali: PickupOption = 'delivery' | 'pickup', AugmentedPointOfService = PointOfService & {pickupOption} (feature-libs/pickup-in-store/root/model/pickup-option.model.ts:L9-L15); StockLocationSearchParams = {productCode} & ({latitude, longitude} | {location}) (feature-libs/pickup-in-store/root/model/stock-location-search-params.model.ts:L7-L26). La chiave localStorage del negozio preferito è preferred_store (feature-libs/pickup-in-store/root/pickup-in-store-constants.ts:L7-L8).

Quattro facade lazy async: true sulla feature pickupInStoreCore (feature-libs/pickup-in-store/root/facade/intended-pickup-location.facade.ts:L19-L34, feature-libs/pickup-in-store/root/facade/pickup-locations-search.facade.ts:L22-L45, feature-libs/pickup-in-store/root/facade/pickup-option.facade.ts:L13-L27, feature-libs/pickup-in-store/root/facade/preferred-store.facade.ts:L16-L30), legate ai service via useExisting (feature-libs/pickup-in-store/core/facade/facade-providers.ts:L19-L37). PickupInStoreCoreModule fornisce config, StockConnector, PickupLocationConnector, facade providers e lo store module (feature-libs/pickup-in-store/core/pickup-in-store-core.module.ts:L14-L23).

3.7 Pickup in store — stato NgRx

Tre feature NgRx: pickup-locations, pickup-option, stock (feature-libs/pickup-in-store/core/store/pickup-in-store-store.module.ts:L29-L53).

FEATURE	FORMA	RIFERIMENTO
pickup-locations	{defaultPointOfService, intendedPickupLocations (per productCode), storeDetails (per nome)}	feature-libs/pickup-in-store/core/store/pickup-location-state.ts:L19-L27
pickup-option	{pageContext: string, pickupOption: {entryNumber, pickupOption}[]}	feature-libs/pickup-in-store/core/store/pickup-option-state.ts:L15-L23
stock	{browserLocation, hideOutOfStock, stockLevel: LoaderState<Record<productCode, …>>, stockLevelAtStore}	feature-libs/pickup-in-store/core/store/stock-state.ts:L21-L28

Punti salienti:

- intendedPickupLocationsReducer: AddLocation sostituisce, RemoveLocation scrive {pickupOption: 'delivery'} invece di cancellare, SetPickupOption fonde il pickupOption (feature-libs/pickup-in-store/core/store/reducers/pickup-locations/pickup-locations.reducer.ts:L14-L31).
- pickupOptionReducer: RemovePickupOption rimuove e decrementa gli entryNumber successivi, rispecchiando la rinumerazione delle entry del carrello (feature-libs/pickup-in-store/core/store/reducers/

pickup-option/pickup-option.reducer.ts:L13-L42).

- Il meta reducer `clearStockState`, su `CLEAR_STOCK_DATA`, passa `undefined` al reducer, che quindi riporta tutta la feature `stock` allo stato iniziale (feature-libs/pickup-in-store/core/store/reducers/stock/index.ts:L42-L54).
- `stockAtStoreReducer` scrive `[productCode]: { [storeName]: stockLevel }`, sostituendo l'intero oggetto del prodotto (feature-libs/pickup-in-store/core/store/reducers/stock/stock-level.reducer.ts:L45-L51).
- `getPickupOptionByProductCode` restituisce `'delivery'` come default (feature-libs/pickup-in-store/core/store/selectors/pickup-locations.selectors.ts:L48-L55); `getStoresWithStockForProductCode` filtra con `storeHasStock` solo se `hideOutOfStock` è true (feature-libs/pickup-in-store/core/store/selectors/stock.selectors.ts:L61-L71); `isInStock` considera non disponibili sia `outOfStock` sia `lowStock` (feature-libs/pickup-in-store/core/utils/utils.ts:L9-L19).

Effetti:

- `StockEffect.loadStockLevels$` (`switchMap`): separa `productCode` dal resto dei parametri, chiama `GET products/${productCode}/stock` con `location` o `latitude/longitude` + `fields: 'FULL'`, errore → `StockLevelFail` (feature-libs/pickup-in-store/core/store/effects/stock.effect.ts:L26-L49, feature-libs/pickup-in-store/occ/adapters/occ-stock.adapter.ts:L35-L51).
- `loadStockLevelAtStore$` (`concatMap`) **senza** `catchError` (feature-libs/pickup-in-store/core/store/effects/stock.effect.ts:L51-L65).
- `PickupLocationEffect.storeDetails$` (`mergeMap`) → `SetStoreDetailsSuccess`/`SetStoreDetailsFailure` (feature-libs/pickup-in-store/core/store/effects/pickup-location.effect.ts:L26-L51); `SetStoreDetailsFailure` è un wrapper che aggiunge all'azione la proprietà `error` (uguale a `payload`) per farsi riconoscere da `CxErrorHandlerEffect` (feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L83-L102).
- `loadDefaultPointOfService$`: se il profilo utente ha `defaultPointOfServiceName` usa quello (con `displayName: ''`), altrimenti legge `preferred_store` dal localStorage; in errore emette uno store vuoto (feature-libs/pickup-in-store/core/store/effects/default-point-of-service-name.effect.ts:L39-L81).
- `setDefaultPointOfService$`: salva in localStorage, aggiorna il profilo (`defaultPointOfServiceName`), e termina sempre con `LoadDefaultPointOfService()` (feature-libs/pickup-in-store/core/store/effects/default-point-of-service-name.effect.ts:L83-L107).

3.8 Pickup in store — flussi UI

Outlet registrati (feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.module.ts:L23-L27, feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.module.ts:L23-L27, feature-libs/pickup-in-store/components/container/pickup-info-container/pickup-info-container.module.ts:L19-L23, feature-libs/pickup-in-store/components/container/pickup-in-store-order-consignment/pickup-in-store-order-consignment-container.module.ts:L22-L26, feature-libs/pickup-in-store/components/container/set-preferred-store/set-preferred-store.module.ts:L28-L32):

OUTLET	POSIZIONE	COMPONENTE
`CartOutlets.ITEM_DELIVERY_DETAILS`	REPLACE	`CartPickupOptionsContainerComponent`
`CartOutlets.ADD_TO_CART_PICKUP_OPTION`	REPLACE	`PdpPickupOptionsContainerComponent`
`CartOutlets.PICKUP_INFO`	REPLACE	`PickupInfoContainerComponent`

OUTLET	POSIZIONE	COMPONENTE
OrderOutlets.ORDER_CONSIGNMENT	AFTER	container consignment
StoreFinderOutlets.PREFERRED_STORE	REPLACE	SetPreferredStoreComponent

Il dialog è registrato nella layout config: PICKUP_IN_STORE → PickupOptionDialogComponent, inlineRoot: true, DIALOG_TYPE.DIALOG (feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog-layout.config.ts:L10-L18).

PDP + add to cart. AddToCartComponent rende l'outlet ADD_TO_CART_PICKUP_OPTION con cxComponentRef e, al click, legge intendedPickupLocation$ dell'istanza del componente PDP: se pickupOption === 'pickup' passa il nome del negozio come pickupStore ad addEntry (feature-libs/cart/base/components/add-to-cart/add-to-cart.component.html:L23-L28, feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L267-L287). L'add to cart viene disabilitato se l'opzione è pickup ma manca displayName (feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L321-L335). Il container PDP imposta il pageContext 'PDP' e, se non c'è un'intended location con displayName, propone il negozio preferito con stock come pickupOption: 'delivery' (feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.component.ts:L85-L173). Sul PDP il cambio opzione aggiorna solo lo stato intended (feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.component.ts:L188-L196).

Dialog. Entrambi i container aprono il dialog con LAUNCH_CALLER.PICKUP_IN_STORE passando productCode (e entryNumber/quantity nel carrello) (feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L295-L310). Nel dialog la ricerca chiama startSearch (→ StockLevel) (feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog.component.ts:L140-L145); StoreListComponent.onSelectStore toglie stockInfo e imposta l'intended location con pickupOption: 'pickup' (feature-libs/pickup-in-store/components/container/store-list/store-list.component.ts:L67-L76). La chiusura con selezione, **solo fuori dal PDP**, aggiorna l'entry con updateEntry(entryNumber, quantity, name, false) (feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog.component.ts:L171-L192). Click sull'host chiude senza selezione (feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog.component.ts:L93-L100).

Carrello. Il container carrello imposta il pageContext dall'id della pagina CMS e l'opzione corrente dall'esistenza di deliveryPointOfService sull'entry (feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L128-L163). Cambio opzione: delivery → updateEntry(entry, qty, undefined, true); pickup → updateEntry(entry, qty, name, true) con il nome da storeDetails$ (feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L255-L289). Nell'adapter OCC del carrello pickupStore === undefined && pickupToDelivery produce un PUT con sola quantity; altrimenti un PATCH con deliveryPointOfService (feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L86-L104). Il container è nascosto per entry di un ordine e per CartType.SELECTIVE (feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.html:L1-L7), e i controlli sono disabilitati se lo stesso prodotto compare in più entry (feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L165-L177).

Riepiloghi. Le tre CMS component di riepilogo (CheckoutReviewPickup, OrderConfirmationPickUpComponent, PickupInStoreDeliveryModeComponent) sono tutte PickUpItemsDetailsComponent con data diversa (feature-libs/pickup-in-store/components/container/pickup-items-details/pickup-items-details.module.ts:L33-L57); il componente sceglie ordine o carrello in base a data.context === 'order' (feature-libs/pickup-in-store/components/container/pickup-items-details/pickup-items-details.component.ts:L58-L72).

`DeliveryPointsService` raggruppa le entry pickup per negozio, carica i dettagli di ogni negozio e combina i risultati (feature-libs/pickup-in-store/components/services/delivery-points.service.ts:L66-L160).

Negozio preferito. `SetPreferredStoreComponent` riceve i nomi dal contesto outlet e chiama `setPreferredStore` (feature-libs/pickup-in-store/components/container/set-preferred-store/set-preferred-store.component.ts:L42-L57). `PreferredStoreService` dispatcha `LoadDefaultPointOfService()` nel costruttore (feature-libs/pickup-in-store/core/services/preferred-store.service.ts:L32-L41); `getPreferredStoreWithProductInStock` richiede lo stock al negozio preferito ed emette solo se `isInStock` (feature-libs/pickup-in-store/core/services/preferred-store.service.ts:L73-L97). `MyPreferredStoreComponent` carica i dettagli del preferito e sulla pagina `storefinderPage` mostra solo l'azione "Get Directions" (feature-libs/pickup-in-store/components/container/my-preferred-store/my-preferred-store.component.ts:L100-L164).

Nella demo il pickup-in-store è aggiunto solo nel ramo non-B2B (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L117-L129).

3.9 Packaging: schematics, stili, test

Schematics `ng-add`**.** Le quattro factory sono identiche salvo il nome: validano l'installazione Spartacus da `package.json`, espandono le feature richieste con `analyzeCrossFeatureDependencies` e concatenano `analyzeApplication` → `addFeatures` → `addPackageJsonDependenciesForLibrary(peerDependencies)` → `finalizeInstallation` (feature-libs/estimated-delivery-date/schematics/add-estimated-delivery-date/index.ts:L25-L45, feature-libs/pickup-in-store/schematics/add-pickup-in-store/index.ts:L25-L45, feature-libs/requested-delivery-date/schematics/add-requested-delivery-date/index.ts:L25-L45, feature-libs/storefinder/schematics/add-storefinder/index.ts:L25-L45). Il test EDD verifica che il modulo lazy finisca nel wrapper `cart-base` e che **non** vengano creati i feature module user/order (feature-libs/estimated-delivery-date/schematics/add-estimated-delivery-date/index_spec.ts:L110-L126), coerente con la T10 (EDD viaggia col chunk del carrello).

Stili globali vs allowlist. Pickup-in-store e storefinder usano placeholder + allowlist (vedi sezione 8), mentre RDD ed EDD scrivono regole dirette sui selettori: `$skipComponentStyles` non ha effetto su di esse (feature-libs/requested-delivery-date/styles/_requested-delivery-date.scss:L1-L25, feature-libs/estimated-delivery-date/styles/_estimated-delivery-date.scss:L1-L23). In particolare RDD azzera il `padding-bottom` delle card dentro `.cx-review-card-shipping`, classe del riepilogo checkout definita fuori dalla lib (feature-libs/requested-delivery-date/styles/_requested-delivery-date.scss:L15-L25, feature-libs/checkout/base/components/checkout-review-submit/checkout-review-submit.component.html:L58-L58): importare gli stili RDD cambia la review anche quando il date picker non è presente. Il root `_index.scss` del pickup assegna `$cx-color-link-primary: #1672b7` senza `!default` prima degli import, quindi un valore impostato prima dall'app viene sovrascritto (feature-libs/pickup-in-store/_index.scss:L1-L7, feature-libs/pickup-in-store/styles/_store.scss:L29-L29).

Test. Le tre config Vitest condividono forks/jsdom/`setup-vitest.ts` e soglie 85/85/65/85; storefinder aggiunge alias di path per gli helper di test importati con path relativi alla root (feature-libs/storefinder/vitest.config.ts:L16-L22, feature-libs/storefinder/vitest.config.ts:L51-L56, feature-libs/requested-delivery-date/vitest.config.ts:L42-L47, feature-libs/estimated-delivery-date/vitest.config.ts:L41-L46).

4. Diagrammi

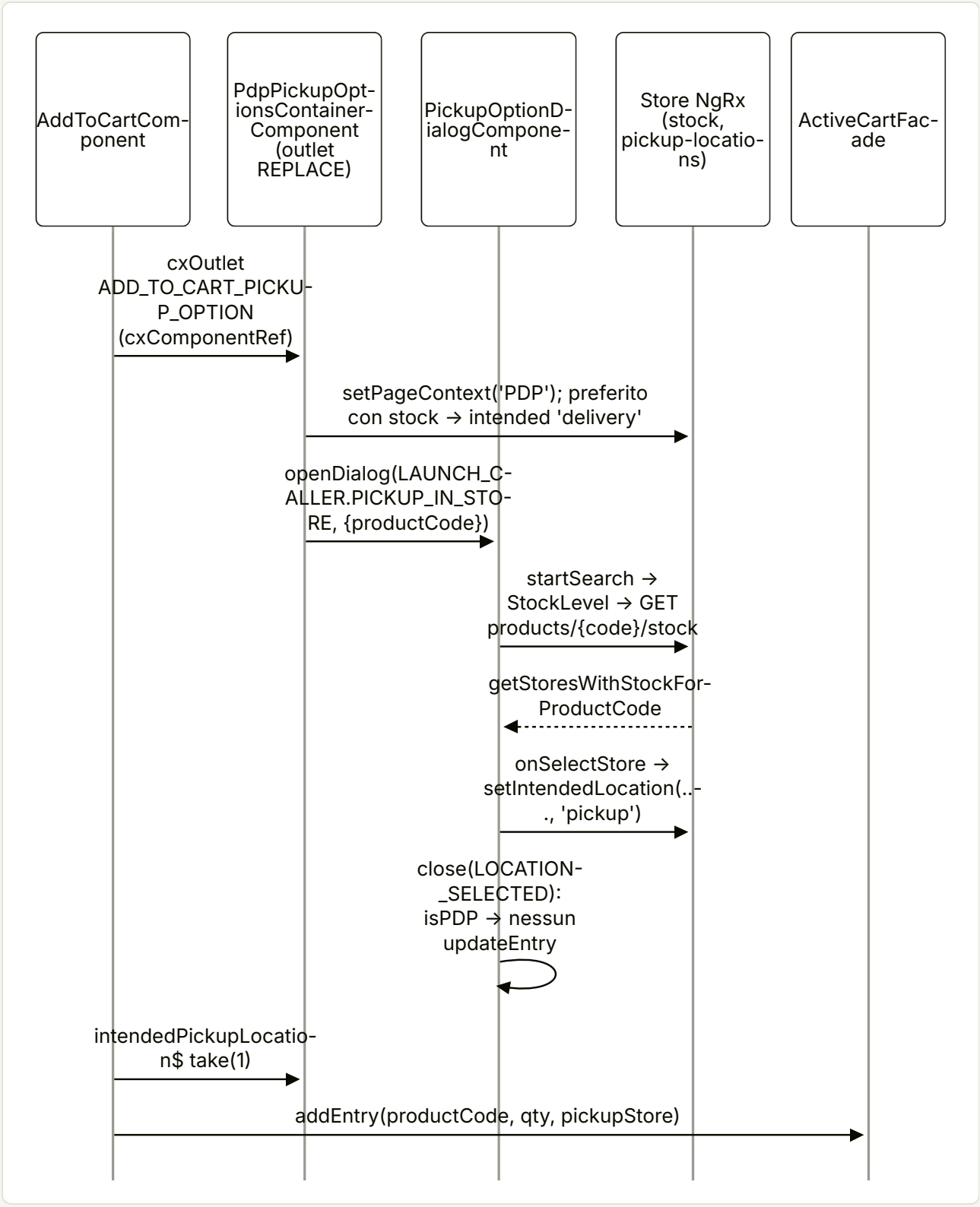
4.1 RDD: dal delivery mode al backend

■ TAVOLA 32.1 diagramma a tutta pagina, in fondo al capitolo

4.2 Store finder: ricerca e cambio contesto

■ TAVOLA 32.2 diagramma a tutta pagina, in fondo al capitolo

4.3 Pickup: dal PDP al carrello



5. Estratti di codice reale

5.1 RDD: outlet eager e CMS lazy

feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L18-L50

```ts
export function defaultRequestedDeliveryDateComponentsConfig(): CmsConfig {
  const config: CmsConfig = {
    featureModules: {
      [REQUESTED_DELIVERY_DATE_FEATURE]: {
        cmsComponents: [
          'DeliveryModeDatePickerComponent',
          'OrderOverviewDeliveryDateComponent',
        ],
      },
      // by default core is bundled together with components
      [REQUESTED_DELIVERY_DATE_CORE_FEATURE]: REQUESTED_DELIVERY_DATE_FEATURE,
    },
  };

  return config;
}

@NgModule({
  providers: [
    provideOutlet({
      id: CartOutlets.DELIVERY_MODE,
      position: OutletPosition.AFTER,
      component: DeliveryModeDatePickerComponent,
    }),
    provideOutlet({
      id: CartOutlets.ORDER_OVERVIEW,
      position: OutletPosition.AFTER,
      component: OrderOverviewDeliveryDateComponent,
    }),
    provideDefaultConfigFactory(defaultRequestedDeliveryDateComponentsConfig),
  ],
})
export class RequestedDeliveryDateRootModule {}
```

5.2 RDD: adapter OCC

feature-libs/requested-delivery-date/occ/adapters/occ-requested-delivery-date.adapter.ts:L25-L44

```ts
  setRequestedDeliveryDate(
    userId: string,
    cartId: string,
```

```
    requestedRetrievalAt: string
  ) {
    let headers = new HttpHeaders({
      'Content-Type': 'application/x-www-form-urlencoded',
    });
    headers = InterceptorUtil.createHeader(USE_CLIENT_TOKEN, true, headers);

    const url = this.occEndpoints.buildUrl('requestedDeliveryDate', {
      urlParams: {
        userId,
        cartId,
      },
      queryParams: { requestedRetrievalAt },
    });

    return this.http.put(url, { headers });
  }
```

5.3 RDD: scrittura automatica all'init

feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L74-L99

TS

```
  ngOnInit(): void {
    if (this.deliveryOutlet?.context$) {
      this.subscription.add(
        this.deliveryOutlet.context$.subscribe((context) => {
          this.cartEntry = context?.item;
          this.isDatePickerReadOnly = context?.readonly || false;
        })
      );
    }

    if (this.isEarliestRetrievalDatePresent()) {
      this.earliestRetrievalAt = this.cartEntry.earliestRetrievalAt;
    }
    if (this.isRequestedDeliveryDatePresent()) {
      this.requestedRetrievalAt = this.cartEntry.requestedRetrievalAt;
    } else {
      //set the value of requestedRetrievalAt as earliestRetrievalAt and update occ.
      this.requestedRetrievalAt = this.earliestRetrievalAt;
      this.form.patchValue({
        requestDeliveryDate: this.requestedRetrievalAt,
      });
      this.setRequestedDeliveryDate();
    }
```

```
    this.form.patchValue({
      requestDeliveryDate: this.requestedRetrievalAt,
    });
```

5.4 RDD: handler degli errori 400

feature-libs/requested-delivery-date/core/http-interceptors/bad-request/requested-delivery-date-badrequest.handler.ts:L20-L48

```ts
export class RequestedDeliveryDateBadRequestHandler extends HttpErrorHandler {
  responseStatus = HttpResponseStatus.BAD_REQUEST;

  hasMatch(errorResponse: HttpErrorResponse): boolean {
    return (
      super.hasMatch(errorResponse) && this.getErrors(errorResponse)?.length > 0
    );
  }
  // ...
  protected getErrors(response: HttpErrorResponse): ErrorModel[] {
    return (response.error?.errors ?? []).filter(
      (error: any) =>
        error?.type === 'ValidationError' &&
        error?.message === 'checkout.multi.requestedretrievaldatevalid.error'
    );
  }

  getPriority(): Priority {
    return Priority.NORMAL;
  }
```

5.5 EDD: componente

feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L25-L37

```ts
export class EstimatedDeliveryDateComponent {
  @Optional() protected cartItemContext = inject(CartItemContext);
  protected orderHistoryFacade = inject(OrderHistoryFacade);
  protected translationService = inject(TranslationService);
  protected datePipe = inject(CxDatePipe);

  readonly orderEntry$: Observable<OrderEntry> =
    this.cartItemContext?.item$ ?? EMPTY;

  readonly consignments$: Observable<Consignment> = this.orderHistoryFacade
```

```
    .getOrderDetails()
    .pipe(map((order: Order) => order?.consignments ?? []))
    .pipe(switchMap((consignments: Consignment[]) => consignments));
```

5.6 Store finder: child routes CMS

feature-libs/storefinder/components/store-finder-components.module.ts:L67-L99

```ts
    provideDefaultConfig({
      cmsComponents: {
        StoreFinderComponent: {
          component: StoreFinderComponent,
          childRoutes: [
            {
              path: 'find',
              component: StoreFinderSearchResultComponent,
            },
            {
              path: 'view-all',
              component: StoreFinderStoresCountComponent,
            },
            {
              path: 'country/:country',
              component: StoreFinderGridComponent,
            },
            // ...
            {
              path: 'country/:country/:store',
              component: StoreFinderStoreComponent,
            },
          ],
        },
      },
    }),
```

5.7 Store finder: facade senza i metodi deprecati

feature-libs/storefinder/root/facade/store-finder.facade.ts:L21-L42

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      // @ts-ignore: Deprecated methods will be removed with next major release
      facade: StoreFinderFacade,
      feature: STORE_FINDER_FEATURE,
      methods: [
```

```
        'getStoresLoading',
        'getStoresLoaded',
        'getFindStoresEntities',
        'getViewAllStoresLoading',
        'getViewAllStoresEntities',
        'findStoresAction',
        'viewAllStores',
        'viewStoreById',
        'callFindStoresAction',
        'getFindStoreEntityById',
      ],
      async: true,
    }),
})
```

5.8 Store finder: due loader sullo stesso entity type + meta reducer

feature-libs/storefinder/core/store/reducers/index.ts:L21-L51

```ts
export function getReducers(): ActionReducerMap<StoresState> {
  return {
    findStores: StateUtils.loaderReducer(STORE_FINDER_DATA, findStoresReducer),
    viewAllStores: StateUtils.loaderReducer(
      STORE_FINDER_DATA,
      viewAllStoresReducer
    ),
  };
}
// ...
export function clearStoreFinderState(
  reducer: ActionReducer<StoresState, Action>
): ActionReducer<StoresState, Action> {
  return function (state, action) {
    if (action.type === SiteContextActions.LANGUAGE_CHANGE) {
      state = undefined;
    }
    if (action.type === StoreFinderActions.CLEAR_STORE_FINDER_DATA) {
      state = undefined;
    }
    return reducer(state, action);
  };
}
```

5.9 Store finder: parametri OCC

feature-libs/storefinder/occ/adapters/occ-store-finder.adapter.ts:L77-L102

```ts
const params: any = {};

if (longitudeLatitude) {
  params['longitude'] = String(longitudeLatitude.longitude);
  params['latitude'] = String(longitudeLatitude.latitude);
} else {
  params['query'] = query;
}

if (radius) {
  params['radius'] = String(radius);
}

if (searchConfig?.pageSize) {
  params['pageSize'] = String(searchConfig.pageSize);
}
if (searchConfig?.currentPage) {
  params['currentPage'] = String(searchConfig.currentPage);
}
if (searchConfig?.sort) {
  params['sort'] = searchConfig.sort;
}

return this.http.get<Occ.StoreFinderSearchPage>(
  this.occEndpointsService.buildUrl('stores', { queryParams: params })
);
```

5.10 Google Maps: chiave di sviluppo e toggle

feature-libs/storefinder/core/service/google-map-renderer.service.ts:L82-L108

```ts
const apiKey =
  this.config.googleMaps?.apiKey === GOOGLE_MAPS_DEVELOPMENT_KEY_CONFIG
    ? ''
    : this.config.googleMaps?.apiKey;

const useAsyncLoading = this.featureToggles.useGoogleMapsAsyncLoading;

// With `loading=async`, Google invokes a global `callback` function once
// the API is ready instead of firing the script `load` event.
const callbackName = useAsyncLoading
  ? this.registerMapCallback(mapElement, locations, selectMarkerHandler)
  : undefined;

this.scriptLoader.embedScript({
  src: this.config.googleMaps?.apiUrl ?? '',
  params: {
    key: apiKey,
```

```
        // Opt into async loading to avoid Google's "loaded directly without
        // loading=async" performance warning.
        ...(useAsyncLoading
          ? { loading: 'async', callback: callbackName }
          : {}),
        // Advanced markers require the optional `marker` library.
        ...(this.featureToggles.useAdvancedGoogleMarkers
          ? { libraries: 'marker' }
          : {}),
```

5.11 Pickup: dipendenze di feature

feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L39-L60

```ts
@NgModule({
  providers: [
    provideDefaultConfigFactory(defaultPickupInStoreComponentsConfig),
    // make pickup lib loaded before add-to-cart
    provideDefaultConfig({
      featureModules: {
        [ADD_TO_CART_FEATURE]: {
          dependencies: [PICKUP_IN_STORE_FEATURE],
        },
      },
    }),
    // make pickup lib loaded before cart base
    provideDefaultConfig({
      featureModules: {
        [CART_BASE_FEATURE]: {
          dependencies: [PICKUP_IN_STORE_FEATURE],
        },
      },
    }),
  ],
})
export class PickupInStoreRootModule {}
```

5.12 Pickup: effetto stock per negozio senza `catchError`

feature-libs/pickup-in-store/core/store/effects/stock.effect.ts:L51-L65

```ts
  loadStockLevelAtStore$ = createEffect(() =>
    this.actions$.pipe(
      ofType(StockLevelActions.STOCK_LEVEL_AT_STORE),
      map(({ payload }: StockLevelActions.StockLevelAtStoreAction) => payload),
      concatMap(({ productCode, storeName }) =>
        this.stockConnector.loadStockLevelAtStore(productCode, storeName).pipe(
```

```
          map((stockLevel) =>
            StockLevelActions.StockLevelAtStoreSuccess({
              payload: { productCode, storeName, stockLevel },
            })
          )
        )
      )
    )
  );
```

5.13 Pickup ↔ delivery nell'adapter del carrello

feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L86-L104

```ts
    // switch from pickup to delivery mode
    if (pickupStore === undefined && pickupToDelivery) {
      return this.http
        .put<CartModification>(url, { quantity: qty }, { headers })
        .pipe(this.converterService.pipeable(CART_MODIFICATION_NORMALIZER));
    }

    let params = {};
    if (pickupStore) {
      params = {
        deliveryPointOfService: {
          name: pickupStore,
        },
      };
    }

    return this.http
      .patch<CartModification>(url, { quantity: qty, ...params }, { headers })
      .pipe(this.converterService.pipeable(CART_MODIFICATION_NORMALIZER));
```

5.14 Pickup: negozio preferito (bug mascherato)

feature-libs/pickup-in-store/core/store/effects/default-point-of-service-name.effect.ts:L93-L106

```ts
      switchMap((preferredStore: PointOfServiceNames) =>
        this.userProfileService
          .update({
            defaultPointOfServiceName: preferredStore.name,
          })
          .pipe(
            map(() => DefaultPointOfServiceActions.LoadDefaultPointOfService()),
            catchError((_error) =>
              of(DefaultPointOfServiceActions.LoadDefaultPointOfService)
```

```
          )
        )
      ),
      map(() => DefaultPointOfServiceActions.LoadDefaultPointOfService())
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
googleMaps.apiUrl	string	https://maps.googleapis.com/maps/api/js	feature-libs/storefinder/core/config/default-store-finder-config.ts:L11-L11	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L96-L96
googleMaps.apiKey	string	'' ('cx-development' = senza chiave)	feature-libs/storefinder/core/config/default-store-finder-config.ts:L12-L12	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L49-L49, feature-libs/storefinder/core/service/google-map-renderer.service.ts:L82-L85
googleMaps.scale	number	5	feature-libs/storefinder/core/config/default-store-finder-config.ts:L13-L13	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L184-L184
googleMaps.selectedMarkerScale	number	17	feature-libs/storefinder/core/config/default-store-finder-config.ts:L14-L14	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L151-L154
googleMaps.radius	number	50000	feature-libs/storefinder/core/config/default-store-finder-config.ts:L15-L15	feature-libs/storefinder/components/store-finder-search-result/store-finder-search-result.component.ts:L80-L80
googleMaps.mapId	string	'DEMO_MAP_ID'	feature-libs/storefinder/core/config/default-store-finder-config.ts:L19-L19	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L188-L190

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
toggle `useAdvancedGoogleMarkers`	boolean	`false`	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1014-L1014	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L105-L105, feature-libs/storefinder/core/service/google-map-renderer.service.ts:L204-L204
toggle `useGoogleMapsAsyncLoading`	boolean	`false`	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1015-L1015	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L87-L87
toggle `a11yStoreFinderLabel`	boolean	`true`	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1019-L1019	feature-libs/storefinder/components/store-finder-search/store-finder-search.component.html:L2-L2
toggle `a11yStoreFinderFocusOnBackButton`	boolean	`false`	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1020-L1020	feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L77-L77
`pickupInStore.consentTemplateId`	string	`'STORE_USER_INFORMATION'`	feature-libs/pickup-in-store/core/config/default-pickup-in-store-config.ts:L10-L12	nessun lettore trovato (grep)
endpoint `requestedDeliveryDate`	string	`users/${userId}/carts/${cartId}/requestedretrievaldate`	feature-libs/requested-delivery-date/occ/config/default-occ-requested-delivery-date-config.ts:L13-L14	feature-libs/requested-delivery-date/occ/adapters/occ-requested-delivery-date.adapter.ts:L35-L41

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
endpoint `store` / `stores` / `storescounts`	string	`stores/${storeId}?fields=FULL`, `stores?fields=…`, `stores/storescounts`	feature-libs/storefinder/occ/adapters/default-occ-store-finder-config.ts:L13-L16	feature-libs/storefinder/occ/adapters/occ-store-finder.adapter.ts:L53-L66, feature-libs/storefinder/occ/adapters/occ-store-finder.adapter.ts:L100-L102
endpoint `storeDetails`	string	`stores/${storeName}`	feature-libs/pickup-in-store/occ/adapters/default-occ-pickup-location-config.ts:L16-L16	feature-libs/pickup-in-store/occ/adapters/occ-pickup-location.adapter.ts:L33-L36
endpoint `stock` / `stockAtStore`	string	`products/${productCode}/stock`, `products/${productCode}/stock/${storeName}`	feature-libs/pickup-in-store/occ/adapters/default-occ-stock-config.ts:L16-L17	feature-libs/pickup-in-store/occ/adapters/occ-stock.adapter.ts:L41-L44, feature-libs/pickup-in-store/occ/adapters/occ-stock.adapter.ts:L59-L61
endpoint `carts` / `cart` / `createCart` (EDD)	string	come cart lib + `arrivalSlots`	feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L10-L16	adapter OCC del cart lib (merge: vedi 3.2)
`featureModules[ADD_TO_CART_FEATURE / CART_BASE_FEATURE].dependencies`	string[]	`[pickupInStore]`	feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L42-L57	feature modules core (capitolo 12)
localStorage `preferred_store`	JSON `PointOfServiceNames`	—	feature-libs/pickup-in-store/root/pickup-in-store-constants.ts:L8-L8	feature-libs/pickup-in-store/core/store/effects/default-point-of-service-name.effect.ts:L53-L58, feature-libs/pickup-in-store/core/store/effects/default-point-of-service-name.effect.ts:L87-L91
Layout `StoreFinderPageTemplate`	slots	`MiddleContent`, `SideContent`	feature-libs/storefinder/root/config/default-store-finder-layout-config.ts:L11-L13	layout engine (capitolo 18)

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
launch.PICKUP_IN_STORE	LayoutConfig	PickupOptionDialogComponent, inline, dialog	feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog-layout.config.ts:L10-L18	LaunchDialogService

Chunk i18n: `requestedDeliveryDate: ['requestedDeliveryDate']` (feature-libs/requested-delivery-date/assets/translations/translations.ts:L9-L12), `storeFinder: ['storeFinder']` (feature-libs/storefinder/assets/translations/translations.ts:L9-L11), 13 chiavi sotto `pickupInStore` (feature-libs/pickup-in-store/assets/translations/translations.ts:L9-L25).

Esempio (solo illustrativo, simboli esportati da `@spartacus/storefinder/root` e `@spartacus/core`): come la demo, per abilitare la mappa in sviluppo.

```ts
// projects/storefrontapp/src/app/private/private.providers.ts:L46-L49 (estratto reale)
      // Using special key value 'cx-development' allows google maps to display
      // without a key, for development or demo purposes:
      googleMaps: { apiKey: GOOGLE_MAPS_DEVELOPMENT_KEY_CONFIG },
```

7. Commenti e note notevoli dal sorgente

- `// by default core is bundled together with components` — feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L27-L27. Lo stesso schema (core → feature principale) vale per pickup (feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L34-L34).
- `//Make components available without being CMS driven` — feature-libs/requested-delivery-date/root/public_api.ts:L7-L7: spiega perché i componenti RDD stanno nell'entry eager.
- `//set the value of requestedRetrievalAt as earliestRetrievalAt and update occ.` — feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L90-L90: la scrittura all'init è voluta.
- JSDoc `Date string in the format 'dd-mm-yyy'` (sic) — feature-libs/requested-delivery-date/root/components/shared/date-validation.service.ts:L13-L16.
- `// TODO: Inline this factory when we start releasing Ivy compiled libraries` — feature-libs/storefinder/root/store-finder-root.module.ts:L17-L17.
- `// @ts-ignore: Deprecated methods will be removed with next major release` — feature-libs/storefinder/root/facade/store-finder.facade.ts:L25-L25.
- Il JSDoc di `StoreFinderFacade` ("Store the Point of Service a user wants to collect a product from before it is added to the cart.") è copiato da `IntendedPickupLocationFacade` — feature-libs/storefinder/root/facade/store-finder.facade.ts:L18-L20, feature-libs/pickup-in-store/root/facade/intended-pickup-location.facade.ts:L16-L18.
- `@deprecated Please use StoreLocationService instead,` (×3 facade, ×3 service) — feature-libs/storefinder/root/facade/store-finder.facade.ts:L60-L71, feature-libs/storefinder/core/facade/store-finder.service.ts:L230-L253.

- CXSPA-4871: The return value of this method signature is wrong, should be StoreFinderSearchPage. — feature-libs/storefinder/core/facade/store-finder.service.ts:L59-L59.
- Commenti su advanced markers, glyphText non nei types pinnati, gmp-click — feature-libs/storefinder/core/service/google-map-renderer.service.ts:L213-L245.
- // Fires when backButton enters the DOM (signal goes undefined → ElementRef), moving focus to it. — feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L78-L78.
- // TODO jsdoc / // TODO split this service — feature-libs/pickup-in-store/root/facade/pickup-locations-search.facade.ts:L18-L20.
- Commento su SetStoreDetailsFailure e CxErrorHandlerEffect ("The error property was added later to the return type, to be recognizable by CxErrorHandlerEffect") — feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L83-L87.
- JSDoc di StockLevelAtStore copiato: "Add a proposed pickup location for a product code." — feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L84-L86.
- // make pickup lib loaded before add-to-cart / before cart base — feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L42-L50.
- Lungo commento esplicativo su deliveryPointsOfService$ — feature-libs/pickup-in-store/components/services/delivery-points.service.ts:L32-L47.
- // Return false to stop \onPickupOptionChange` being called after this–feature-libs/pickup-in-store/components/presentational/pickup-options/pickup-options.component.ts:L134-L134`.
- /* istanbul ignore else */ posizionato prima di una chiamata non condizionale — feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L259-L259.

8. Extension point

SIMBOLO	TIPO	DEFINIZIONE	PUBLIC API	RICETTA
RequestedDeliveryDateFacade	facade astratto	feature-libs/requested-delivery-date/root/facade/requested-delivery-date.facade.ts:L24-L33	@spartacus/requested-delivery-date/root	50 — override facade via DI
RequestedDeliveryDateAdapter	classe astratta	feature-libs/requested-delivery-date/core/connectors/requested-delivery-date.adapter.ts:L9-L18	@spartacus/requested-delivery-date/core	52 — adapter custom
RequestedDeliveryDateBadRequestHandler	HttpErrorHandler	feature-libs/requested-delivery-date/core/http-interceptors/bad-request/requested-delivery-date-badrequest.handler.ts:L17-L49	NOTA non esportato (feature-libs/requested-delivery-date/core/public_api.ts:L7-L9)	54 — API pubblica vs interna
DateValidationService	service root	feature-libs/requested-delivery-date/root/components/shared/date-validation.service.ts:L9-L52	non esportato (feature-libs/requested-delivery-date/root/components/index.ts:L7-L9)	54

SIMBOLO	TIPO	DEFINIZIONE	PUBLIC API	RICETTA
Outlet `CartOutlets.DELIVERY_MODE` / `ORDER_OVERVIEW` / `ITEM_DETAILS`	outlet	`feature-libs/cart/base/root/models/cart-outlets.model.ts:L13-L24`	`@spartacus/cart/base/root`	51 — outlet
`defaultOccCartWithEddConfig`	`OccConfig`	`feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L18-L28`	non esportato (`feature-libs/estimated-delivery-date/show-estimated-delivery-date/public_api.ts:L7-L8`)	52 — campi OCC aggiuntivi
`EstimatedDeliveryDateComponent`	componente	`feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L19-L74`	`@spartacus/estimated-delivery-date`	51 — sostituire componente in outlet
`StoreFinderFacade`	facade astratto	`feature-libs/storefinder/root/facade/store-finder.facade.ts:L43-L73`	`@spartacus/storefinder/root`	50
`StoreFinderAdapter`, `STORE_FINDER_SEARCH_PAGE_NORMALIZER`, `STORE_COUNT_NORMALIZER`	adapter / token	`feature-libs/storefinder/core/connectors/store-finder.adapter.ts:L11-L22`, `feature-libs/storefinder/core/connectors/converters.ts:L11-L17`	`@spartacus/storefinder/core`	52 — adapter e converter
`StoreFinderConfig` (`googleMaps`)	config astratta	`feature-libs/storefinder/core/config/store-finder-config.ts:L10-L27`	`@spartacus/storefinder/core`	50 — override config
`GoogleMapRendererService`, `StoreLocationService`	service root	`feature-libs/storefinder/core/service/google-map-renderer.service.ts:L44-L68`, `feature-libs/storefinder/core/service/store-location.service.ts:L10-L41`	`@spartacus/storefinder/core`	50 — override service
`StoreFinderActions`, `StoreFinderSelectors`	NgRx	`feature-libs/storefinder/core/store/actions/index.ts:L7-L8`, `feature-libs/storefinder/core/store/selectors/index.ts:L7-L8`	`@spartacus/storefinder/core` (effetti/reducer non esportati)	54
`StoreFinderOutlets.PREFERRED_STORE`	outlet	`feature-libs/storefinder/root/model/store-entities.model.ts:L9-L11`	`@spartacus/storefinder/root`	51 — outlet
`StoreFinderComponent` + `childRoutes`	componente CMS	`feature-libs/storefinder/components/store-finder-components.module.ts:L66-L99`	`@spartacus/storefinder/components`	51 — sostituire componente CMS, route custom
`AbstractStoreItemComponent`	directive base	`feature-libs/storefinder/components/abstract-store-item/abstract-store-item.component.ts:L11-L24`	`@spartacus/storefinder/components`	51 — estendere un componente
4 facade pickup	facade astratti	`feature-libs/pickup-in-store/root/facade/intended-pickup-location.facade.ts:L19-L34` e simili	`@spartacus/pickup-in-store/root`	50

SIMBOLO	TIPO	DEFINIZIONE	PUBLIC API	RICETTA
`StockAdapter`, `PickupLocationAdapter`	adapter	`feature-libs/pickup-in-store/occ/pickup-in-store-occ.module.ts:L21-L29` (binding)	`@spartacus/pickup-in-store/core`	52
NgRx pickup (azioni, effetti, reducer, selettori)	NgRx	`feature-libs/pickup-in-store/core/store/index.ts:L7-L14`	`@spartacus/pickup-in-store/core` (effetti e reducer inclusi)	54
`LAUNCH_CALLER.PICKUP_IN_STORE` + `defaultPickupOptionsDialogLayoutConfig`	launch dialog	`feature-libs/pickup-in-store/root/model/augmented-core.model.ts:L9-L15`, `feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog-layout.config.ts:L10-L18`	`/root`, `/components`	51 — layout config
`featureModules[...].dependencies`	config	`feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L39-L60`	config	53 — feature lib lazy
`DeliveryPointsService`, `CurrentLocationService`	service root	`feature-libs/pickup-in-store/components/services/delivery-points.service.ts:L66-L160`	**NOTA** non esportati da `feature-libs/pickup-in-store/components/public_api.ts:L7-L9`	54
Allowlist SCSS (`$skipComponentStyles`)	Sass	`feature-libs/pickup-in-store/styles/_index.scss:L12-L36`, `feature-libs/storefinder/styles/components/_index.scss:L13-L37`	stili	53 — stili
Traduzioni `requestedDeliveryDate`, `storeFinder`, `pickupInStore`	i18n chunk	`feature-libs/pickup-in-store/assets/translations/translations.ts:L9-L25`	`/assets`	53 — traduzioni

Note pratiche:

- **Aggiungere campi al carrello senza perdere EDD.** Se l'app ridefinisce `backend.occ.endpoints.cart` con `provideConfig`, la sua stringa sostituisce sia quella della cart lib sia quella EDD (`core-libs/core/src/config/services/configuration.service.ts:L92-L100`): va incluso `arrivalSlots` a mano (stringa di riferimento in `feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L10-L28`). Ricetta: 52.
- **Componenti in outlet**: RDD/EDD/pickup usano `provideOutlet`. Per sostituire, ad esempio, il date picker si può registrare un altro `provideOutlet` sullo stesso id con `OutletPosition.REPLACE` o AFTER, come descritto nel capitolo 17 e in 51.
- **Store finder con un altro provider di mappe**: `GoogleMapRendererService` è `providedIn: 'root'` ed esportato; può essere sostituito via DI (`{ provide: GoogleMapRendererService, useClass: … }`), ricetta 50. `StoreFinderMapComponent` chiama solo `renderMap` e `centerMap` (`feature-libs/storefinder/components/store-finder-map/store-finder-map.component.ts:L45-L49`).

9. Trappole, casi limite, comportamenti controintuitivi

Requested delivery date

- **T1 — header passati come body.** `this.http.put(url, { headers })` passa l'oggetto `{headers}` come **body** (il secondo parametro di `put` è `body`), quindi `Content-Type: application/x-www-form-urlencoded` e l'header `USE_CLIENT_TOKEN` non vengono applicati come opzioni; il valore utile viaggia comunque in query string (feature-libs/requested-delivery-date/occ/adapters/occ-requested-delivery-date.adapter.ts:L43-L43, angular:packages/common/http/src/client.ts:L4451-L4454). DA VERIFICARE il comportamento del backend OCC con un body JSON `{"headers":{…}}`: nessun backend disponibile.
- **T2 — contesto sincrono.** `ngOnInit` legge `cartEntry` subito dopo la `subscribe` a `context$`: funziona solo se l'outlet emette sincronamente (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L75-L88).
- **T3 — scrittura all'apertura.** Se il cart non ha `requestedRetrievalAt`, la sola visualizzazione dello step delivery mode genera un `PUT` (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L89-L96).
- **T4 — due gestori d'errore** con lo stesso messaggio: componente (`UnknownResourceError`) e `HttpErrorHandler` globale (`ValidationError`) (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L172-L176, feature-libs/requested-delivery-date/core/http-interceptors/bad-request/requested-delivery-date-badrequest.handler.ts:L38-L44).
- **T5 — formato data.** `DateValidationService` presume `dd-mm-yyyy`; stringhe ISO non matchano la regex e passano invariate a `new Date` (feature-libs/requested-delivery-date/root/components/shared/date-validation.service.ts:L33-L35).

Estimated delivery date

- **T6 — `@Optional()` inutile.** Il decoratore è su un campo inizializzato con `inject(CartItemContext)` senza `{optional: true}`: non rende opzionale l'iniezione, e il fallback `?? EMPTY` non scatta se il token manca (feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L26-L32).
- **T7 — `consignments$` al singolare.** È tipato `Observable<Consignment>` e il template lo usa come singola consignment; con più consignment emette in sequenza e il template mostra l'ultima (feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L34-L37, feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.html:L42-L52).
- **T8 — endpoint sostituiti interamente.** L'override EDD rimpiazza le stringhe `carts`/`cart`/`createCart` della cart lib; una personalizzazione applicativa le rimpiazza a sua volta (sezione 3.2; core-libs/core/src/config/services/configuration.service.ts:L76-L100).
- **T9 — data non localizzata.** Formato `M/d/yyyy` cablato nel template (feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.html:L21-L21).
- **T10 — non è lazy nella demo.** Con `CX_ESTIMATED_DELIVERY_DATE` attiva, EDD arriva col chunk del carrello; con la build di default (`false`) non è presente affatto. Il root module è vuoto (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L16-L18, projects/storefrontapp/src/environments/environment.ts:L39-L39).

Store finder

- **T11 — loader condiviso.** `findStores` e `viewAllStores` usano lo stesso entity type: un `FindStores` mette `loading: true` anche in `viewAllStores`, e un `ViewAllStoresSuccess` imposta `success: true` anche in `findStores` (il cui `value` resta quello calcolato da `findStoresReducer`, che ignora l'azione) (feature-libs/storefinder/core/store/reducers/index.ts:L21-L29, core-libs/core/src/state/utils/loader/loader.reducer.ts:L32-L60, core-libs/core/src/state/utils/loader/loader.reducer.ts:L80-L84).

- **T12 — due** `STORE_FINDER_FEATURE`. `'storeFinder'` (root, nome feature lazy) e `'stores'` (core, chiave NgRx) (feature-libs/storefinder/root/feature-name.ts:L7-L7, feature-libs/storefinder/core/store/store-finder-state.ts:L10-L10).
- **T13 — metodi deprecati assenti dal proxy.** `getStoreLatitude`/`getStoreLongitude`/`getDirections` non sono in `methods`: chiamati sul facade lazy non esistono (feature-libs/storefinder/root/facade/store-finder.facade.ts:L28-L39, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L120-L123). Il ramo legacy dei marker, `AbstractStoreItemComponent` e `StoreFinderListComponent` li usano ancora tramite `StoreFinderService` (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L255-L258, feature-libs/storefinder/components/abstract-store-item/abstract-store-item.component.ts:L17-L19, feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L89-L92).
- **T14 — array vuoto.** `renderMap` accede a `locations[0]` senza controllo: `Object.entries(undefined)` lancia (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L58-L60).
- **T15 — due** `#storeMap`. Il template della lista ha una mappa desktop e una mobile con lo stesso riferimento; `@ViewChild('storeMap')` ne prende una sola (feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.html:L59-L59, feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.html:L141-L141, feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L58-L59).
- **T16 — subscription sovrascritta.** `reloadStoreEntitiesOnContextChange` assegna `this.subscription =` sopra l'istanza del field initializer (feature-libs/storefinder/core/facade/store-finder.service.ts:L31-L31, feature-libs/storefinder/core/facade/store-finder.service.ts:L193-L193).
- **T17 — filtro per paese lato client** dopo una ricerca con `pageSize: -1` (feature-libs/storefinder/core/store/effects/find-stores.effect.ts:L40-L48, feature-libs/storefinder/core/facade/store-finder.service.ts:L215-L224).
- **T18 —** `currentPage: 0` **non inviato** perché falsy (feature-libs/storefinder/occ/adapters/occ-store-finder.adapter.ts:L93-L95).
- **T19 — SSR.** `reloadStoreEntitiesOnContextChange` gira solo in browser (feature-libs/storefinder/core/facade/store-finder.service.ts:L192-L192); il callback Google async non viene registrato senza `defaultView` (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L132-L140).
- **T20 — stile schedule.** Placeholder/allowlist `cx-store-finder-schedule`, ma il componente ha selector `cx-schedule` (feature-libs/storefinder/styles/components/_store-finder-schedule.scss:L1-L1, feature-libs/storefinder/components/schedule-component/schedule.component.ts:L16-L16).

Pickup in store

- **T21 — selector duplicato.** `PdpPickupOptionsContainerComponent` ha lo stesso selector `cx-cart-pickup-options-container` del container carrello (feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.component.ts:L57-L57, feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L79-L79).
- **T22 —** `pickupOption$` **assegnato in un** `tap`: esiste solo dopo che la subscription combinata ha emesso (feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.component.ts:L94-L98, feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.component.ts:L131-L145).
- **T23 —** `catchError` **che restituisce il creator.** In `setDefaultPointOfService$` il `catchError` emette `LoadDefaultPointOfService` non invocato; il `map` finale maschera il bug emettendo sempre `LoadDefaultPointOfService()` (feature-libs/pickup-in-store/core/store/effects/default-point-of-service-name.effect.ts:L100-L105).

- **T24 —** `tap` **no-op** che restituisce un oggetto (feature-libs/pickup-in-store/core/services/preferred-store.service.ts:L90-L93).
- **T25 —** `clearPreferredStore` rimuove solo la chiave localStorage, non lo stato NgRx né il profilo (feature-libs/pickup-in-store/core/services/preferred-store.service.ts:L64-L66).
- **T26 —** `lowStock` **= non disponibile**: "nascondi esauriti" nasconde anche i negozi con stock basso (feature-libs/pickup-in-store/core/utils/utils.ts:L9-L15).
- **T27 — un solo negozio per prodotto** in `stockLevelAtStore` (feature-libs/pickup-in-store/core/store/reducers/stock/stock-level.reducer.ts:L47-L50).
- **T28 — effetto senza** `catchError`**.** Un errore HTTP in `loadStockLevelAtStore$` arriva al default error handler di NgRx, che risottoscrive l'effetto fino a 10 volte e poi lo lascia terminare (feature-libs/pickup-in-store/core/store/effects/stock.effect.ts:L51-L65, deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L174-L184).
- **T29 —** `ClearStockData` **azzera tutto**, compresi `hideOutOfStock` e `browserLocation` (feature-libs/pickup-in-store/core/store/reducers/stock/index.ts:L42-L52).
- **T30 — codice morto.** Azioni definite ma non usate fuori dal file (`SET_PICKUP_OPTION_TO_DELIVERY*`, `SET_PICKUP_OPTION_TO_PICKUP_IN_STORE*`, `StockLevelOnHold`, …) e config `pickupInStore.consentTemplateId` senza lettori (feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L22-L33, feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L35-L41, feature-libs/pickup-in-store/core/config/default-pickup-in-store-config.ts:L10-L12).
- **T31 — geolocalizzazione negata** nel dialog: `useMyLocation` non passa una error callback, lo spinner resta attivo (feature-libs/pickup-in-store/components/container/store-search/store-search.component.ts:L49-L57, feature-libs/pickup-in-store/components/services/current-location.service.ts:L28-L38).
- **T32 — subscription accumulate.** Ogni `onPickupOptionChange('pickup')` aggiunge una subscription a `storeDetails$` attiva fino al destroy (possibili `updateEntry` ripetuti) (feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L270-L288); analogamente `MyPreferredStoreComponent` sottoscrive senza unsubscribe (feature-libs/pickup-in-store/components/container/my-preferred-store/my-preferred-store.component.ts:L118-L163).
- **T33 —** `take(1)` **dopo** `combineLatest` con `filter(!!details)` in `PickupInfoContainerComponent`: se un negozio non carica mai, non emette nulla (feature-libs/pickup-in-store/components/container/pickup-info-container/pickup-info-container.component.ts:L45-L66).
- **T34 — outlet** `PICKUP_INFO` **mai renderizzato.** Il pickup registra un REPLACE su `CartOutlets.PICKUP_INFO`, ma nessun template del repo lo rende con `cxOutlet` (grep): il container non compare a meno che l'app non aggiunga l'outlet (feature-libs/cart/base/root/models/cart-outlets.model.ts:L21-L21, feature-libs/pickup-in-store/components/container/pickup-info-container/pickup-info-container.module.ts:L19-L23).
- **T35 —** `RemoveLocation` non cancella la chiave ma la imposta a `{pickupOption: 'delivery'}` (feature-libs/pickup-in-store/core/store/reducers/pickup-locations/pickup-locations.reducer.ts:L20-L23).
- **T36 — profilo utente anonimo.** `loadDefaultPointOfService$` ricade sul localStorage se `userProfileService.get()` non ha `defaultPointOfServiceName` (feature-libs/pickup-in-store/core/store/effects/default-point-of-service-name.effect.ts:L42-L60). DA VERIFICARE cosa emetta `get()` per l'utente anonimo: non verificato in `@spartacus/user/profile`.

10. Come fare debug

1. **Redux DevTools**: cercare le chiavi `stores`, `pickup-locations`, `pickup-option`, `stock` (feature-libs/storefinder/core/store/store-finder-state.ts:L10-L10, feature-libs/pickup-in-store/core/store/pickup-in-store-store.module.ts:L29-L53). Per lo store finder controllare `findStores.loading/success` *e* `viewAllStores.loading/success` insieme (T11). Le azioni hanno prefisso `[StoreFinder]` (feature-libs/storefinder/core/store/store-finder-state.ts:L11-L11).
2. **La feature è caricata?** RDD/pickup/storefinder sono lazy: i facade risolvono la feature al primo metodo chiamato (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L56). Nel tab Network si vede il chunk; per pickup deve comparire *prima* del chunk add-to-cart/cart (dipendenze in feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L42-L57).
3. **L'outlet appare?** Verificare che il template ospite renda l'id (`cx-delivery-mode`, `cx-order-overview`, `cx-cart-item.details`, `cx-cart-item.delivery-details`, `cx-add-to-cart-pickup-option`) (feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L31). Ricordare T34 per `cx-pickup-info`.
4. **EDD non mostra nulla**: nel Network la risposta del cart deve contenere `arrivalSlots` dentro `entries`; se manca, l'endpoint effettivo non è quello EDD (sezione 3.2, T8). Si può ispezionare la config con `inject(OccConfig)`/`Config` dal debugger.
5. **RDD**: nel Network la richiesta `PUT …/requestedretrievaldate?requestedRetrievalAt=…`; il body conterrà `{"headers":…}` (T1). Breakpoint in `setRequestedDeliveryDate` del componente per vedere quale guard la fa uscire (feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L134-L145).
6. **Mappa vuota**: in dev mode compare il warning "A Google Maps api key is required..." se manca la chiave (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L49-L56); con il toggle async cercare su `window` la funzione `__spartacusGoogleMapsInit_<n>` (feature-libs/storefinder/core/service/google-map-renderer.service.ts:L130-L139).
7. **Negozio preferito**: in DevTools → Application → Local Storage la chiave `preferred_store` (feature-libs/pickup-in-store/root/pickup-in-store-constants.ts:L7-L8); nello stato `pickup-locations.defaultPointOfService`.
8. **Pickup ↔ delivery nel carrello**: osservare se parte un `PUT` (verso delivery) o un `PATCH` con `deliveryPointOfService` (verso pickup) su `updateEntries` (feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L86-L104).
9. **Test unitari** delle lib: `nx test-vitest <library>` o `npx vitest run --config feature-libs/pickup-in-store/vitest.config.ts <spec>` (le lib hanno `vitest.config.ts`, vedi mappa nel dossier; `node_modules` non è installato in questo ambiente, quindi non eseguito qui).

11. Domande di autoverifica

1. **Perché i componenti RDD stanno nell'entry point `root` e non nel modulo lazy?** Perché sono registrati con `provideOutlet` nel root module eager, che richiede la classe del componente; il modulo lazy contiene solo core + OCC. — feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L35-L50, feature-libs/requested-delivery-date/root/public_api.ts:L7-L11, feature-libs/requested-delivery-date/requested-delivery-date.module.ts:L11-L14.
2. **Cosa succede quando l'utente apre lo step delivery mode con un carrello senza data richiesta?** Il picker imposta la data a `earliestRetrievalAt` e chiama subito il backend con un `PUT`; al successo ricarica i delivery mode supportati. — feature-libs/requested-delivery-date/root/components/delivery-mode-

date-picker/delivery-mode-date-picker.component.ts:L84-L99, feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L146-L158.

3. **Perché l'header** `USE_CLIENT_TOKEN` **dell'adapter RDD non ha effetto?** Perché gli header sono passati come secondo argomento di `put`, che è il body. — feature-libs/requested-delivery-date/occ/adapters/occ-requested-delivery-date.adapter.ts:L43-L43, angular:packages/common/http/src/client.ts:L4451-L4454.
4. **Come fa EDD a ottenere** `arrivalSlots` **dal backend senza toccare la cart lib?** Fornisce una default config che ridefinisce gli endpoint `carts`/`cart`/`createCart` con `arrivalSlots` nei fields; le default chunk sono fuse in ordine con `deepMerge`. — feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L28-L29, feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L10-L28, core-libs/core/src/config/services/configuration.service.ts:L76-L100.
5. **Perché nello store finder una ricerca può far apparire "in caricamento" anche la vista dei conteggi?** Perché i due sotto-stati usano `loaderReducer` con lo stesso entity type `STORE_FINDER_DATA`. — feature-libs/storefinder/core/store/reducers/index.ts:L21-L29, core-libs/core/src/state/utils/loader/loader.reducer.ts:L32-L44.
6. **Cosa succede ai risultati dello store finder dopo un cambio lingua?** L'effetto emette `ClearStoreFinderData`, il meta reducer azzera lo stato, `viewAllStores$` ricarica i conteggi e `StoreFinderService` ricarica per country/store dai route params. — feature-libs/storefinder/core/store/effects/view-all-stores.effect.ts:L28-L68, feature-libs/storefinder/core/store/reducers/index.ts:L39-L53, feature-libs/storefinder/core/facade/store-finder.service.ts:L191-L213.
7. **Che valore usare per vedere la mappa senza chiave Google in sviluppo, e cosa viene inviato?** `GOOGLE_MAPS_DEVELOPMENT_KEY_CONFIG` (`'cx-development'`); al caricamento lo script riceve `key` vuota. — feature-libs/storefinder/root/config/constants.ts:L7-L7, feature-libs/storefinder/core/service/google-map-renderer.service.ts:L82-L85, projects/storefrontapp/src/app/private/private.providers.ts:L46-L49.
8. **Come garantisce pickup-in-store che i suoi outlet REPLACE siano registrati quando si apre il PDP o il carrello?** Si dichiara dipendenza delle feature `ADD_TO_CART_FEATURE` e `CART_BASE_FEATURE`, così viene caricato prima di esse. — feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L39-L60.
9. **Come arriva il negozio scelto sul PDP fino alla richiesta di add to cart?** La selezione nel dialog imposta l'intended location con `pickupOption: 'pickup'`; `AddToCartComponent` legge `intendedPickupLocation$` dal componente in outlet (via `cxComponentRef`) e passa il nome a `addEntry`. — feature-libs/pickup-in-store/components/container/store-list/store-list.component.ts:L67-L76, feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L267-L287.
10. **Quale chiamata HTTP parte quando nel carrello si passa da pickup a delivery, e viceversa?** Verso delivery: `PUT` con sola `quantity`; verso pickup: `PATCH` con `deliveryPointOfService.name`. — feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L255-L289, feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L86-L104.

TAVOLA 32.1 **4.1 RDD: dal delivery mode al backend**

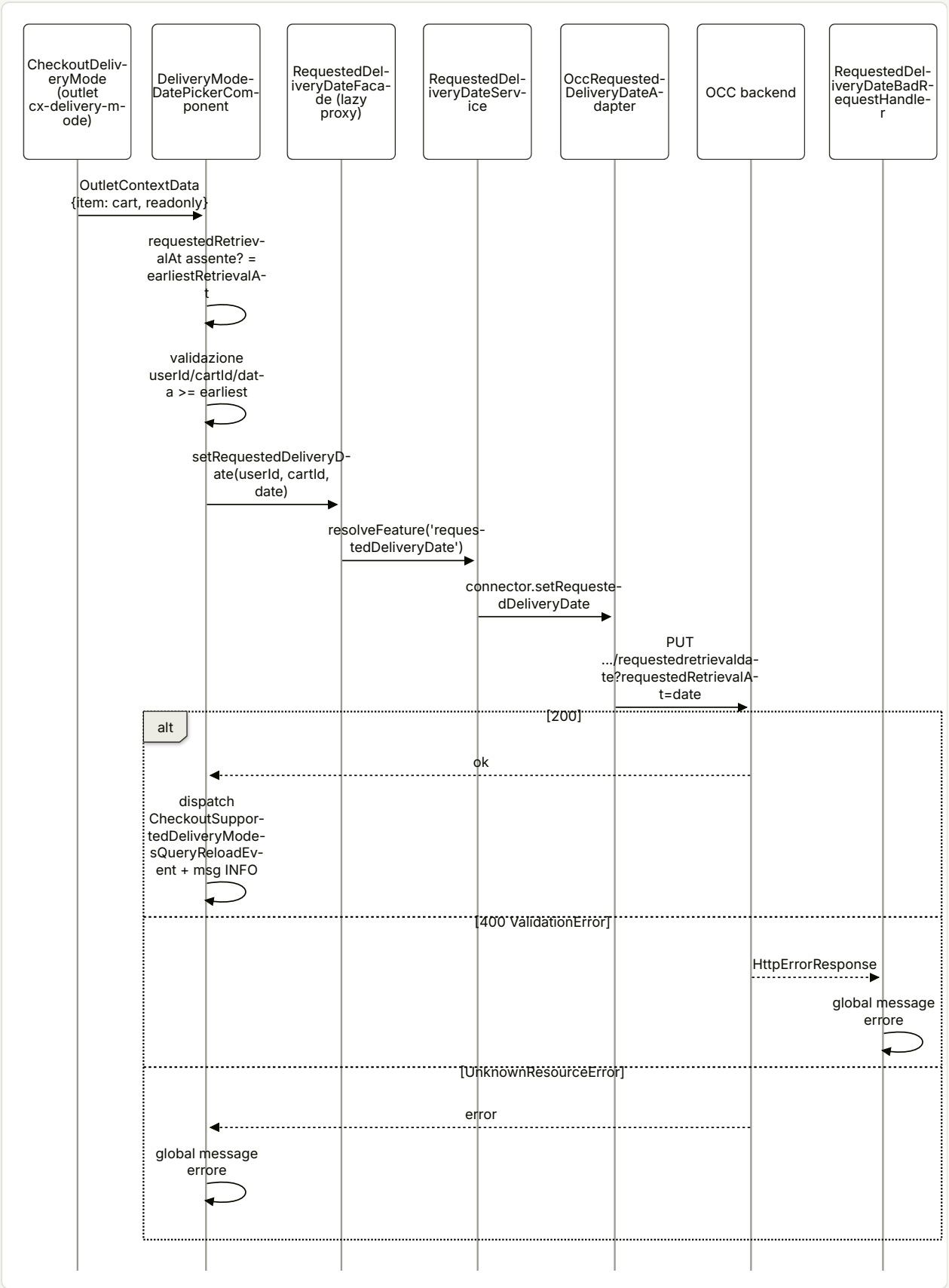
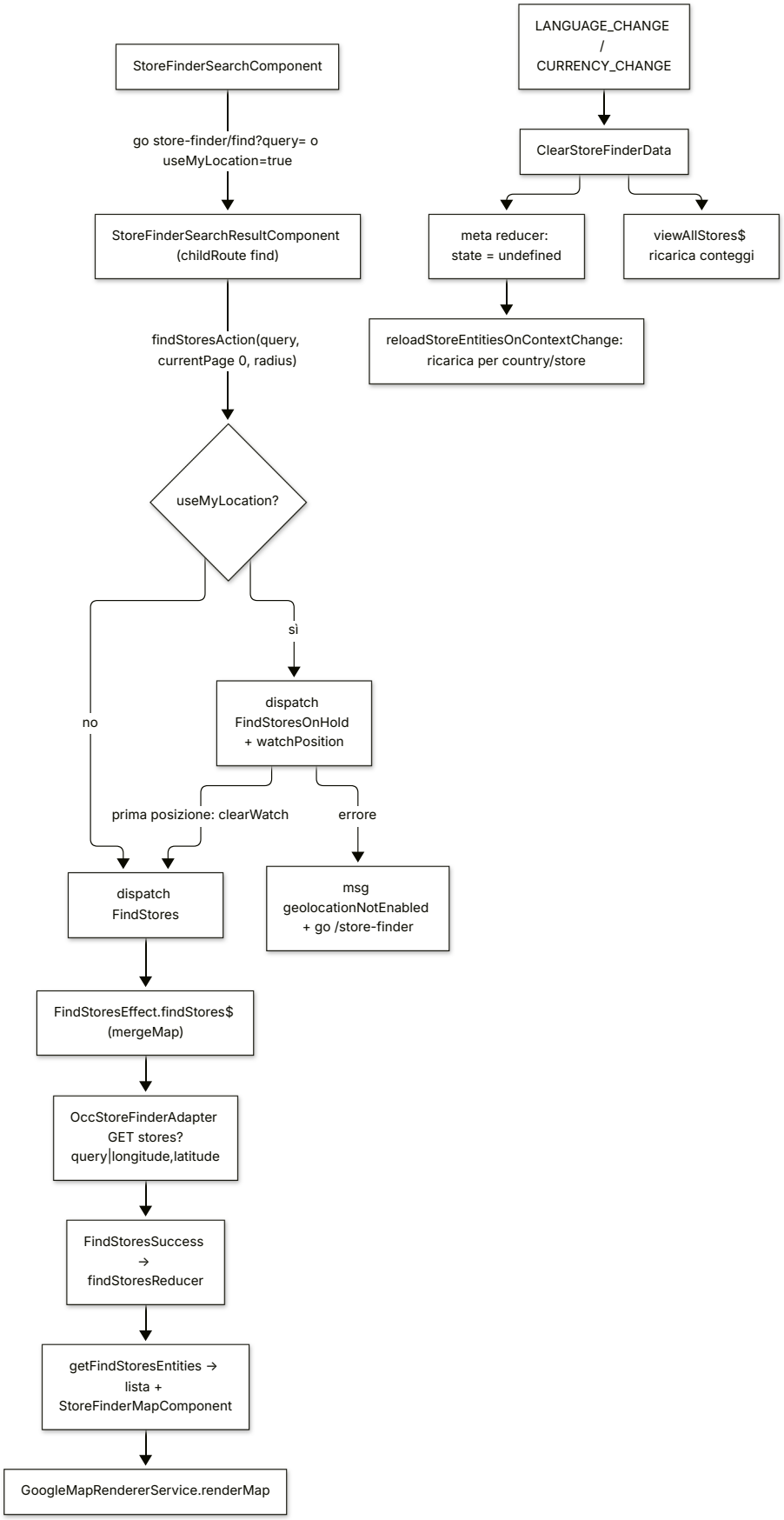


TAVOLA 32.2 **4.2 Store finder: ricerca e cambio contesto**



33

Tracking, SmartEdit, Qualtrics

Librerie che collegano lo storefront a sistemi esterni (GTM, AEP, SmartEdit, Qualtrics).

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Questo capitolo raccoglie tre librerie opzionali di `feature-libs/` che hanno una cosa in comune: **non aggiungono funzionalità di commercio**, ma collegano lo storefront a sistemi esterni che osservano o modificano la pagina dall'esterno. Sono:

LIBRERIA	ENTRY POINT	SISTEMA ESTERNO	MECCANISMO SPARTACUS USATO
`@spartacus/tracking`	`tms/core`, `tms/gtm`, `tms/aep`, `personalization`, `personalization/root`, `personalization/core`	Google Tag Manager, Adobe Experience Platform Launch, personalizzazione SAP Commerce	`EventService`, `APP_INITIALIZER`, `HTTP_INTERCEPTORS`, feature CMS lazy
`@spartacus/smartedit`	`smartedit`, `smartedit/root`, `smartedit/core`	SAP SmartEdit (editor WYSIWYG del CMS)	`APP_INITIALIZER`, `HTTP_INTERCEPTORS`, `ComponentDecorator`/`SlotDecorator`, `FeatureModulesService`
`@spartacus/qualtrics`	`qualtrics`, `qualtrics/root`, `qualtrics/components`	Qualtrics Site Intercept (QSI)	componente CMS lazy + `ScriptLoader`

Le tre librerie riusano in modo esemplare i meccanismi descritti nei capitoli precedenti, ed è per questo che vale la pena leggerle anche se non si usa nessuno dei prodotti esterni:

- **Tracking/TMS** è il consumatore "canonico" del bus eventi: un servizio root si sottoscrive, all'avvio dell'app, a un elenco di classi `CxEvent` scelto in configurazione e inoltra ogni evento a un *collector* che scrive nel data layer della pagina (`feature-libs/tracking/tms/core/services/tms.service.ts:L46-L96`). Il bus eventi è descritto in 20 — Facade, command & query, eventi.
- **Personalization** lavora a livello HTTP: due interceptor fanno da "cookie applicativo", rimandando al backend OCC gli header `Occ-Personalization-Id` / `Occ-Personalization-Time` che il backend stesso ha restituito, persistiti in `localStorage` (`feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L66-L97`). Un servizio separato decodifica il *contesto di personalizzazione* (segmenti e azioni) da un componente CMS nascosto (`feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L30-L69`).
- **SmartEdit** è l'esempio più completo di integrazione "a strati": un root module sempre caricato decide all'avvio se l'app gira dentro l'iframe di SmartEdit (`feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L61-L72`), e solo in quel caso carica lazy il modulo che **decora** ogni slot e componente CMS con il "contratto" HTML richiesto da SmartEdit (`feature-libs/smartedit/core/services/smart-edit.service.ts:L158-L191`) ed espone su `window.smartedit` le callback di re-render (`feature-libs/smartedit/core/services/smart-edit.service.ts:L38-L52`).
- **Qualtrics** è la più piccola: un componente CMS senza template che inserisce lo script di deployment Qualtrics e un servizio che, all'evento `qsi_js_loaded`, invoca l'API QSI (`feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L55-L139`).

Tutte e tre seguono la convenzione root/lazy di Spartacus spiegata in 12 — Feature module e lazy loading: un `*RootModule` piccolo importato eagerly che registra configurazione di default, interceptor o `featureModules`, e un modulo pesante caricato on demand tramite `featureModules.<nome>.module`. La demo app le importa tutte e tre incondizionatamente (`projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L263-L272`).

2. Mappa dei file

2.1 feature-libs/tracking

PATH	RUOLO
feature-libs/tracking/public_api.ts	entry radice vuota (export {};)
feature-libs/tracking/tms/public_api.ts	entry tms vuota
feature-libs/tracking/tms/core/base-tms.module.ts	BaseTmsModule.forRoot() + tmsFactory (APP_INITIALIZER)
feature-libs/tracking/tms/core/config/tms-config.ts	TmsConfig, TmsCollectorConfig, TmsCollectors + augmentation di Config
feature-libs/tracking/tms/core/model/tms.model.ts	interfacce TmsCollector, WindowObject
feature-libs/tracking/tms/core/services/tms.service.ts	TmsService: sottoscrive gli eventi e invoca i collector
feature-libs/tracking/tms/core/public_api.ts	API @spartacus/tracking/tms/core (+ TmsCollectors augmentabile)
feature-libs/tracking/tms/gtm/gtm.module.ts	GtmModule: solo default config
feature-libs/tracking/tms/gtm/config/default-gtm.config.ts	GtmCollectorConfig, defaultGoogleTagManagerConfig, augmentation gtm
feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts	GtmCollectorService: snippet GTM, push su array
feature-libs/tracking/tms/aep/aep.module.ts	AepModule: solo default config
feature-libs/tracking/tms/aep/config/default-aep.config.ts	AepCollectorConfig, defaultAdobeExperiencePlatformConfig, augmentation aep
feature-libs/tracking/tms/aep/services/aep-collector.service.ts	AepCollectorService: script Launch, merge su oggetto
feature-libs/tracking/personalization/root/personalization-root.module.ts	interceptor + default config + feature CMS personalization
feature-libs/tracking/personalization/root/config/personalization-config.ts	PersonalizationConfig
feature-libs/tracking/personalization/root/config/default-personalization-config.ts	default (disabilitata, nomi header, contesto)
feature-libs/tracking/personalization/root/feature-name.ts	PERSONALIZATION_FEATURE
feature-libs/tracking/personalization/root/http-interceptors/index.ts	array interceptors (Id, poi Time)
feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts	header + localStorage personalization-id
feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts	header + localStorage personalization-time
feature-libs/tracking/personalization/root/public_api.ts	API .../personalization/root
feature-libs/tracking/personalization/personalization.module.ts	PersonalizationModule (entry lazy) → importa PersonalizationCoreModule

PATH	RUOLO
feature-libs/tracking/personalization/core/personalization-core.module.ts	PersonalizationCoreModule: @NgModule({}) vuoto
feature-libs/tracking/personalization/core/services/personalization-context.service.ts	PersonalizationContextService: decodifica il contesto dal CMS
feature-libs/tracking/personalization/core/model/personalization-context.model.ts	PersonalizationContext, PersonalizationAction
feature-libs/tracking/schematics/add-tracking/index.ts	addTrackingFeatures (ng-add)
feature-libs/tracking/vitest.config.ts	configurazione Vitest e soglie di coverage

2.2 feature-libs/smartedit

PATH	RUOLO
feature-libs/smartedit/smart-edit.module.ts	SmartEditModule (entry lazy @spartacus/smartedit)
feature-libs/smartedit/root/smart-edit-root.module.ts	interceptor + default config + APP_INITIALIZER (smartEditFactory)
feature-libs/smartedit/root/config/smart-edit-config.ts	SmartEditConfig
feature-libs/smartedit/root/config/default-smart-edit-config.ts	default cx-preview / localhost:9002
feature-libs/smartedit/root/feature-name.ts	SMART_EDIT_FEATURE
feature-libs/smartedit/root/services/smart-edit-launcher.service.ts	SmartEditLauncherService: rileva SmartEdit, gestisce cmsTicketId, carica injector e feature
feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts	CmsTicketInterceptor: aggiunge cmsTicketId alle chiamate OCC
feature-libs/smartedit/root/http-interceptors/index.ts	provider HTTP_INTERCEPTORS
feature-libs/smartedit/core/smart-edit-core.module.ts	SmartEditCoreModule: decoratori + avvio processCmsPage()
feature-libs/smartedit/core/services/smart-edit.service.ts	SmartEditService: contratto HTML, preview page, re-render
feature-libs/smartedit/core/decorators/index.ts	smartEditDecorators (multi provider)
feature-libs/smartedit/core/decorators/smart-edit-component-decorator.ts	SmartEditComponentDecorator
feature-libs/smartedit/core/decorators/smart-edit-slot-decorator.ts	SmartEditSlotDecorator
feature-libs/smartedit/assets/webApplicationInjector.js	script SmartEdit minificato ("Version: 0.0.5"), pubblicato come asset
feature-libs/smartedit/schematics/add-smartedit/index.ts	addSmartEditFeatures (ng-add)

2.3 feature-libs/qualtrics

PATH	RUOLO
feature-libs/qualtrics/qualtrics.module.ts	QualtricsModule (entry lazy @spartacus/qualtrics)

PATH	RUOLO
feature-libs/qualtrics/root/qualtrics-root.module.ts	feature CMS `qualtrics` → `QualtricsComponent`
feature-libs/qualtrics/root/feature-name.ts	`QUALTRICS_FEATURE`
feature-libs/qualtrics/components/qualtrics-components.module.ts	mapping CMS dei due componenti + `defaultQualtricsConfig`
feature-libs/qualtrics/components/qualtrics-loader/qualtrics.component.ts	`cx-qualtrics`: inserisce lo script di deployment
feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts	`QualtricsLoaderService`: integrazione QSI
feature-libs/qualtrics/components/qualtrics-loader/config/qualtrics-config.ts	`QualtricsConfig`
feature-libs/qualtrics/components/qualtrics-loader/config/default-qualtrics-config.ts	`{ qualtrics: {} }`
feature-libs/qualtrics/components/qualtrics-embedded-feedback/qualtrics-embedded-feedback.component.ts	`cx-qualtrics-embedded-feedback`: componente vuoto
feature-libs/qualtrics/styles/_qualtrics-embedded-feedback.scss	stile placeholder
feature-libs/qualtrics/schematics/add-qualtrics/index.ts	`addQualtricsFeatures` (ng-add)

2.4 Codice di core usato dalle tre librerie

PATH	RUOLO NEL CAPITOLO
core-libs/core/src/event/event.service.ts	`EventService.get()` fornisce gli stream che il TMS sottoscrive
core-libs/core/src/util/script-loader.service.ts	`ScriptLoader.embedScript()` usato da AEP, SmartEdit e Qualtrics
core-libs/core/src/cms/decorators/component-decorator.ts, core-libs/core/src/cms/decorators/slot-decorator.ts	classi astratte implementate da SmartEdit
core-libs/core/src/cms/services/dynamic-attribute.service.ts	applica tutti i decoratori a slot e componenti
core-libs/core/src/lazy-loading/unified-injector.ts	trova i decoratori anche nei moduli lazy
core-libs/core/src/lazy-loading/feature-modules.service.ts	`resolveFeature()` usato dal launcher SmartEdit
core-libs/core/src/cms/facade/cms.service.ts	`clearComponentState`, `refreshPageById`, `refreshLatestPage`, `refreshComponent`
core-libs/storefront/cms-structure/services/cms-features.service.ts	mappa componente CMS → feature lazy (Personalization, Qualtrics)
projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts	configurazione demo di TMS e personalization
projects/storefrontapp/src/app/spartacus/features/smartedit/smartedit-feature.module.ts	configurazione demo SmartEdit
projects/storefrontapp/src/app/spartacus/features/qualtrics/qualtrics-feature.module.ts	configurazione demo Qualtrics

File aggiuntivi

PATH	RUOLO
feature-libs/qualtrics/_index.scss	entry SCSS: importa @spartacus/styles/scss/core e ./styles/index
feature-libs/qualtrics/public_api.ts	entry point primario @spartacus/qualtrics: riesporta qualtrics.module
feature-libs/qualtrics/vitest.config.ts	configurazione Vitest di qualtrics (jsdom, coverage v8, nessun alias)
feature-libs/smartedit/public_api.ts	entry point primario @spartacus/smartedit: riesporta smart-edit.module
feature-libs/smartedit/vitest.config.ts	configurazione Vitest di smartedit (alias verso mock-window-ref)

3. Come funziona passo per passo

3.1 TMS: dalla configurazione all'avvio

1. **Import.** L'app importa `BaseTmsModule.forRoot()` più uno o più moduli vendor (`GtmModule`, `AepModule`) e fornisce un `TmsConfig` con gli eventi da tracciare. La demo lo fa in un unico `PersonalizationFeatureModule` (projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts:L22-L39).
2. **Default del collector.** `GtmModule` e `AepModule` non hanno logica: forniscono solo una default config (feature-libs/tracking/tms/gtm/gtm.module.ts:L11-L14, feature-libs/tracking/tms/aep/aep.module.ts:L11-L14) che imposta `tagManager.gtm.collector = GtmCollectorService` (feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L20-L26) e `tagManager.aep.collector = AepCollectorService` (feature-libs/tracking/tms/aep/config/default-aep.config.ts:L20-L26). Nessun evento è attivo di default: gli `events` li decide l'app. Poiché default config e config dell'app vengono fuse (vedi 11 — Configurazione), nella config finale la voce `gtm` contiene sia il `collector` del default sia gli `events` dell'app.
3. **Tipi augmentabili.** `TmsConfig` è un `abstract class` con `useExisting: Config` e una sola proprietà `tagManager?: TmsCollectors` (feature-libs/tracking/tms/core/config/tms-config.ts:L38-L47), fusa nell'interfaccia `Config` di core per module augmentation (feature-libs/tracking/tms/core/config/tms-config.ts:L49-L51). `TmsCollectors` è un dizionario `[tms: string]: TmsCollectorConfig | undefined` (feature-libs/tracking/tms/core/config/tms-config.ts:L31-L33) che i pacchetti vendor estendono a loro volta: GTM aggiunge `gtm?: GtmCollectorConfig` con `gtmId` (feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L10-L18), AEP aggiunge `aep?: AepCollectorConfig` con `scriptUrl` (feature-libs/tracking/tms/aep/config/default-aep.config.ts:L10-L18). Per questo `TmsCollectors` è esportato dentro il blocco delimitato dai commenti AUGMENTABLE_TYPES_START/END del public API (feature-libs/tracking/tms/core/public_api.ts:L12-L14).
4. **`APP_INITIALIZER`.** `BaseTmsModule.forRoot()` registra un `APP_INITIALIZER` multi con factory `tmsFactory` e dipendenza `TmsService` (feature-libs/tracking/tms/core/base-tms.module.ts:L18-L33); la factory restituisce `() => service.collect()` (feature-libs/tracking/tms/core/base-tms.module.ts:L10-L16). Senza `forRoot()` il modulo è un `@NgModule({})` vuoto (feature-libs/tracking/tms/core/base-tms.module.ts:L18-L19): importare `BaseTmsModule` "nudo" non attiva nulla. Il token `APP_INITIALIZER` è deprecato da Angular v19 a favore di `provideAppInitializer` (angular:packages/core/src/application/application_init.ts:L38-L38); l'ordine degli initializer è trattato in 10 — Bootstrap e inizializzazione.
5. **Guardia SSR.** `collect()` esce subito se non si è nel browser (feature-libs/tracking/tms/core/services/tms.service.ts:L46-L49): in SSR nessun data layer è creato e nessuno script vendor inserito.
6. **Iterazione sui TMS.** `collect()` scorre con `for...in` le chiavi di `tagManager`, saltando le proprietà non proprie (feature-libs/tracking/tms/core/services/tms.service.ts:L51-L54). Una voce senza `collector`

viene saltata con warning in dev mode (feature-libs/tracking/tms/core/services/tms.service.ts:L59-L66).

7. **Stream eventi.** Per ogni classe in `events` ottiene lo stream con `eventsService.get(event)`; senza `events` la lista è vuota (feature-libs/tracking/tms/core/services/tms.service.ts:L68-L70). `EventService.get` restituisce l'`output$` del *merging subject* del tipo, validato in dev mode (core-libs/core/src/event/event.service.ts:L90-L96).
8. **Istanza e `init`.** Il collector è ottenuto con `this.injector.get(collectorConfig.collector)` — l'`Injector` iniettato in `TmsService`, che è `providedIn: 'root'` (feature-libs/tracking/tms/core/services/tms.service.ts:L27-L41) — e `init` è chiamato subito con `nativeWindow!` (feature-libs/tracking/tms/core/services/tms.service.ts:L71-L75).
9. **Push.** Gli stream sono uniti da `mapEvents` (protected) con `merge(...events)` (feature-libs/tracking/tms/core/services/tms.service.ts:L104-L108). A ogni evento: log con emoji 🎤 se `debug`, poi `collector.map?.(event) ?? event`, poi `collector.pushEvent(config, window, event)`; la subscription è accumulata in `this.subscription` (feature-libs/tracking/tms/core/services/tms.service.ts:L77-L94) e chiusa in `ngOnDestroy` (feature-libs/tracking/tms/core/services/tms.service.ts:L113-L115).

3.2 I due collector: GTM e AEP a confronto

ASPETTO	GTMCOLLECTORSERVICE	AEPCOLLECTORSERVICE
Dipendenza iniettata	`WindowRef` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L19-L21)	`ScriptLoader` (feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L19-L21)
Nome data layer di default	`'dataLayer'` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L26-L26)	`'digitalData'` (feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L26-L26)
Tipo del data layer	array, `?? []` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L27-L27)	oggetto, `?? {}` (feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L27-L27)
Caricamento script vendor	solo con `gtmId`: snippet GTM inline che inserisce `gtm.js` prima del primo `<script>` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L29-L51)	solo con `scriptUrl`: `scriptLoader.embedScript({ src })` (feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L29-L31)
`pushEvent`	`window[dataLayer].push(event)` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L55-L62)	sostituzione con spread shallow `{ ...precedente, ...event }` (feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L34-L44)
`map`	non implementato	non implementato

Dettagli dello snippet GTM: prima pusha `{ 'gtm.start': <timestamp>, event: 'gtm.js' }` nel data layer (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L38-L38), poi crea un `<script async>` con `src = 'https://www.googletagmanager.com/gtm.js?id=' + gtmId` e, se il data layer non si chiama `dataLayer`, aggiunge `&l=<nome>` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L41-L43), infine lo inserisce prima del primo `<script>` del documento (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L39-L44).

Per AEP, `ScriptLoader.embedScript` crea un `<script async defer>`, non duplica uno script già presente (controllo fatto solo nel browser) e lo appende a `head` per default (core-libs/core/src/util/script-loader.service.ts:L55-L68, core-libs/core/src/util/script-loader.service.ts:L91-L93).

3.3 Personalization: gli interceptor degli header OCC

1. **Registrazione.** `PersonalizationRootModule` aggiunge l'array `interceptors`, la default config e una default config factory per la feature CMS (feature-libs/tracking/personalization/root/personalization-root.module.ts:L30-L37). Gli interceptor sono provider `HTTP_INTERCEPTORS` multi con `useExisting`, nell'ordine Id

→ Time (feature-libs/tracking/personalization/root/http-interceptors/index.ts:L13-L24). Angular costruisce la catena con `reduceRight`, quindi il primo registrato è il più esterno (angular:packages/common/http/src/interceptor.ts:L233-L240; vedi 06 — HttpClient e interceptor).

2. **Default.** `personalization.enabled` è `false`, gli header sono `Occ-Personalization-Id` e `Occ-Personalization-Time`, il contesto si trova nello slot `PlaceholderContentSlot` nel componente `PersonalizationScriptComponent` (feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L9-L21). La forma della config è definita in `PersonalizationConfig` (feature-libs/tracking/personalization/root/config/personalization-config.ts:L10-L30).
3. **Costruzione dell'interceptor (solo browser).** `enabled = (localStorage && personalization.enabled) || false` (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L34-L37). Se abilitato, il nome header è messo in minuscolo e l'id iniziale letto dalla chiave `personalization-id` di `localStorage` (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L39-L49, chiave a feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L25-L25). Se disabilitato e c'è un id residuo, viene **rimosso** (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L50-L54): spegnere la feature pulisce lo storage al successivo avvio.
4. **Richiesta.** Se disabilitato, la richiesta passa invariata (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L62-L64). Altrimenti l'header è aggiunto solo se nome e valore sono noti e l'URL contiene `occEndpoints.getBaseUrl()` (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L66-L76).
5. **Risposta.** Su `HttpResponse`, se l'header compare fra `event.headers.keys()` e differisce dal valore in memoria, il nuovo valore è salvato in memoria e in `localStorage` (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L78-L97).
6. **Timestamp.** `OccPersonalizationTimeInterceptor` è identico ma usa la chiave `personalization-time` e `httpHeaderName.timestamp` (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L25-L25, feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L45-L49).

In pratica il backend assegna un id di personalizzazione alla prima risposta; lo storefront lo memorizza e lo rimanda a ogni chiamata OCC successiva, così il backend può riconoscere l'utente anonimo e applicare le stesse *customization*. DA VERIFICARE: che il backend emetta davvero questi header ed esponga il CORS `Access-Control-Expose-Headers` non è verificabile nel repo (dossier Q3).

3.4 Personalization: il contesto (segmenti e azioni)

1. **Feature CMS.** La default config factory associa la feature `personalization` al solo componente CMS `PersonalizationScriptComponent`, senza `module` (feature-libs/tracking/personalization/root/personalization-root.module.ts:L17-L28). Il `module` lo fornisce l'app: la demo punta a `import('@spartacus/tracking/personalization')` → `PersonalizationModule` (projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts:L40-L48). `CmsFeaturesService` considera un componente "gestito da una feature" solo se la feature ha sia `module` sia `cmsComponents` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76).
2. **Modulo lazy vuoto.** `PersonalizationModule` importa soltanto `PersonalizationCoreModule`, che è un `@NgModule({})` vuoto (feature-libs/tracking/personalization/personalization.module.ts:L10-L13, feature-libs/tracking/personalization/core/personalization-core.module.ts:L9-L10). Il vero codice di `core` è il servizio `PersonalizationContextService`, `providedIn: 'root'` (feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L19-L28): viene istanziato da chi lo inietta, non dal caricamento del modulo.

3. **Lettura del contesto.** Senza `personalization.context`, `getPersonalizationContext()` restituisce `EMPTY` con warning in dev (feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L30-L35). Altrimenti: pagina corrente → slot `context.slotPosition` → componente con `uid === context.componentId` → `properties.script.data` (feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L37-L50).
4. **Decodifica.** Il payload è `JSON.parse(atob(data))`; poi ogni valore di ogni action e ogni segment è a sua volta decodificato base64 (feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L54-L69). Il risultato ha la forma `{ actions: PersonalizationAction[]; segments: string[] }` (feature-libs/tracking/personalization/core/model/personalization-context.model.ts:L7-L19).
5. **Consumatore.** L'unico consumatore nel repo è `ProfileTagPushEventsService` di CDS, che combina ogni push event con `withLatestFrom(merge(of({segments: undefined, actions: undefined}), getPersonalizationContext()))` e scrive `segments`/`actions` in `item.data` (integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L98-L112). CDS è trattato in 34 — Integration libs: CDC, CDP, CDS.

3.5 SmartEdit: avvio e rilevamento

1. **Root module.** `SmartEditRootModule` registra `CmsTicketInterceptor`, la default config e un `APP_INITIALIZER` (`smartEditFactory`) che chiama `SmartEditLauncherService.load()` (feature-libs/smartedit/root/smartedit-root.module.ts:L13-L34). Default: `storefrontPreviewRoute: 'cx-preview'`, `allowOrigin: 'localhost:9002'` (feature-libs/smartedit/root/config/default-smart-edit-config.ts:L9-L14).
2. **Parse dell'URL.** `isLaunchedInSmartEdit()` separa `location.path()` su `?`, cerca il parametro che inizia con `cmsTicketId=` e ne prende il valore con `split('=')[1]` (feature-libs/smartedit/root/services/smartedit-launcher.service.ts:L61-L72).
3. **Pagina iniziale.** Se l'ultimo segmento del path è `storefrontPreviewRoute` **e** c'è un `cmsTicketId`, il ticket è persistito in `sessionStorage` (chiave `smartedit.cmsTicketId`) e memorizzato (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L74-L89, chiave a feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L27-L32).
4. **Fallback da sessionStorage.** Altrimenti il ticket viene ripristinato da `sessionStorage` e la sua sola presenza basta per considerarsi "in SmartEdit" — il caso d'uso dichiarato sono i redirect full-page, ad esempio il flusso OIDC di CDC (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L91-L96). Persistenza e ripristino sono saltati fuori dal browser, per evitare il `SecurityError` in SSR (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L98-L118).
5. **Caricamento.** Se in SmartEdit, `load()` risolve la feature `smartEdit` (lazy load di `SmartEditModule`) e inserisce lo script con `src: 'assets/webApplicationInjector.js'` (path relativo hardcoded) con `id: 'text/smartedit-injector'` e `data-smartedit-allow-origin` preso dalla config (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L43-L56). `ScriptLoader` usa `setAttribute` per le chiavi che iniziano con `data-` (e, solo con `disableKeyRestriction`, per quelle che non sono proprietà dello `<script>`); le altre diventano proprietà del DOM element. SmartEdit non passa `disableKeyRestriction`, quindi `id` diventa `script.id` (core-libs/core/src/util/script-loader.service.ts:L70-L82). Il file `webApplicationInjector.js` è minificato ("Version: 0.0.5") e fa parte degli asset pubblicati da ng-packagr (feature-libs/smartedit/assets/webApplicationInjector.js:L1-L2, feature-libs/smartedit/ng-package.json:L7-L11).
6. **`resolveFeature`.** `FeatureModulesService.resolveFeature` restituisce un errore "No module defined for Feature Module ..." se la feature non ha `module`; altrimenti risolve dipendenze e modulo con `shareReplay()` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73). La demo configura `featureModules.smartEdit` con `import('@spartacus/smartedit')` → `SmartEditModule` (projects/storefrontapp/src/app/spartacus/features/smartedit/smartedit-feature.module.ts:L14-L27).

DA VERIFICARE: l'URL esatto con cui SmartEdit apre lo storefront (`.../cx-preview?cmsTicketId=...`) è dedotto dal parsing in feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L74-L89; il protocollo di `webApplicationInjector.js` non è leggibile (file minificato di terze parti).

3.6 SmartEdit: il `cmsTicketId` sulle chiamate OCC

`CmsTicketInterceptor` legge `service.cmsTicketId` dal launcher; se assente la richiesta passa invariata (feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L29-L32). Altrimenti decide per sottostringa dell'URL:

- `/productList` → legge il `PageContext` corrente con `take(1)` e aggiunge `cmsTicketId` più `categoryCode = pageContext.id` se presente (feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L33-L35, feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L47-L72);
- `/cms/` o `/products/` → aggiunge il query param `cmsTicketId` (feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L36-L42).

Il ticket permette al backend di restituire la versione *in anteprima* (staged) dei contenuti. L'interceptor è registrato multi con `useExisting` (feature-libs/smartedit/root/http-interceptors/index.ts:L11-L17). Poiché il root module è eager, l'interceptor è sempre nella catena, ma è un pass-through quando non c'è ticket.

3.7 SmartEdit: modulo lazy, decoratori e contratto HTML

1. **Lazy load.** `SmartEditModule` importa `SmartEditCoreModule` (feature-libs/smartedit/smart-edit.module.ts:L10-L13), che fornisce `smartEditDecorators` e nel costruttore chiama `smartEditService.processCmsPage()` (feature-libs/smartedit/core/smart-edit-core.module.ts:L11-L18).
2. **Decoratori multi.** `smartEditDecorators` registra `SmartEditComponentDecorator` su `ComponentDecorator` e `SmartEditSlotDecorator` su `SlotDecorator` (multi, `useExisting`) (feature-libs/smartedit/core/decorators/index.ts:L12-L23). Sono classi astratte di core con il solo metodo `decorate(element, renderer, data?)` (core-libs/core/src/cms/decorators/component-decorator.ts:L10-L23, core-libs/core/src/cms/decorators/slot-decorator.ts:L10-L23).
3. **Come i decoratori lazy arrivano al rendering.** `DynamicAttributeService` raccoglie i decoratori con `unifiedInjector.getMulti(...)` + `shareReplay(1)` (core-libs/core/src/cms/services/dynamic-attribute.service.ts:L24-L32). `UnifiedInjector.injectors$` parte dal root injector e aggiunge l'injector di ogni modulo lazy caricato, mentre `getMulti` accumula le istanze con `scan` (core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35, core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93). Così i provider di `SmartEditCoreModule` — che vive in un injector figlio — diventano visibili a un servizio root.
4. **Applicazione.** `PageSlotComponent` chiama `addAttributesToSlot` quando lo slot è presente (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L157-L163); `ComponentWrapperDirective.decorate` chiama `addAttributesToComponent` (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L138-L144). Entrambi i metodi leggono in modo sincrono (`getLastValueSync`) l'elenco dei decoratori e li invocano tutti (core-libs/core/src/cms/services/dynamic-attribute.service.ts:L40-L64). Il rendering CMS è descritto in 16 — CMS (storefront): rendering dinamico.
5. **Contratto.** Entrambi i decoratori passano `properties` del componente/slot a `addSmartEditContract` (feature-libs/smartedit/core/decorators/smart-edit-component-decorator.ts:L19-L31, feature-libs/smartedit/core/decorators/smart-edit-slot-decorator.ts:L19-L27). Il metodo itera i gruppi di `properties` (tipicamente `smartedit`): `classes` è splittata su spazi e applicata con `addClass`; ogni altra proprietà diventa un attributo `data-<gruppo>-<prop-in-kebab-case>` (feature-libs/smartedit/core/services/smart-edit.service.ts:L158-L191). Per esempio una proprietà `componentId` nel gruppo

`smartedit` diventa `data-smartedit-component-id` (trasformazione `split(/(?=[A-Z])/).join('-').toLowerCase()` a feature-libs/smartedit/core/services/smart-edit.service.ts:L178-L186).

6. **Contratto di pagina.** `processCmsPage()` attende il primo `BaseSite` non nullo, ne legge `defaultPreviewCategoryCode`/`defaultPreviewProductCode` (campi dichiarati in core-libs/core/src/model/misc.model.ts:L116-L121), poi per ogni pagina CMS corrente salva `pageId`, chiama `goToPreviewPage` e `addPageContract` (feature-libs/smartedit/core/services/smart-edit.service.ts:L55-L76). `addPageContract` crea un renderer per `body`, rimuove **tutte** le classi presenti sul `<body>` e applica il contratto da `cmsPage.properties` (feature-libs/smartedit/core/services/smart-edit.service.ts:L78-L94).
7. **Preview page.** `goToPreviewPage` agisce solo sulla prima pagina: se è `PRODUCT_PAGE` con `defaultPreviewProductCode` naviga a `cxRoute: 'product'`, se è `CATEGORY_PAGE` con `defaultPreviewCategoryCode` naviga a `cxRoute: 'category'` (feature-libs/smartedit/core/services/smart-edit.service.ts:L96-L121). Serve quando SmartEdit apre un *template* di pagina prodotto/categoria: lo storefront ha bisogno di un prodotto/categoria reale da mostrare.

3.8 SmartEdit: re-render dopo un edit

1. Il costruttore di `SmartEditService`, se esiste `nativeWindow`, crea `window.smartedit` ed espone `renderComponent` (delegato al metodo protected) e `reprocessPage` (feature-libs/smartedit/core/services/smart-edit.service.ts:L38-L52).
2. `renderComponent(componentId, componentType?, parentId?)` gira dentro `zone.run` — chiamato da codice esterno ad Angular, deve rientrare in zona per far scattare la change detection (vedi 04 — Change detection, zone e signals). Senza `parentId` si tratta di uno slot: svuota tutta la cache dei componenti e ricarica la pagina per `pageId` (o l'ultima pagina); con `parentId` e `componentType` ricarica il singolo componente. Restituisce sempre `true` (feature-libs/smartedit/core/services/smart-edit.service.ts:L126-L149).
3. Lato core: `clearComponentState()` senza argomenti cancella tutti i componenti in cache e dispatcha `ClearCmsComponent` per ciascuno, e il JSDoc cita proprio SmartEdit (core-libs/core/src/cms/facade/cms.service.ts:L296-L309); `refreshPageById` dispatcha `LoadCmsPageData({ id })`, `refreshLatestPage` usa il page context corrente, `refreshComponent` dispatcha `LoadCmsComponent` (core-libs/core/src/cms/facade/cms.service.ts:L197-L223). Lo store CMS è descritto in 15 — CMS (core).
4. `reprocessPage()` è vuoto con un `TODO: reprocess page API` (feature-libs/smartedit/core/services/smart-edit.service.ts:L151-L153).

3.9 Qualtrics

1. **Root.** `QualtricsRootModule` fornisce solo la default config factory che mappa la feature `qualtrics` sul componente CMS `QualtricsComponent` (feature-libs/qualtrics/root/qualtrics-root.module.ts:L11-L27). La demo aggiunge `module: () => import('@spartacus/qualtrics')` → `QualtricsModule` (projects/storefrontapp/src/app/spartacus/features/qualtrics/qualtrics-feature.module.ts:L14-L27).
2. **Lazy.** `QualtricsModule` importa `QualtricsComponentsModule` (feature-libs/qualtrics/qualtrics.module.ts:L10-L13), che importa i due componenti standalone, li mappa in `cmsComponents` e fornisce `defaultQualtricsConfig` (feature-libs/qualtrics/components/qualtrics-components.module.ts:L14-L35). Il default è `{ qualtrics: {} }`, cioè nessuno `scriptSource` (feature-libs/qualtrics/components/qualtrics-loader/config/default-qualtrics-config.ts:L9-L11).
3. **Componente.** `QualtricsComponent` (`cx-qualtrics`, template vuoto) nel **costruttore** chiama `qualtricsLoader.addScript(scriptSource)` se configurato, altrimenti warning in dev (feature-libs/qualtrics/components/qualtrics-loader/qualtrics.component.ts:L15-L33). Il JSDoc di `QualtricsConfig` suggerisce di tenere lo script di deployment negli asset locali (feature-libs/qualtrics/components/qualtrics-loader/config/qualtrics-config.ts:L10-L34).

4. **Service.** `QualtricsLoaderService` nel costruttore chiama `initialize()`, che sottoscrive `qsi$` e invoca `run()` a ogni emissione (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L77-L83, feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L113-L115). `qsiLoaded$` è `fromEvent(window, 'qsi_js_loaded')` solo nel browser, altrimenti `EMPTY` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L55-L58); `qsi$` passa per l'hook `isDataLoaded()`, prende `window.QSI`, filtra i falsy e salva in `qsiApi` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L66-L71).
5. `addScript`. Se lo script è già nel DOM (`script[src="..."]`) chiama `run(true)`, altrimenti lo inserisce con `scriptLoader.embedScript({ src })` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L91-L106). Il primo ramo (`run(true)`) copre la navigazione SPA: tornando su una pagina con il componente, lo script c'è già e serve solo scaricare e rivalutare gli intercept.
6. `run`. Esce (log in dev) se `qsiApi.API` non c'è; con `reload` chiama `API.unload()`; poi `API.load().done(API.run())` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L123-L139).
7. **Hook.** `isDataLoaded()` è protected e di default `of(true)` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L141-L149).
8. **Componente embedded.** `QualtricsEmbeddedFeedbackComponent` (`cx-qualtrics-embedded-feedback`) ha template vuoto e nessuna logica (feature-libs/qualtrics/components/qualtrics-embedded-feedback/qualtrics-embedded-feedback.component.ts:L9-L13): è solo un segnaposto nel DOM in cui un intercept Qualtrics "embedded" può renderizzarsi. Anche il suo stile è un placeholder (feature-libs/qualtrics/styles/_qualtrics-embedded-feedback.scss:L1-L4).

3.10 Installazione con gli schematics

`addTrackingFeatures`, `addSmartEditFeatures` e `addQualtricsFeatures` hanno la stessa sequenza: `validateSpartacusInstallation`, `analyzeCrossFeatureDependencies`, poi `chain([analyzeApplication, addFeatures, addPackageJsonDependenciesForLibrary(peerDependencies), finalizeInstallation])` (feature-libs/tracking/schematics/add-tracking/index.ts:L25-L43, feature-libs/smartedit/schematics/add-smartedit/index.ts:L25-L43, feature-libs/qualtrics/schematics/add-qualtrics/index.ts:L25-L43). I nomi delle feature selezionabili sono `Personalization`, `TMS-GTM`, `TMS-AEPL`, `SmartEdit`, `Qualtrics` (core-libs/schematics/src/shared/libs-constants.ts:L352-L364).

- **Personalization**: feature module `PersonalizationModule`, root `PersonalizationRootModule`, chunk lazy con `PERSONALIZATION_FEATURE`, dipendenza da User Profile (core-libs/schematics/src/shared/lib-configs/tracking-schematics-config.ts:L29-L50).
- **GTM / AEP**: root module `BaseTmsModule.forRoot()` e una `customConfig` che genera `<TmsConfig>{ tagManager: { gtm|aep: { events: [] } } }` forzando `lazy: false` (core-libs/schematics/src/shared/lib-configs/tracking-schematics-config.ts:L57-L103, core-libs/schematics/src/shared/lib-configs/tracking-schematics-config.ts:L107-L154); anche queste dichiarano la dipendenza da User Profile (core-libs/schematics/src/shared/lib-configs/tracking-schematics-config.ts:L75-L75, core-libs/schematics/src/shared/lib-configs/tracking-schematics-config.ts:L125-L125).
- **SmartEdit**: genera `<SmartEditConfig>` con `storefrontPreviewRoute`/`allowOrigin` presi dalle opzioni o dai placeholder `STOREFRONT_PREVIEW_ROUTE_PLACEHOLDER`/`ALLOWED_ORIGIN_PLACEHOLDER` (core-libs/schematics/src/shared/lib-configs/smartedit-schematics-config.ts:L54-L75) e aggiunge l'asset `smartedit/assets` (`glob: '**/*'`) ad `angular.json` (core-libs/schematics/src/shared/lib-configs/smartedit-schematics-config.ts:L48-L51).
- **Qualtrics**: crea `qualtrics-embedded-feedback.scss` con gli import delle funzioni/variabili/mixin bootstrap (core-libs/schematics/src/shared/lib-configs/qualtrics-schematics-config.ts:L45-L53).

Gli schematics in generale sono in 43 — Schematics e migrazioni.

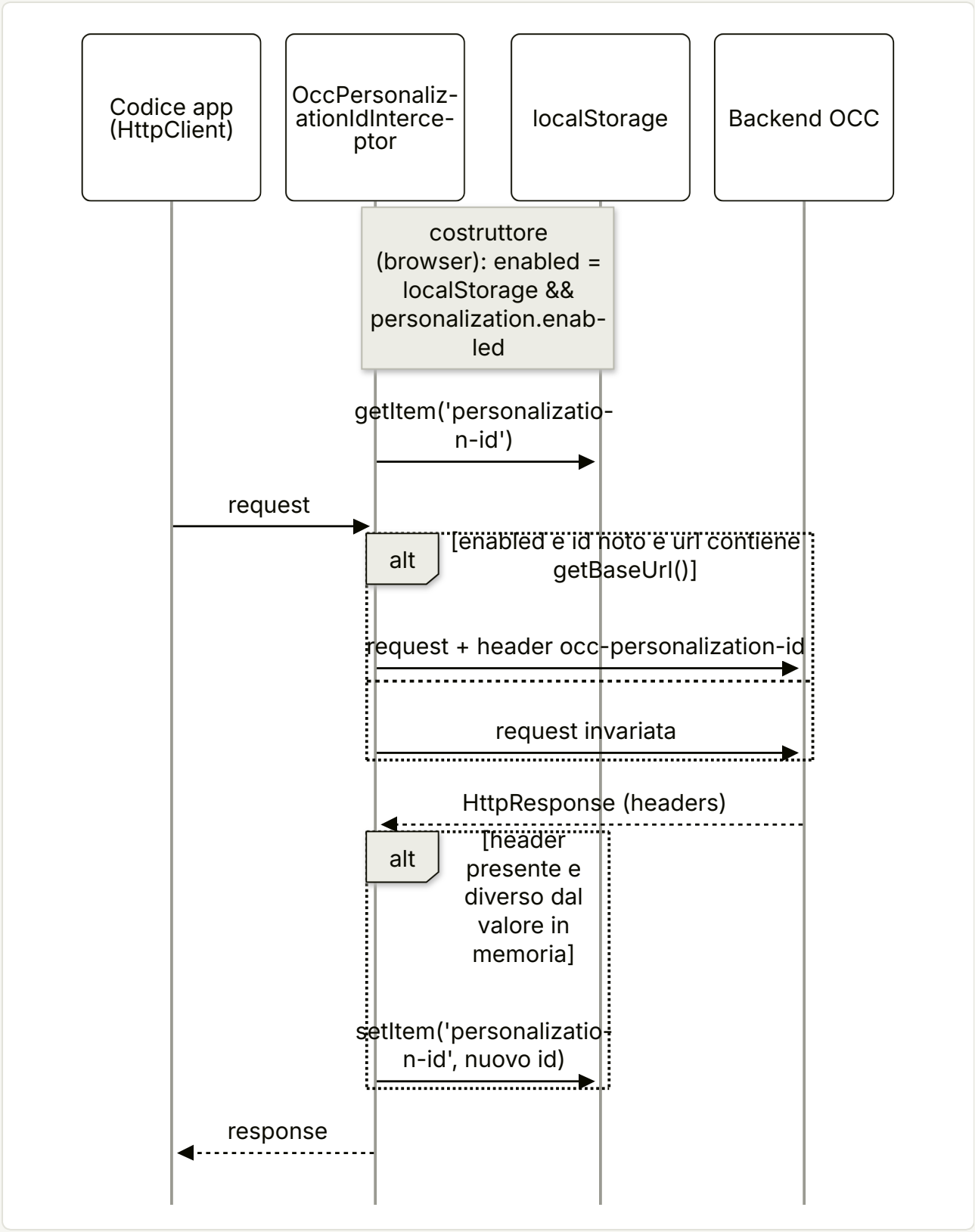
4. Diagrammi

4.1 TMS: bootstrap e flusso di un evento

■ **TAVOLA 33.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: initializer `feature-libs/tracking/tms/core/base-tms.module.ts:L13-L29`; guardia SSR `feature-libs/tracking/tms/core/services/tms.service.ts:L46-L49`; loop e `init` `feature-libs/tracking/tms/core/services/tms.service.ts:L51-L75`; push `feature-libs/tracking/tms/core/services/tms.service.ts:L77-L94`; GTM `feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L25-L62`.

4.2 Personalization: ciclo dell'header id



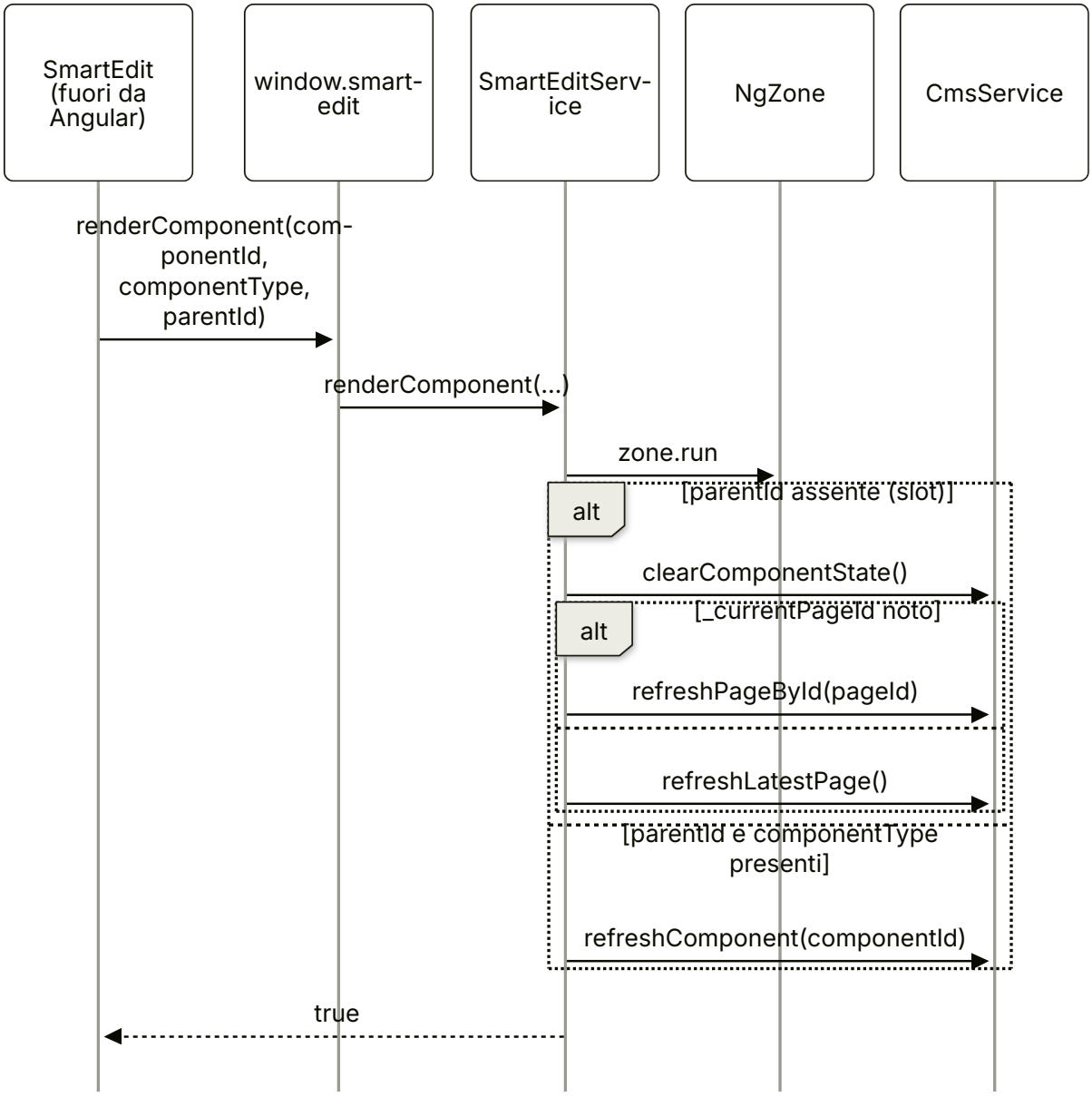
Riferimenti: `feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L34-L49` (costruttore), `feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L66-L76` (richiesta), `feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L78-L97` (risposta).

4.3 SmartEdit: avvio, lazy load e decorazione

■ **TAVOLA 33.2** diagramma a tutta pagina, in fondo al capitolo

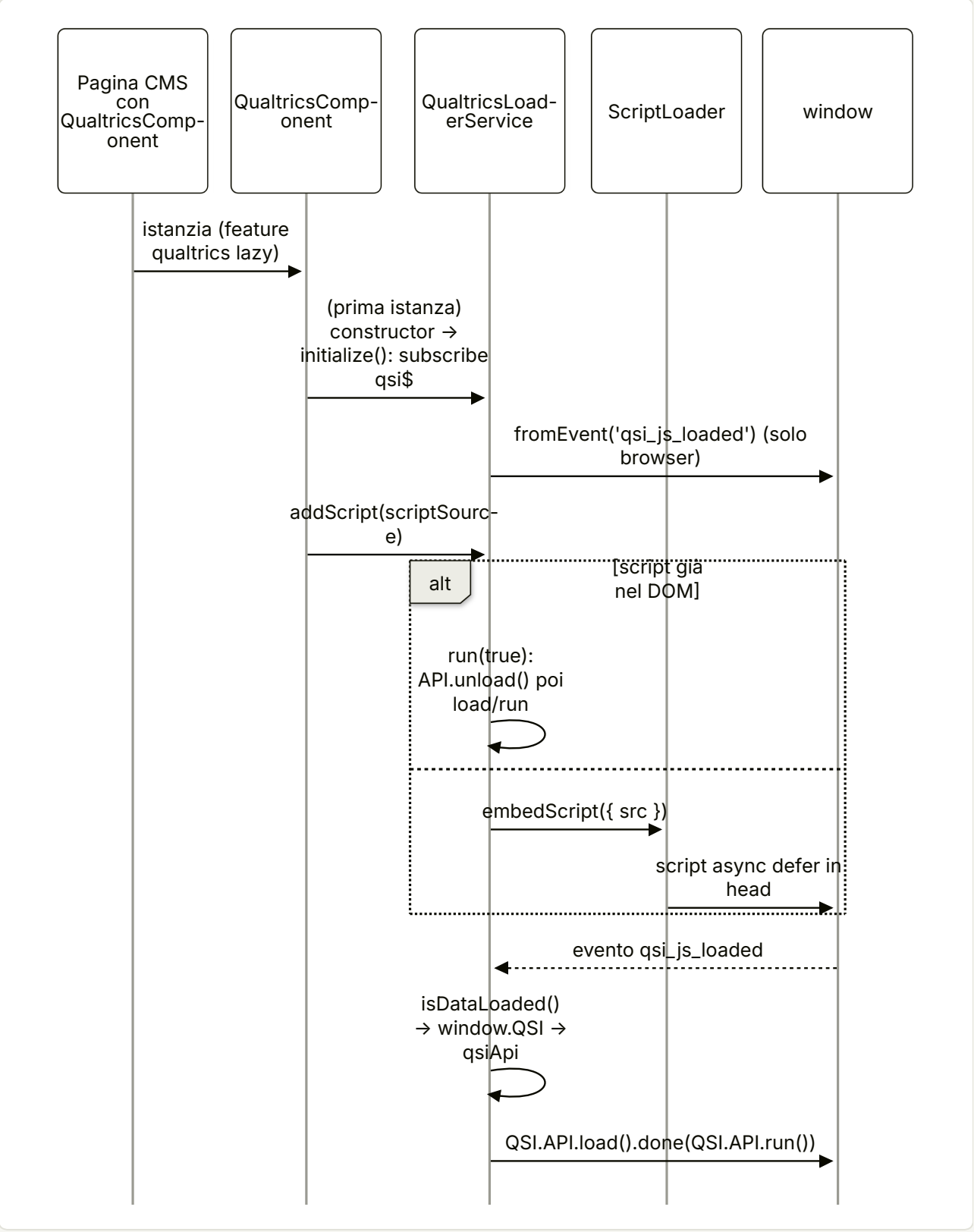
Riferimenti: feature-libs/smartedit/root/smart-edit-root.module.ts:L13-L34; feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L43-L96; feature-libs/smartedit/core/smart-edit-core.module.ts:L11-L18; core-libs/core/src/cms/services/dynamic-attribute.service.ts:L24-L64; feature-libs/smartedit/core/services/smart-edit.service.ts:L55-L191; feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L25-L72.

4.4 SmartEdit: re-render dopo un edit



Riferimenti: feature-libs/smartedit/core/services/smart-edit.service.ts:L38-L52, feature-libs/smartedit/core/services/smart-edit.service.ts:L126-L149, core-libs/core/src/cms/facade/cms.service.ts:L197-L223, core-libs/core/src/cms/facade/cms.service.ts:L302-L309.

4.5 Qualtrics: script e API QSI



Riferimenti: `feature-libs/qualtrics/components/qualtrics-loader/qualtrics.component.ts:L22-L33`, `feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L55-L139`, `core-libs/core/src/util/script-loader.service.ts:L55-L93`.

5. Estratti di codice reale

5.1 `BaseTmsModule` e `tmsFactory`

TS

```ts
// feature-libs/tracking/tms/core/base-tms.module.ts:L10-L33
/**
 * The factory that conditionally (based on the configuration) starts collecting events
 */
export function tmsFactory(service: TmsService): () => void {
  const result = () => service.collect();
  return result;
}

@NgModule({})
export class BaseTmsModule {
  static forRoot(): ModuleWithProviders<BaseTmsModule> {
    return {
      ngModule: BaseTmsModule,
      providers: [
        {
          provide: APP_INITIALIZER,
          useFactory: tmsFactory,
          deps: [TmsService],
          multi: true,
        },
      ],
    };
  }
}
```

5.2 `TmsConfig` e i tipi di configurazione

TS

```ts
// feature-libs/tracking/tms/core/config/tms-config.ts:L14-L51
export interface TmsCollectorConfig {
  /** Should be enabled in development mode only */
  debug?: boolean;
  /**
   * The name for the data layer object.
   */
  dataLayerProperty?: string;
  /**
   * Events to send to the configured TMS.
   */
  events?: AbstractType<CxEvent>[];
  /**
   * The collector service implementation.
```

```
   */
  collector?: Type<TmsCollector>;
}

export interface TmsCollectors {
  [tms: string]: TmsCollectorConfig | undefined;
}

/**
 * TMS configuration
 */
@Injectable({
  providedIn: 'root',
  useExisting: Config,
})
export abstract class TmsConfig {
  /**
   * Tag manager config
   */
  tagManager?: TmsCollectors;
}

declare module '@spartacus/core' {
  interface Config extends TmsConfig {}
}
```

5.3 Il contratto `TmsCollector`

TS

```
// feature-libs/tracking/tms/core/model/tms.model.ts:L14-L40
export interface WindowObject extends Window {
  [prop: string]: any;
}

/**
 * Interface that a class can implement in order to be recognized as a TMS collector.
 */
export interface TmsCollector {
  /**
   * Initializes the data layer.
   */
  init(config: TmsCollectorConfig, windowObject: WindowObject): void;

  /**
   * Pushes the provided event to the data layer.
   */
  pushEvent<T extends CxEvent>(
    config: TmsCollectorConfig,
    windowObject: WindowObject,
```

```
    event: T | any
  ): void;

  /**
   * Maps the Spartacus event to another object.
   */
  map?<T extends CxEvent>(event: T): T | object;
}
```

5.4 `TmsService.collect()` (cuore del TMS)

TS

```
// feature-libs/tracking/tms/core/services/tms.service.ts:L46-L108
  collect(): void {
    if (!this.windowRef.isBrowser()) {
      return;
    }

    for (const tmsCollectorConfig in this.tmsConfig.tagManager) {
      if (!this.tmsConfig.tagManager?.hasOwnProperty(tmsCollectorConfig)) {
        continue;
      }

      const collectorConfig =
        this.tmsConfig.tagManager[tmsCollectorConfig] ?? {};

      if (!collectorConfig.collector) {
        if (isDevMode()) {
          this.logger.warn(
            `Skipping the '${tmsCollectorConfig}', as the collector is not defined.`
          );
        }
        continue;
      }

      const events =
        collectorConfig.events?.map((event) => this.eventsService.get(event)) ||
        [];
      const collector = this.injector.get<TmsCollector>(
        collectorConfig.collector
      );
      // eslint-disable-next-line @typescript-eslint/no-non-null-assertion
      collector.init(collectorConfig, this.windowRef.nativeWindow!);

      this.subscription.add(
        this.mapEvents(events).subscribe((event) => {
          if (collectorConfig.debug) {
            this.logger.log(
              `🎤 Pushing the following event to ${tmsCollectorConfig}: `,
```

```
            event
          );
        }

        event = collector.map?.(event) ?? event;
        collector.pushEvent(
          collectorConfig,
          // eslint-disable-next-line @typescript-eslint/no-non-null-assertion
          this.windowRef.nativeWindow!,
          event
        );
      })
    );
  }
}
// ...
protected mapEvents<T extends CxEvent>(
  events: Observable<T>[]
): Observable<T> {
  return merge(...events);
}
```

5.5 GtmCollectorService

TS

```
// feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L25-L62
  init(config: GtmCollectorConfig, windowObject: WindowObject): void {
    const dataLayerProperty = config.dataLayerProperty ?? 'dataLayer';
    windowObject[dataLayerProperty] = windowObject[dataLayerProperty] ?? [];

    if (config.gtmId) {
      (function (
        w: WindowObject,
        d: Document,
        s: string,
        l: string,
        i: string
      ) {
        w[l] = w[l] || [];
        w[l].push({ 'gtm.start': new Date().getTime(), event: 'gtm.js' });
        const f = d.getElementsByTagName(s)[0];
        const j = d.createElement(s) as HTMLScriptElement;
        const dl = l !== 'dataLayer' ? '&l=' + l : '';
        j.async = true;
        j.src = 'https://www.googletagmanager.com/gtm.js?id=' + i + dl;
        f.parentNode?.insertBefore(j, f);
      })(
        windowObject,
        this.winRef.document,
```

```
        'script',
        dataLayerProperty,
        config.gtmId
      );
    }
  }

  pushEvent<T extends CxEvent>(
    config: TmsCollectorConfig,
    windowObject: WindowObject,
    event: T | any
  ): void {
    const dataLayerProperty = config.dataLayerProperty ?? 'dataLayer';
    windowObject[dataLayerProperty].push(event);
  }
```

5.6 AepCollectorService

TS

```
// feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L25-L44
  init(config: AepCollectorConfig, windowObject: WindowObject): void {
    const dataLayerProperty = config.dataLayerProperty ?? 'digitalData';
    windowObject[dataLayerProperty] = windowObject[dataLayerProperty] ?? {};

    if (config.scriptUrl) {
      this.scriptLoader.embedScript({ src: config.scriptUrl });
    }
  }

  pushEvent<T extends CxEvent>(
    config: TmsCollectorConfig,
    windowObject: WindowObject,
    event: T | any
  ): void {
    const dataLayerProperty = config.dataLayerProperty ?? 'digitalData';
    windowObject[dataLayerProperty] = {
      ...windowObject[dataLayerProperty],
      ...event,
    };
  }
```

5.7 Configurazione TMS della demo

TS

```
// projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts:L22-L51
@NgModule({
  imports: [
    BaseTmsModule.forRoot(),
```

```
    GtmModule,
    AepModule,
    PersonalizationRootModule,
  ],
  providers: [
    provideConfig(<TmsConfig>{
      tagManager: {
        gtm: {
          events: [NavigationEvent, CartAddEntrySuccessEvent],
        },
        aep: {
          events: [NavigationEvent, CartRemoveEntrySuccessEvent],
        },
      },
    }),
    provideConfig(<CmsConfig>{
      featureModules: {
        [PERSONALIZATION_FEATURE]: {
          module: () =>
            import('@spartacus/tracking/personalization').then(
              (m) => m.PersonalizationModule
            ),
        },
      },
    }),
  ],
})
```

5.8 OccPersonalizationIdInterceptor

TS

```
// feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-
id.interceptor.ts:L29-L98
  constructor(
    private config: PersonalizationConfig,
    private occEndpoints: OccEndpointsService,
    private winRef: WindowRef
  ) {
    if (this.winRef.isBrowser()) {
      this.enabled =
        (this.winRef.localStorage && this.config.personalization?.enabled) ||
        false;

      if (this.enabled) {
        if (!this.config.personalization?.httpHeaderName && isDevMode()) {
          this.logger.warn(
            `There is no httpHeaderName configured in Personalization`
          );
        }
```

```
      this.requestHeader =
        this.config.personalization?.httpHeaderName?.id.toLowerCase();
      this.personalizationId = this.winRef.localStorage?.getItem(
        this.PERSONALIZATION_ID_KEY
      );
    } else if (
      this.winRef.localStorage?.getItem(this.PERSONALIZATION_ID_KEY)
    ) {
      this.winRef.localStorage.removeItem(this.PERSONALIZATION_ID_KEY);
    }
  }
}

intercept(
  request: HttpRequest<any>,
  next: HttpHandler
): Observable<HttpEvent<any>> {
  if (!this.enabled) {
    return next.handle(request);
  }

  if (
    this.requestHeader &&
    this.personalizationId &&
    request.url.includes(this.occEndpoints.getBaseUrl())
  ) {
    request = request.clone({
      setHeaders: {
        [this.requestHeader]: this.personalizationId,
      },
    });
  }

  return next.handle(request).pipe(
    tap((event) => {
      if (
        event instanceof HttpResponse &&
        this.requestHeader &&
        event.headers.keys().includes(this.requestHeader)
      ) {
        const receivedId = event.headers.get(this.requestHeader);
        if (this.personalizationId !== receivedId) {
          this.personalizationId = receivedId;
          if (this.personalizationId) {
            this.winRef.localStorage?.setItem(
              this.PERSONALIZATION_ID_KEY,
              this.personalizationId
            );
          }
```

```
        }
      }
    })
  );
}
```

5.9 PersonalizationContextService

```ts
// feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L30-L69
  getPersonalizationContext(): Observable<PersonalizationContext | undefined> {
    if (!this.config.personalization?.context) {
      if (isDevMode()) {
        this.logger.warn(`There is no context configured in Personalization.`);
      }
      return EMPTY;
    } else {
      const context = this.config.personalization.context;
      return this.cmsService.getCurrentPage().pipe(
        filter<Page>(Boolean),
        map((page: Page) => page.slots?.[context.slotPosition]),
        filter<any>(Boolean),
        map((slot) => {
          const scriptComponent = slot.components?.find(
            (i: ContentSlotComponentData) => i.uid === context.componentId
          );
          return this.buildPersonalizationContext(
            scriptComponent?.properties?.script?.data
          );
        })
      );
    }
  }

  private buildPersonalizationContext(
    data: string
  ): PersonalizationContext | undefined {
    if (data) {
      const context = JSON.parse(atob(data));
      context.actions.forEach((action: any) => {
        Object.keys(action).forEach((key) => {
          action[key] = atob(action[key]);
        });
      });
      for (let i = 0; i < context.segments.length; i++) {
        context.segments[i] = atob(context.segments[i]);
      }
```

```
      return context;
    }
  }
```

5.10 SmartEditRootModule

```ts
// feature-libs/smartedit/root/smart-edit-root.module.ts:L13-L34
export function smartEditFactory(
  smartEditLauncherService: SmartEditLauncherService
): () => void {
  const isReady = () => {
    smartEditLauncherService.load();
  };
  return isReady;
}

@NgModule({
  providers: [
    ...interceptors,
    provideDefaultConfig(defaultSmartEditConfig),
    {
      provide: APP_INITIALIZER,
      useFactory: smartEditFactory,
      deps: [SmartEditLauncherService],
      multi: true,
    },
  ],
})
export class SmartEditRootModule {}
```

5.11 SmartEditLauncherService: load e rilevamento

```ts
// feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L43-L96
  load(): void {
    if (this.isLaunchedInSmartEdit()) {
      this.featureModulesService.resolveFeature(SMART_EDIT_FEATURE).subscribe();

      this.scriptLoader?.embedScript({
        src: 'assets/webApplicationInjector.js',
        params: undefined,
        attributes: {
          id: 'text/smartedit-injector',
          'data-smartedit-allow-origin': this.config.smartEdit?.allowOrigin,
        },
      });
    }
```

```
  }

  /**
   * Indicates whether Spartacus is launched in SmartEdit
   */
  isLaunchedInSmartEdit(): boolean {
    const [path, params] = this.location.path().split('?');
    const cmsToken = params
      ?.split('&')
      .find((param) => param.startsWith('cmsTicketId='));
    const cmsTicketId = cmsToken?.split('=')[1];

    return (
      this.isInitialSmartEditPage(path, cmsTicketId) ||
      this.isFullPageRedirectInSmartEdit()
    );
  }

  private isInitialSmartEditPage(
    path: string,
    cmsTicketId: string | undefined
  ): boolean {
    // When both the SmartEdit cmsTicketId and the storefrontPreviewRoute values are found in the URL,
store the cmsTicketId in sessionStorage.
    // so it survives full-page navigation (e.g. CDC OIDC redirect flow).
    if (
      path.split('/').pop() === this.config.smartEdit?.storefrontPreviewRoute &&
      !!cmsTicketId
    ) {
      this.persistCmsTicketId(cmsTicketId);
      this._cmsTicketId = cmsTicketId;
      return true;
    }
    return false;
  }

  private isFullPageRedirectInSmartEdit(): boolean {
    // Fall back to sessionStorage for scenarios where a full-page redirect
    // drops the cmsTicketId and other SmartEdit context from the URL.
    this._cmsTicketId = this.restoreCmsTicketId();
    return !!this._cmsTicketId;
  }
```

5.12 `CmsTicketInterceptor.intercept`

TS

```
// feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L25-L45
  intercept(
    request: HttpRequest<any>,
```

```
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    const cmsTicketId = this.service.cmsTicketId;
    if (!cmsTicketId) {
      return next.handle(request);
    }
    if (request.url.includes('/productList')) {
      return this.setRequestForProductListPage(request, next, cmsTicketId);
    }
    if (request.url.includes('/cms/') || request.url.includes('/products/')) {
      request = request.clone({
        setParams: {
          cmsTicketId,
        },
      });
    }

    return next.handle(request);
  }
```

5.13 `SmartEditCoreModule` e i decoratori

TS

```
// feature-libs/smartedit/core/smart-edit-core.module.ts:L11-L18
@NgModule({
  providers: [...smartEditDecorators],
})
export class SmartEditCoreModule {
  constructor(private smartEditService: SmartEditService) {
    this.smartEditService.processCmsPage();
  }
}
```

TS

```
// feature-libs/smartedit/core/decorators/index.ts:L12-L23
export const smartEditDecorators: Provider[] = [
  {
    provide: ComponentDecorator,
    useExisting: SmartEditComponentDecorator,
    multi: true,
  },
  {
    provide: SlotDecorator,
    useExisting: SmartEditSlotDecorator,
    multi: true,
  },
];
```

TS

```ts
// core-libs/core/src/cms/services/dynamic-attribute.service.ts:L24-L48
  private componentDecorators$ = this.unifiedInjector
    .getMulti(ComponentDecorator)
    .pipe(shareReplay(1));

  private slotDecorators$ = this.unifiedInjector
    .getMulti(SlotDecorator)
    .pipe(shareReplay(1));

  constructor(protected unifiedInjector: UnifiedInjector) {}
  // ...
  addAttributesToComponent(
    element: Element,
    renderer: Renderer2,
    componentData?: ContentSlotComponentData
  ) {
    (getLastValueSync(this.componentDecorators$) || []).forEach((decorator) =>
      decorator.decorate(element, renderer, componentData)
    );
  }
```

5.14 `SmartEditService`: contratto e re-render

TS

```ts
// feature-libs/smartedit/core/services/smart-edit.service.ts:L126-L191
  protected renderComponent(
    componentId: string,
    componentType?: string,
    parentId?: string
  ): boolean {
    if (componentId) {
      this.zone.run(() => {
        // without parentId, it is slot
        if (!parentId) {
          // Clear all cached component data to ensure child components are also refreshed
          this.cmsService.clearComponentState();
          if (this._currentPageId) {
            this.cmsService.refreshPageById(this._currentPageId);
          } else {
            this.cmsService.refreshLatestPage();
          }
        } else if (componentType) {
          this.cmsService.refreshComponent(componentId);
        }
      });
    }

    return true;
  }
```

```
protected reprocessPage() {
  // TODO: reprocess page API
}

/**
 * add smartedit HTML markup contract
 */
public addSmartEditContract(
  element: Element,
  renderer: Renderer2,
  properties: any
): void {
  if (properties) {
    // check each group of properties, e.g. smartedit
    Object.keys(properties).forEach((group) => {
      const name = 'data-' + group + '-';
      const groupProps = properties[group];

      // check each property in the group
      Object.keys(groupProps).forEach((propName) => {
        const propValue = groupProps[propName];
        if (propName === 'classes') {
          const classes = propValue.split(' ');
          classes.forEach((classItem: string) => {
            renderer.addClass(element, classItem);
          });
        } else {
          renderer.setAttribute(
            element,
            name +
              propName
                .split(/(?=[A-Z])/)
                .join('-')
                .toLowerCase(),
            propValue
          );
        }
      });
    });
  }
}
```

5.15 QualtricsLoaderService

```ts
// feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L55-L71
  private qsiLoaded$: Observable<any> =
    isPlatformBrowser(this.platformId) && this.window
```

```
      ? fromEvent(this.window, QUALTRICS_EVENT_NAME)
      : EMPTY;

  /**
   * Emits the Qualtrics Site Intercept (QSI) JavaScript API whenever available.
   *
   * The API is emitted when the JavaScript resource holding this API is fully loaded.
   * The API is also stored locally in the service, in case it's required later on.
   */
  protected qsi$: Observable<any> = this.qsiLoaded$.pipe(
    switchMap(() => this.isDataLoaded()),
    map((dataLoaded: boolean) => (dataLoaded ? this.window?.QSI : EMPTY)),
    filter((qsi: any) => Boolean(qsi)),
    tap((qsi: any) => (this.qsiApi = qsi))
  );
```

TS

```
// feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L123-L149
  protected run(reload = false): void {
    if (!this.qsiApi?.API) {
      if (isDevMode()) {
        this.logger.log('The QSI api is not available');
      }
      return;
    }

    if (reload) {
      // Removes any currently displaying creatives
      this.qsiApi.API.unload();
    }

    // Starts the intercept code evaluation right after loading the Site Intercept
    // code for any defined intercepts or creatives
    this.qsiApi.API.load().done(this.qsiApi.API.run());
  }

  /**
   * This logic exist in order to let the client(s) add their own logic to wait for any kind of page
data.
   * You can observe any data in this method.
   *
   * Defaults to true.
   */
  protected isDataLoaded(): Observable<boolean> {
    return of(true);
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
tagManager	TmsCollectors (dizionario)	voci gtm/aep dai moduli vendor	feature-libs/tracking/tms/core/config/tms-config.ts:L42-L47	feature-libs/tracking/tms/core/services/tms.service.ts:L51-L57
tagManager.<tms>.collector	Type<TmsCollector>	GtmCollectorService / AepCollectorService	feature-libs/tracking/tms/core/config/tms-config.ts:L28-L28; default feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L20-L26, feature-libs/tracking/tms/aep/config/default-aep.config.ts:L20-L26	feature-libs/tracking/tms/core/services/tms.service.ts:L59-L73
tagManager.<tms>.events	AbstractType<CxEvent>[]	nessuno ([] dallo schematic)	feature-libs/tracking/tms/core/config/tms-config.ts:L24-L24	feature-libs/tracking/tms/core/services/tms.service.ts:L68-L70
tagManager.<tms>.debug	boolean	undefined	feature-libs/tracking/tms/core/config/tms-config.ts:L16-L16	feature-libs/tracking/tms/core/services/tms.service.ts:L79-L84
tagManager.<tms>.dataLayerProperty	string	'dataLayer' (GTM) / 'digitalData' (AEP)	feature-libs/tracking/tms/core/config/tms-config.ts:L20-L20	feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L26-L26, feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L26-L26
tagManager.gtm.gtmId	string	undefined	feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L10-L12	feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L29-L51
tagManager.aep.scriptUrl	string	undefined	feature-libs/tracking/tms/aep/config/default-aep.config.ts:L10-L12	feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L29-L31
personalization.enabled	boolean	false	feature-libs/tracking/personalization/root/config/personalization-config.ts:L16-L16; default feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L11-L11	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L35-L37 (idem nel Time interceptor)

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
personalization.httpHeaderName.id	string	'Occ-Personalization-Id'	feature-libs/tracking/personalization/root/config/personalization-config.ts:L17-L20; feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L13-L13	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L45-L46
personalization.httpHeaderName.timestamp	string	'Occ-Personalization-Time'	idem; feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L14-L14	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L45-L46
personalization.context.slotPosition	string	'PlaceholderContentSlot'	feature-libs/tracking/personalization/root/config/personalization-config.ts:L21-L24; feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L17-L17	feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L40-L40
personalization.context.componentId	string	'PersonalizationScriptComponent'	idem; feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L18-L18	feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L44-L44
featureModules.personalization.cmsComponents	string[]	['PersonalizationScriptComponent']	feature-libs/tracking/personalization/root/personalization-root.module.ts:L18-L28	core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76
featureModules.personalization.module	() => Promise<any>	nessun default (solo app)	demo projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts:L40-L48	core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L68
smartEdit.storefrontPreviewRoute	string	'cx-preview'	feature-libs/smartedit/root/config/smart-edit-config.ts:L15-L18; feature-libs/smartedit/root/config/default-smart-edit-config.ts:L11-L11	feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L81-L81

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
smartEdit.allowOrigin	string	'localhost:9002'	idem; feature-libs/smartedit/root/config/default-smart-edit-config.ts:L12-L12	feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L52-L52
featureModules.smartEdit	FeatureModuleConfig	nessun default (solo app)	demo projects/storefrontapp/src/app/spartacus/features/smartedit/smartedit-feature.module.ts:L17-L24	feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L45-L45
qualtrics.scriptSource	string	undefined (qualtrics: {})	feature-libs/qualtrics/components/qualtrics-loader/config/qualtrics-config.ts:L22-L29; feature-libs/qualtrics/components/qualtrics-loader/config/default-qualtrics-config.ts:L9-L11	feature-libs/qualtrics/components/qualtrics-loader/qualtrics.component.ts:L26-L27
featureModules.qualtrics.cmsComponents	string[]	['QualtricsComponent']	feature-libs/qualtrics/root/qualtrics-root.module.ts:L12-L22	core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76

Chiavi di storage del browser:

CHIAVE	STORAGE	SCRITTA/LETTA IN
personalization-id	localStorage	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L25-L25
personalization-time	localStorage	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L25-L25
smartedit.cmsTicketId	sessionStorage	feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L28-L28

Variabili globali su window:

PROPRIETÀ	SCRITTA DA
window.dataLayer (o dataLayerProperty)	feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L26-L27
window.digitalData (o dataLayerProperty)	feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L26-L27
window.smartedit.renderComponent, window.smartedit.reprocessPage	feature-libs/smartedit/core/services/smart-edit.service.ts:L41-L51
window.QSI (letta, scritta dallo script Qualtrics)	feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L68-L68

7. Commenti e note notevoli dal sorgente

- `/** Should be enabled in development mode only */` sul flag `debug` del collector (feature-libs/tracking/tms/core/config/tms-config.ts:L15-L16): il log con emoji 🎤 stampa l'oggetto evento completo a ogni push.
- `/** Called only once to start collecting and dispatching events */` (feature-libs/tracking/tms/core/services/tms.service.ts:L43-L46): non c'è protezione contro chiamate ripetute; è l'`APP_INITIALIZER` a garantire che avvenga una volta.
- Il JSDoc di `mapEvents` documenta un parametro `@param collector` che la firma non ha più (feature-libs/tracking/tms/core/services/tms.service.ts:L98-L106): segno che in passato il mapping era per-collector; oggi è un semplice `merge` e il mapping per-collector si fa con `TmsCollector.map`.
- `// TODO: Inline this factory when we start releasing Ivy compiled libraries` (feature-libs/tracking/personalization/root/personalization-root.module.ts:L17-L17, identico in feature-libs/qualtrics/root/qualtrics-root.module.ts:L11-L11): residuo dell'era View Engine, quando le factory dovevano essere funzioni esportate per l'AOT.
- `// CHECK SONAR` a fine file (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L101-L101), probabilmente legato alla duplicazione quasi totale con l'interceptor Id.
- `* Public API Surface of personalizatiion` (refuso) (feature-libs/tracking/personalization/public_api.ts:L8-L8).
- `// before adding contract to page, we need redirect to that page` (feature-libs/smartedit/core/services/smart-edit.service.ts:L71-L71) e `// only the first page is the smartedit preview page` (feature-libs/smartedit/core/services/smart-edit.service.ts:L100-L100).
- `// Clear all cached component data to ensure child components are also refreshed` (feature-libs/smartedit/core/services/smart-edit.service.ts:L135-L135): spiega perché un edit su uno slot svuota l'intera cache componenti invece dei soli figli.
- `// TODO: reprocess page API` (feature-libs/smartedit/core/services/smart-edit.service.ts:L152-L152).
- I commenti sul `sessionStorage` per i redirect full-page (CDC OIDC) e sul `SecurityError` in SSR (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L78-L79, feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L92-L93, feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L99-L99).
- `// Starts the intercept code evaluation right after loading the Site Intercept code for any defined intercepts or creatives` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L136-L138): l'intento è chiaro, l'implementazione però passa il *risultato* di `run()` a `done` (vedi trappola T-Q2).
- Nessun `@deprecated`/`FIXME` nei file del capitolo; il token `APP_INITIALIZER` usato da `BaseTmsModule` e `SmartEditRootModule` è però deprecato in Angular (angular:packages/core/src/application/application_init.ts:L38-L38).

8. Extension point

EXTENSION POINT	TIPO	DEFINIZIONE	EXPORT
`TmsConfig` / `tagManager`	config astratta	feature-libs/tracking/tms/core/config/tms-config.ts:L38-L47	feature-libs/tracking/tms/core/public_api.ts:L8-L8
`TmsCollectors`	interfaccia augmentabile (nuovi TMS)	feature-libs/tracking/tms/core/config/tms-config.ts:L31-L33	feature-libs/tracking/tms/core/public_api.ts:L12-L14
`TmsCollector`	interfaccia per collector custom (`init`, `pushEvent`, `map?`)	feature-libs/tracking/tms/core/model/tms.model.ts:L21-L40	feature-libs/tracking/tms/core/public_api.ts:L9-L9
`TmsCollectorConfig.collector`	sostituzione del collector di un TMS esistente	feature-libs/tracking/tms/core/config/tms-config.ts:L28-L28	idem
`TmsService.mapEvents`	metodo protected	feature-libs/tracking/tms/core/services/tms.service.ts:L104-L108	feature-libs/tracking/tms/core/public_api.ts:L10-L10
`GtmCollectorService` / `AepCollectorService`	servizi estendibili (dipendenze protected)	feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L19-L21, feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L19-L21	feature-libs/tracking/tms/gtm/public_api.ts:L8-L8, feature-libs/tracking/tms/aep/public_api.ts:L8-L8
`PersonalizationConfig`	config astratta	feature-libs/tracking/personalization/root/config/personalization-config.ts:L10-L26	feature-libs/tracking/personalization/root/public_api.ts:L7-L7
`OccPersonalizationIdInterceptor` / `OccPersonalizationTimeInterceptor`	classi esportate (stato `private`, chiave e logger `protected`)	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L20-L27	feature-libs/tracking/personalization/root/public_api.ts:L10-L11
`PersonalizationContextService`	servizio root (decodifica `private`)	feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L19-L70	feature-libs/tracking/personalization/core/public_api.ts:L9-L9
`SmartEditConfig`	config astratta	feature-libs/smartedit/root/config/smart-edit-config.ts:L10-L19	feature-libs/smartedit/root/public_api.ts:L7-L7
`SmartEditLauncherService`	servizio (`isLaunchedInSmartEdit` pubblico, helper `private`)	feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L21-L119	feature-libs/smartedit/root/public_api.ts:L9-L9
`CmsTicketInterceptor.setRequestForProductListPage`	metodo protected	feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L47-L72	feature-libs/smartedit/root/public_api.ts:L11-L11

EXTENSION POINT	TIPO	DEFINIZIONE	EXPORT
`SmartEditService`	protected `addPageContract`, `goToPreviewPage`, `renderComponent`, `reprocessPage`; pubblici `processCmsPage`, `addSmartEditContract`	feature-libs/smartedit/core/services/smart-edit.service.ts:L55-L191	feature-libs/smartedit/core/public_api.ts:L7-L7
`ComponentDecorator` / `SlotDecorator`	classi astratte multi-provider di core	core-libs/core/src/cms/decorators/component-decorator.ts:L10-L23, core-libs/core/src/cms/decorators/slot-decorator.ts:L10-L23	core-libs/core/src/cms/index.ts:L10-L10
`QualtricsConfig`	config astratta	feature-libs/qualtrics/components/qualtrics-loader/config/qualtrics-config.ts:L14-L30	feature-libs/qualtrics/components/qualtrics-loader/index.ts:L7-L7
`QualtricsLoaderService.isDataLoaded()`	hook protected (default `of(true)`)	feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L147-L149	feature-libs/qualtrics/components/qualtrics-loader/index.ts:L8-L8
Mapping CMS `QualtricsComponent` / `QualtricsEmbeddedFeedbackComponent`	`cmsComponents` sovrascrivibili	feature-libs/qualtrics/components/qualtrics-components.module.ts:L21-L30	feature-libs/qualtrics/components/public_api.ts:L7-L8
`PERSONALIZATION_FEATURE`, `SMART_EDIT_FEATURE`, `QUALTRICS_FEATURE`	costanti per `featureModules`	feature-libs/tracking/personalization/root/feature-name.ts:L7-L7, feature-libs/smartedit/root/feature-name.ts:L7-L7, feature-libs/qualtrics/root/feature-name.ts:L7-L7	public_api dei rispettivi root

Nota: `GtmCollectorConfig` e `AepCollectorConfig` (con `gtmId`/`scriptUrl`) **non** sono esportate dai public API di `tms/gtm` e `tms/aep`, che esportano solo module e services (feature-libs/tracking/tms/gtm/public_api.ts:L7-L8); le chiavi sono comunque tipizzate grazie all'augmentation di `TmsCollectors` dichiarata nel file di default config (feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L14-L18).

8.1 Schema: un collector per un TMS custom

Esempio illustrativo (non presente nel repo), che usa solo simboli esportati da `@spartacus/tracking/tms/core` (`TmsCollector`, `TmsCollectorConfig`, `WindowObject`, `TmsConfig`) e dall'event bus di core:

```ts
// esempio — non nel repo
@Injectable({ providedIn: 'root' }) // oppure fornito nei providers di un modulo eager: TmsService lo risolve dall'injector root (T3)
export class MyTmsCollector implements TmsCollector {
  init(config: TmsCollectorConfig, w: WindowObject): void {
    const name = config.dataLayerProperty ?? 'myLayer';
    w[name] = w[name] ?? [];
  }
```

```
  map<T extends CxEvent>(event: T): object {
    return { name: (event.constructor as any).type, payload: event };
  }
  pushEvent<T extends CxEvent>(config: TmsCollectorConfig, w: WindowObject, event: T | any): void {
    w[config.dataLayerProperty ?? 'myLayer'].push(event);
  }
}

// nel modulo dell'app, insieme a BaseTmsModule.forRoot()
provideConfig(<TmsConfig>{
  tagManager: { myTms: { collector: MyTmsCollector, events: [NavigationEvent] } },
});
```

Il funzionamento poggia su feature-libs/tracking/tms/core/services/tms.service.ts:L51-L94 (qualsiasi chiave di `tagManager` con un `collector` viene considerata). Per sostituire solo la trasformazione degli eventi di GTM basta estendere `GtmCollectorService` aggiungendo `map` e impostarla come `tagManager.gtm.collector`.

8.2 Schema: attendere dati applicativi prima di Qualtrics

TS

```
// esempio — non nel repo
@Injectable({ providedIn: 'root' })
export class MyQualtricsLoaderService extends QualtricsLoaderService {
  // inject() solo in un field initializer: isDataLoaded() gira dentro switchMap, fuori dall'injection context
  protected activeCart = inject(ActiveCartFacade);
  protected isDataLoaded(): Observable<boolean> {
    return this.activeCart.isStable(); // qualsiasi Observable<boolean>
  }
}
// providers: [{ provide: QualtricsLoaderService, useClass: MyQualtricsLoaderService }]
```

Hook documentato a feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L141-L149. Attenzione alla trappola T-Q1: se l'Observable emette `false`, il servizio finisce per memorizzare `EMPTY` come API.

8.3 Ricette correlate (Parte III)

- Override di config (`provideConfig`), sostituzione di servizi via DI (`useClass`), interceptor ed eventi custom: 50 — Ricette: configurazione, DI, interceptor, eventi.
- Sostituire il mapping CMS di `QualtricsComponent`/`QualtricsEmbeddedFeedbackComponent` o creare un componente CMS che faccia da contenitore: 51 — Ricette: componenti CMS, outlet, route e layout.
- Aggiungere/organizzare feature lazy (`featureModules`) come fanno Personalization, SmartEdit e Qualtrics: 53 — Ricette: feature, cart, checkout, stili, i18n.
- Estendere i servizi protetti (`SmartEditService`, `QualtricsLoaderService`, collector) in modo robusto agli upgrade: 54 — Upgrade-safe.

9. Trappole, casi limite, comportamenti controintuitivi

Tracking / TMS

- **T1 — Config letta una sola volta.** `collect()` legge `tagManager` una volta, nell'`APP_INITIALIZER`: collector o eventi aggiunti dopo (ad esempio config fornita da un modulo lazy) non vengono considerati (feature-libs/tracking/tms/core/base-tms.module.ts:L13-L16, feature-libs/tracking/tms/core/services/tms.service.ts:L46-L51).
- **T2 — Collector senza eventi.** Senza `events`, il collector viene comunque inizializzato (data layer creato, script GTM/AEP caricato) ma non riceve nulla; lo schematic genera proprio `events: []` (feature-libs/tracking/tms/core/services/tms.service.ts:L68-L75, core-libs/schematics/src/shared/lib-configs/tracking-schematics-config.ts:L89-L95).
- **T3 — Collector nell'injector root.** `this.injector.get(collector)` usa l'injector di `TmsService` (root): un collector non disponibile nel root injector (es. fornito solo in un modulo lazy) genera un errore di DI (feature-libs/tracking/tms/core/services/tms.service.ts:L71-L73).
- **T4 — AEP sovrascrive.** `pushEvent` di AEP fa un merge shallow sull'oggetto: proprietà omonime di eventi diversi si sovrascrivono e la storia non è conservata, a differenza dell'array GTM (feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L39-L43 vs feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L60-L61). Inoltre, essendo un nuovo oggetto, chi conserva un riferimento al vecchio `digitalData` non vede gli aggiornamenti.
- **T5 — Data layer GTM non-array.** `pushEvent` assume che `window[dataLayerProperty]` sia un array (creato in `init`); se un altro script lo sostituisce con un non-array, `push` fallisce (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L27-L27, feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L61-L61).
- **T6 — Nessun `<script>` nel documento.** Lo snippet GTM inserisce lo script prima di `getElementsByTagName('script')[0]`; se non esiste nessun `<script>`, `f` è undefined e `f.parentNode` lancia — l'optional chaining è solo su `parentNode` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L39-L44). In un'app Angular buildata gli script del bundle ci sono, quindi è un caso essenzialmente teorico (test o ambienti particolari).
- **T7 — Demo senza vendor.** Nella demo GTM/AEP sono configurati senza `gtmId`/`scriptUrl`: i data layer si popolano ma nessuno script vendor è caricato (projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts:L30-L39). È utile per ispezionare `window.dataLayer` in console, non per vedere dati in GTM.
- **T8 — Eventi interi nel data layer.** Senza `map`, nel data layer finisce l'istanza `CxEvent` così com'è (es. `NavigationEvent` con `context`, `semanticRoute`, `url`: core-libs/storefront/events/navigation/navigation.event.ts:L13-L29). Un evento carrello può contenere dati che non si vogliono inviare a un TMS: valutare un `map` che selezioni i campi (feature-libs/tracking/tms/core/services/tms.service.ts:L86-L86).

Personalization

- **T9 — Niente header in SSR.** `enabled` resta `false` fuori dal browser, quindi le richieste OCC lato server non portano gli header di personalizzazione (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L24-L24, feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L34-L37). La pagina renderizzata dal server è quindi non personalizzata.
- **T10 — `httpHeaderName` parziale.** `httpHeaderName?.id.toLowerCase()` ha l'optional chaining solo fino a `httpHeaderName?.`: se `httpHeaderName` c'è ma `id` manca, `TypeError` nel costruttore; il warning in dev copre

solo l'assenza di `httpHeaderName` (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L40-L46). Poiché `id` e `timestamp` sono tipizzati come obbligatori (feature-libs/tracking/personalization/root/config/personalization-config.ts:L17-L20), TypeScript protegge solo se la config è tipizzata.

- **T11 — Casing degli header.** Il controllo `event.headers.keys().includes(this.requestHeader)` confronta col nome minuscolo, mentre `HttpHeaders.keys()` restituisce i nomi nel casing originale (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L83-L83, angular:packages/common/http/src/headers.ts:L115-L119, angular:packages/common/http/src/headers.ts:L172-L176). Funziona perché XHR/fetch espongono gli header di risposta in minuscolo. DA VERIFICARE per backend/proxy e backend HTTP specifici (non verificabile nel repo).
- **T12 — Filtro URL per sottostringa.** `request.url.includes(occEndpoints.getBaseUrl())`: richieste non-OCC il cui URL contiene la base URL riceverebbero gli header (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L69-L69).
- **T13 — Payload malformato.** `buildPersonalizationContext` usa `JSON.parse(atob(...))` senza try/catch: un payload malformato genera un errore nello stream, che termina (feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L57-L66).
- **T14 — Feature lazy senza mapping.** `PersonalizationModule`/`PersonalizationCoreModule` non mappano `PersonalizationScriptComponent` in `cmsComponents` (feature-libs/tracking/personalization/core/personalization-core.module.ts:L9-L10). Quando una pagina contiene quel componente, `CmsFeaturesService.getCmsMapping` risolve la feature e restituisce `componentsMappings?.[type]`, cioè undefined (core-libs/storefront/cms-structure/services/cms-features.service.ts:L89-L103); `CmsComponentsService` scarta gli undefined con `filter(isNotUndefined)` e unisce i resolver con `forkJoin` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L109-L110, core-libs/storefront/cms-structure/services/cms-components.service.ts:L117-L141). In RxJS un `forkJoin` con una sorgente che completa senza emettere completa senza emettere (deps:rxjs-7.8.2/package/src/internal/observable/forkJoin.ts:L77-L80). `determineMappings` è sottoscritto da `ComponentWrapperDirective.ngOnInit` (che lancia il componente solo nella callback `next`) e da `CmsPageGuardService.canActivatePage` (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L67-L79, core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L71-L75); le guard del router applicano `first()` al risultato (angular:packages/router/src/operators/check_guards.ts:L168-L168). DA VERIFICARE l'effetto concreto a runtime (componente semplicemente non renderizzato, warning "No component implementation found" da core-libs/storefront/cms-structure/services/cms-components.service.ts:L165-L181, o impatto sulla guard di pagina): richiede una pagina CMS reale con `PersonalizationScriptComponent` e non è osservabile senza backend.
- **T15 — Contesto iniziale vuoto per CDS.** CDS parte da `{ segments: undefined, actions: undefined }` e usa `withLatestFrom`: i push event emessi prima che la pagina corrente contenga il componente script portano segmenti/azioni undefined (integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L98-L112).

SmartEdit

- **T16 — `resolveFeature` senza error handler.** `load()` sottoscrive `resolveFeature('smartEdit')` senza gestione errori: se l'app non configura `featureModules.smartEdit.module`, l'errore "No module defined for Feature Module smartEdit" non è gestito (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L45-L45, core-libs/core/src/lazy-loading/feature-modules.service.ts:L47-L52).
- **T17 — Ticket "appiccicoso".** Il ticket salvato in `sessionStorage` non viene mai rimosso nel codice: per tutta la vita del tab, ogni reload è considerato "in SmartEdit" (fallback), con script injector caricato e `cmsTicketId` aggiunto alle chiamate (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L91-L96, feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L109-

L118). In sviluppo, aprire lo storefront nello stesso tab dopo averlo usato in SmartEdit dà comportamenti inattesi: chiudere il tab o cancellare la chiave.

- **T18 — Parsing semplificato.** `isInitialSmartEditPage` confronta solo l'ultimo segmento del path con `storefrontPreviewRoute`, e il valore di `cmsTicketId` non è URL-decoded (`split('=')[1]`) (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L66-L66, feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L80-L83).
- **T19 — Path dell'injector hardcoded.** Lo `src` è sempre il path relativo hardcoded `assets/webApplicationInjector.js`: l'app deve copiare l'asset — lo fa lo schematic in `angular.json` (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L47-L48, core-libs/schematics/src/shared/lib-configs/smartedit-schematics-config.ts:L48-L51).
- **T20 — SSR.** In SSR `isLaunchedInSmartEdit` può essere vero solo per la pagina iniziale (il fallback da `sessionStorage` è saltato): in quel caso `load()` risolve comunque la feature e chiama `embedScript`, che senza callback non è bloccato in SSR e non controlla i duplicati lato server (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L43-L55, core-libs/core/src/util/script-loader.service.ts:L55-L63).
- **T21 — `id` come proprietà.** `id: 'text/smartedit-injector'` non inizia con `data-`, quindi `ScriptLoader` lo assegna come proprietà `script.id` (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L50-L53, core-libs/core/src/util/script-loader.service.ts:L73-L80).
- **T22 — Decisione per sottostringa.** `CmsTicketInterceptor` decide su `/cms/`, `/products/`, `/productList` nell'URL, non sugli endpoint configurati: endpoint OCC personalizzati con path diversi non ricevono il ticket (feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L33-L42).
- **T23 — Classi del `<body>` cancellate.** `addPageContract` rimuove tutte le classi del `<body>`, comprese quelle aggiunte dall'app o da altre librerie (feature-libs/smartedit/core/services/smart-edit.service.ts:L85-L90). Accade a ogni cambio pagina, ma solo in SmartEdit.
- **T24 — `reprocessPage` non legato.** `window.smartedit.reprocessPage = this.reprocessPage` assegna il metodo senza `bind` ed è comunque un no-op (feature-libs/smartedit/core/services/smart-edit.service.ts:L51-L51, feature-libs/smartedit/core/services/smart-edit.service.ts:L151-L153). Chi lo sovrascrive in una sottoclasse non può usare `this`.
- **T25 — `renderComponent` silenzioso.** Con `parentId` ma senza `componentType` non accade nulla, e il metodo restituisce sempre `true` (feature-libs/smartedit/core/services/smart-edit.service.ts:L131-L148).
- **T26 — Subscription annidata.** `processCmsPage` contiene una subscribe annidata senza unsubscribe su `getCurrentPage()` che vive quanto l'app (feature-libs/smartedit/core/services/smart-edit.service.ts:L66-L74).
- **T27 — Preview solo con codici del BaseSite.** `goToPreviewPage` naviga solo per la prima pagina e solo se il BaseSite definisce i codici di preview; la route `product` riceve `name: ''` (feature-libs/smartedit/core/services/smart-edit.service.ts:L101-L119).

Qualtrics

- **T-Q1 — `EMPTY` scambiato per l'API.** Se `isDataLoaded()` emette `false`, `map` restituisce `EMPTY` (un Observable, quindi truthy), che supera `filter(Boolean)` e diventa `qsiApi`; `run()` trova `qsiApi.API` undefined e logga "The QSI api is not available" (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L68-L70, feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L124-L129). Un successivo `addScript` → `run(true)` userebbe lo stesso `qsiApi` errato finché non arriva un nuovo `qsi_js_loaded` con dati pronti.
- **T-Q2 — `done(run())`.** `this.qsiApi.API.load().done(this.qsiApi.API.run())` invoca `run()` immediatamente e passa il suo valore di ritorno a `done`, invece di una callback (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L138-L138): `run` parte prima che `load` abbia finito.

- **T-Q3 — Script anche in SSR.** `QualtricsComponent` agisce nel costruttore: anche in SSR chiama `addScript` → `embedScript`, e lo script Qualtrics finisce nell'HTML generato dal server (feature-libs/qualtrics/components/qualtrics-loader/qualtrics.component.ts:L22-L28, core-libs/core/src/util/script-loader.service.ts:L55-L63). Nel browser, durante l'idratazione, `hasScript` trova lo script già presente e `addScript` passa al ramo `run(true)` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L91-L94).
- **T-Q4 — Componente embedded senza feature.** `QualtricsEmbeddedFeedbackComponent` non è nella lista `cmsComponents` della feature `qualtrics` (feature-libs/qualtrics/root/qualtrics-root.module.ts:L14-L18) pur essendo mappato in `QualtricsComponentsModule` (feature-libs/qualtrics/components/qualtrics-components.module.ts:L21-L30). Poiché `CmsFeaturesService` associa a una feature solo i componenti elencati in `cmsComponents` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L72), una pagina che contenga solo il componente embedded non provoca il caricamento lazy. DA VERIFICARE: se un'altra parte dell'app (o una pagina precedente con `QualtricsComponent`) ha già caricato la feature, il mapping potrebbe essere comunque disponibile tramite la config fusa; dipende dall'ordine di navigazione.
- **T-Q5 — Evento perso.** Il listener `qsi_js_loaded` è registrato solo alla prima istanziazione di `QualtricsLoaderService` (root): se l'evento è già scattato prima (script inserito da altri, es. in `index.html`), non viene intercettato, e serve un `addScript` che passi per `run(true)` — che però a sua volta esce perché `qsiApi` non è stato valorizzato (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L77-L83, feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L123-L129).

---

10. Come fare debug

TMS.

- Impostare `debug: true` sulla voce di `tagManager`: ogni push viene loggato con 🎤 `Pushing the following event to <tms>` (feature-libs/tracking/tms/core/services/tms.service.ts:L79-L84). Il logger è `LoggerService`, quindi segue eventuali override del logger (vedi 41 — Errori HTTP, error handling, logger).
- In console del browser: `window.dataLayer` (GTM) o `window.digitalData` (AEP), o il nome configurato in `dataLayerProperty`.
- Nessun evento? Controllare: (a) che `BaseTmsModule.forRoot()` sia importato (senza `forRoot` non c'è initializer: feature-libs/tracking/tms/core/base-tms.module.ts:L18-L19); (b) che la voce abbia un `collector` (altrimenti warning "Skipping the '...'", feature-libs/tracking/tms/core/services/tms.service.ts:L59-L66); (c) che la classe evento in `events` sia esattamente quella emessa — `EventService.get` è per tipo (core-libs/core/src/event/event.service.ts:L90-L96).
- Breakpoint utili: `TmsService.collect` e `GtmCollectorService.pushEvent`.

Personalization.

- DevTools → Application → Local Storage: chiavi `personalization-id` e `personalization-time`. DevTools → Network: header `occ-personalization-id`/`occ-personalization-time` sulle richieste OCC.
- Se le chiavi spariscono a ogni avvio, `personalization.enabled` è false (pulizia a feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L50-L54).
- Warning in dev "There is no httpHeaderName configured in Personalization" e "There is no context configured in Personalization." indicano config mancanti (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L40-L44, feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L31-L35).

- Per il contesto: ispezionare nello store CMS la pagina corrente, slot `PlaceholderContentSlot`, componente `PersonalizationScriptComponent`, `properties.script.data`, e decodificarlo a mano con `JSON.parse(atob(...))`.

SmartEdit.

- Simulare SmartEdit fuori dall'editor aprendo `/<...>/cx-preview?cmsTicketId=<qualsiasi>`: il launcher persiste il ticket e carica la feature (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L80-L87). Verificare in Network che il chunk `@spartacus/smartedit` e lo script `webApplicationInjector.js` (cartella `assets`) vengano scaricati, e che le chiamate `/cms/` portino `cmsTicketId`.
- DevTools → Application → Session Storage: chiave `smartedit.cmsTicketId`. Cancellarla per "uscire" da SmartEdit (T17).
- Ispezionare il DOM: slot e componenti dovrebbero avere attributi `data-smartedit-*` e classi dal contratto (feature-libs/smartedit/core/services/smart-edit.service.ts:L158-L191); se mancano, `properties` dal backend è vuoto (senza ticket valido il backend non le fornisce — DA VERIFICARE lato backend).
- In console: `window.smartedit.renderComponent('<uid>', '<type>', '<parentUid>')` forza il refresh di un componente; senza terzo argomento ricarica la pagina (feature-libs/smartedit/core/services/smart-edit.service.ts:L126-L149).
- Errore "No module defined for Feature Module smartEdit" in console → manca `featureModules.smartEdit.module` (T16).

Qualtrics.

- Warning "We're unable to add the Qualtrics deployment code..." → manca `qualtrics.scriptSource` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics.component.ts:L28-L31).
- Log "The QSI api is not available" → `window.QSI.API` assente al momento di `run` (T-Q1, T-Q5).
- In console: `window.QSI`, e `dispatchEvent(new Event('qsi_js_loaded'))` per simulare l'evento di caricamento letto da `fromEvent` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L55-L58).
- Ricordare che la feature si carica solo se la pagina CMS contiene `QualtricsComponent` (feature-libs/qualtrics/root/qualtrics-root.module.ts:L11-L27).

Test. Le tre librerie hanno `vitest.config.ts` con ambiente `jsdom`, pool `forks` e coverage v8 con soglie 85% (statements/lines/functions) e 65% (branches) (feature-libs/tracking/vitest.config.ts:L21-L55): `nx test-vitest tracking`, `nx test-vitest smartedit`, `nx test-vitest qualtrics` (vedi 44 — Strategia di test e tooling).

---

11. Domande di autoverifica

1. **Perché importare `BaseTmsModule` senza `forRoot()` non attiva il tracking?** Perché `@NgModule({})` è vuoto: è `forRoot()` a registrare l'`APP_INITIALIZER` con `tmsFactory` che chiama `TmsService.collect()` (feature-libs/tracking/tms/core/base-tms.module.ts:L18-L33).
2. **Ho importato `GtmModule` e messo `gtmId`, ma `window.dataLayer` resta con il solo `gtm.js`. Perché?** Probabilmente manca `events`: il collector è inizializzato comunque (snippet GTM eseguito), ma senza eventi non riceve push (feature-libs/tracking/tms/core/services/tms.service.ts:L68-L75). `GtmModule` non definisce eventi di default (feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L20-L26).
3. **Qual è la differenza osservabile fra il data layer GTM e quello AEP dopo tre eventi?** GTM: array con tre elementi aggiunti con `push` (feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L60-

L61). AEP: un solo oggetto, ricreato a ogni evento con merge shallow, in cui le proprietà omonime sono sovrascritte dall'ultimo evento (feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L39-L43).

4. **Perché un collector custom deve essere disponibile nel root injector?** Perché `TmsService` (`providedIn: 'root'`) lo risolve con il proprio `Injector` (feature-libs/tracking/tms/core/services/tms.service.ts:L27-L41, feature-libs/tracking/tms/core/services/tms.service.ts:L71-L73).
5. **Le richieste OCC fatte durante l'SSR portano l'header** `occ-personalization-id`**?** No: l'interceptor si abilita solo nel browser (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L34-L37), quindi in SSR `enabled` resta `false` e la richiesta passa invariata (feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L62-L64).
6. **Da dove legge Spartacus segmenti e azioni di personalizzazione, e in che formato?** Dalla pagina CMS corrente, slot `PlaceholderContentSlot`, componente `PersonalizationScriptComponent`, `properties.script.data`: JSON in base64 con valori di actions e segments a loro volta in base64 (feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L37-L69, default a feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L16-L19).
7. **Come fa Spartacus a sapere di essere dentro SmartEdit dopo un redirect full-page che ha perso i parametri dell'URL?** Il ticket viene salvato in `sessionStorage` alla prima pagina (`cx-preview` + `cmsTicketId`) e, se l'URL non porta più il contesto, è ripristinato da lì; la sua presenza basta (feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L74-L96).
8. **Come arrivano i decoratori di** `SmartEditCoreModule`**, fornito in un modulo lazy, a** `DynamicAttributeService` **che è un servizio root?** Tramite `UnifiedInjector.getMulti`, che osserva il root injector più gli injector dei moduli lazy caricati e accumula le istanze multi (core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35, core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93, core-libs/core/src/cms/services/dynamic-attribute.service.ts:L24-L32).
9. **Una proprietà** `catalogVersionUuid` **nel gruppo** `smartedit` **di un componente: quale attributo HTML produce?** `data-smartedit-catalog-version-uuid`: prefisso `data-<gruppo>-` più nome spezzato sulle maiuscole, unito con `-` e messo in minuscolo (feature-libs/smartedit/core/services/smart-edit.service.ts:L166-L186).
10. **Perché** `renderComponent` **usa** `zone.run`**, e cosa fa se viene chiamato senza** `parentId`**?** Perché è invocato da SmartEdit tramite `window.smartedit`, fuori dalla zona Angular; senza `parentId` tratta l'oggetto come slot: `clearComponentState()` e ricarica della pagina per id o dell'ultima pagina (feature-libs/smartedit/core/services/smart-edit.service.ts:L126-L149).
11. **Cosa c'è di sbagliato in** `this.qsiApi.API.load().done(this.qsiApi.API.run())`**?** `run()` viene eseguito subito e a `done` viene passato il suo valore di ritorno, non una callback: la valutazione non attende il completamento di `load()` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L138-L138).
12. **Come si fa attendere a Qualtrics che certi dati applicativi siano pronti?** Estendendo `QualtricsLoaderService` e sovrascrivendo l'hook protected `isDataLoaded()` (default `of(true)`), che `qsi$` attraversa con `switchMap` prima di leggere `window.QSI` (feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L66-L71, feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L141-L149).

TAVOLA 33.1 **4.1 TMS: bootstrap e flusso di un evento**

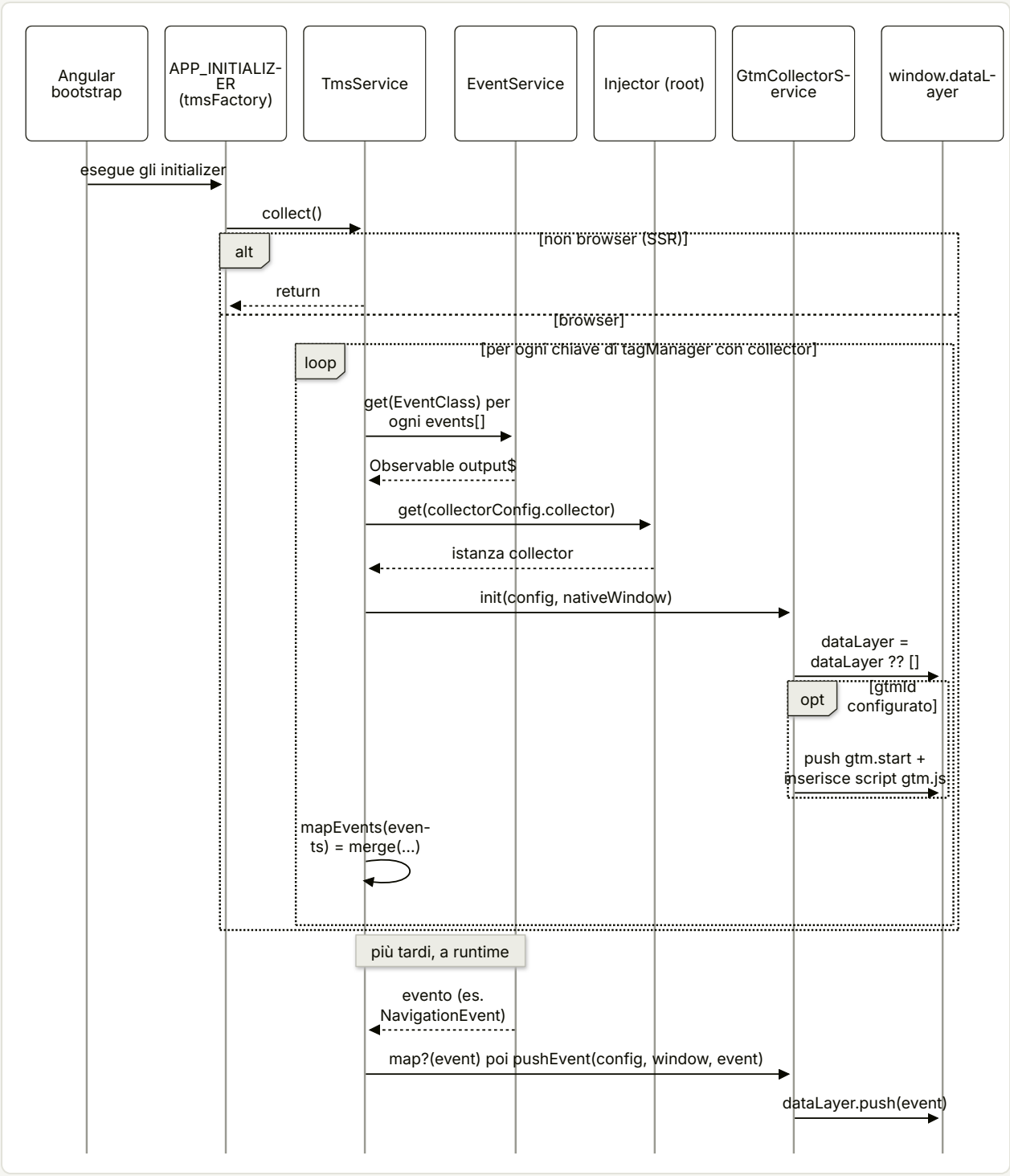
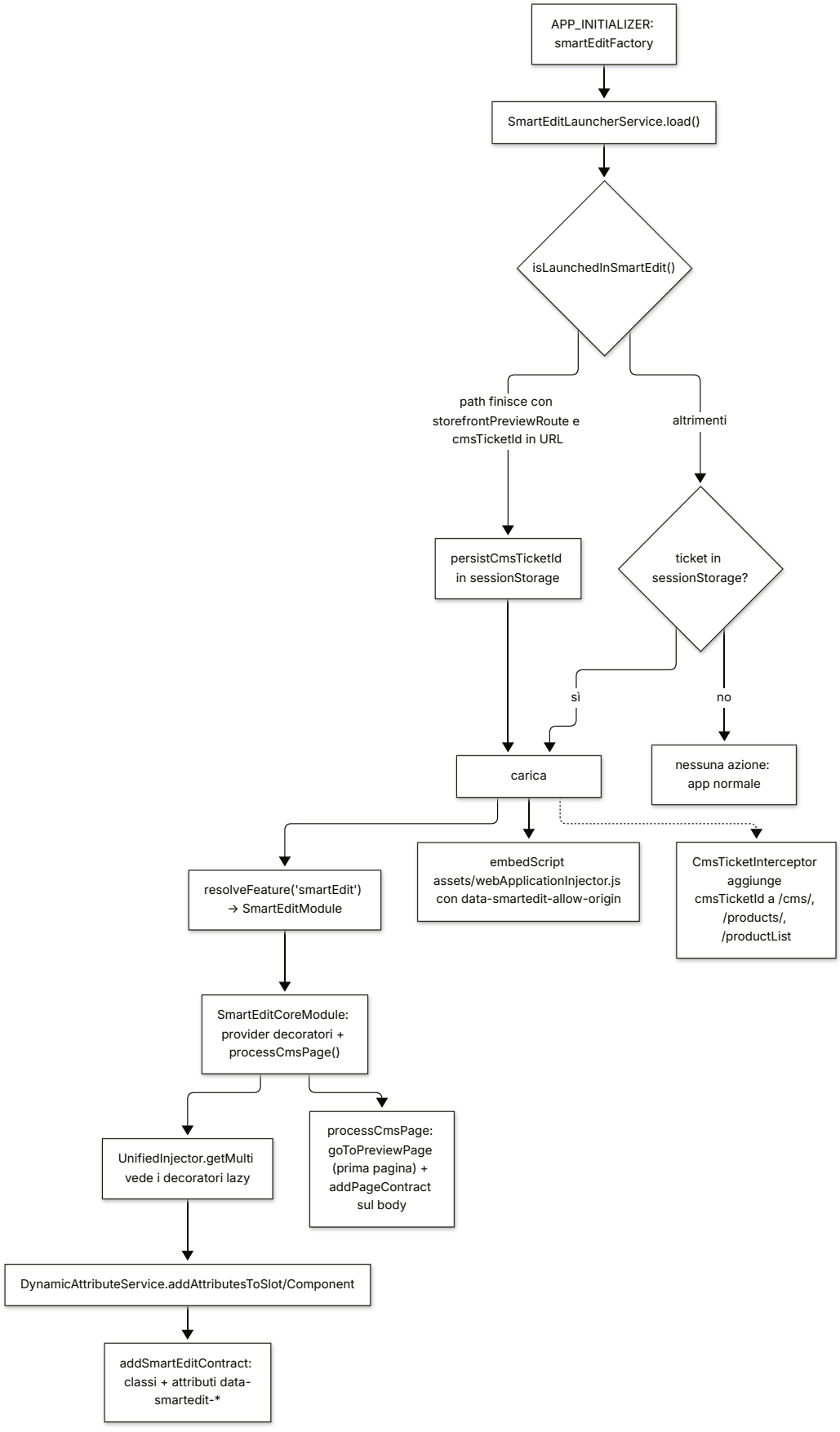


TAVOLA 33.2 **4.3 SmartEdit: avvio, lazy load e decorazione**



34

Integration libs: CDC, CDP, CDS, segment-refs

Integrazioni con Customer Data Cloud, CDP, Context-Driven Services, segmenti.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Le librerie di `integration-libs/` collegano lo storefront a prodotti SAP che **non** fanno parte del backend Commerce standard e che richiedono addon o servizi cloud dedicati. Questo capitolo copre quattro di esse:

LIBRERIA	SISTEMA ESTERNO	COSA FA NELLO STOREFRONT	MECCANISMO DOMINANTE
`@spartacus/cdc`	SAP Customer Data Cloud (ex Gigya)	Sostituisce login, registrazione, logout, gestione profilo, consensi e (in B2B) amministrazione utenti/unità con le API del Gigya JS SDK; poi ottiene un token OAuth Commerce con `grant_type=custom`	override via DI di guard, adapter, component service; `APP_INITIALIZER`; feature lazy con facade
`@spartacus/cdp`	SAP Customer Data Platform	Rende read-only il customer ticketing (niente chiusura, riapertura, invio messaggi)	override via DI di 3 component service
`@spartacus/cds`	SAP Context-Driven Services (Emarsys/ProfileTag)	Carousel di merchandising da strategie CDS, tracciamento comportamentale con lo script ProfileTag, recent/trending searches	adapter/connector, componenti CMS, `HTTP_INTERCEPTORS`, `APP_INITIALIZER`, outlet
`@spartacus/segment-refs`	segmenti esterni (riferimenti a segmenti per la personalizzazione OCC)	Inoltra all'OCC un header con il riferimento di segmento letto da query param/localStorage	un solo `HTTP_INTERCEPTORS`

Il filo conduttore — e la ragione per cui conviene leggere queste librerie anche se non si usano i prodotti — è che **quasi tutta l'integrazione è fatta sostituendo servizi di core e delle feature lib via Dependency Injection**, senza modificare una riga delle librerie originali. In CDC si trovano tutti gli stili di override che Spartacus offre:

1. **override di un token nel root injector** con `useExisting`/`useClass` (guard di login/logout, `UserConsentAdapter`): integration-libs/cdc/root/cdc-root.module.ts:L53-L69, integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L16-L31;
2. **override "scoped" nell'injector di un singolo componente CMS** tramite `cmsComponents.<Tipo>.providers` (login form, registrazione, profilo): integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L44-L68, meccanismo in core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50;
3. **override nell'injector di una feature lazy** importando il modulo di estensione *dopo* quello originale nel wrapper lazy (`CdcAdministrationModule`, `CdpCustomerTicketingModule`): projects/storefrontapp/src/app/spartacus/features/organization/administration-wrapper.module.ts:L12-L20, projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts:L13-L22;
4. **override di configurazione** (route semantiche, `authentication.customLoginPage`): integration-libs/cdc/root/config/default-cdc-routing-config.ts:L9-L35, integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L44-L48;
5. **estensione per sottoclasse** di classi core con metodi `protected` sovrascritti (`CdcLoginGuard extends LoginGuard`, `CdcUserConsentAdapter extends OccUserConsentAdapter`): integration-libs/cdc/root/guards/cdc-login.guard.ts:L12-L32, integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L35-L65.

CDS, al contrario, è soprattutto **additivo**: aggiunge interceptor, initializer, componenti CMS e outlet (integration-libs/cds/src/profiletag/profile-tag.module.ts:L16-L32, integration-libs/cds/src/cds.

module.ts:L26-L47), e sostituisce un solo token astratto che è suo (MerchandisingStrategyAdapter, integration-libs/cds/src/merchandising/merchandising.module.ts:L12-L21). CDP e segment-refs sono librerie minime, utili come esempi "da una pagina" di override (integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts:L17-L35) e di interceptor (integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L69-L86).

Prerequisiti utili: 01 — Dependency Injection, 12 — Feature module e lazy loading, 11 — Configurazione, 20 — Facade, command & query, eventi, 22 — Autenticazione, 16 — CMS rendering.

---

2. Mappa dei file

2.1 integration-libs/cdc

Entry point secondari: @spartacus/cdc (lazy), /root (eager), /core, /components, /user-account, /user-profile, /organization/administration, /organization/user-registration.

PATH	RUOLO
integration-libs/cdc/public_api.ts	esporta solo CdcModule (L7-L7)
integration-libs/cdc/cdc.module.ts	CdcModule = CdcComponentsModule + CdcCoreModule, target del lazy load di CDC_FEATURE
integration-libs/cdc/root/public_api.ts	entry @spartacus/cdc/root
integration-libs/cdc/root/cdc-root.module.ts	CdcRootModule eager: override dei 3 guard, APP_INITIALIZER Gigya, featureModules, routing
integration-libs/cdc/root/feature-name.ts	CDC_FEATURE = 'cdc', CDC_CORE_FEATURE = 'cdcCore'
integration-libs/cdc/root/config/cdc-config.ts	CdcConfig (array cdc[] per base site)
integration-libs/cdc/root/config/default-cdc-routing-config.ts	route cdcLogin, cdcCheckoutLogin, cdcOrgRegistration, ridefinizione login/loginForm
integration-libs/cdc/root/facade/cdc-auth.facade.ts	CdcAuthFacade (facade lazy su cdcCore)
integration-libs/cdc/root/events/cdc-event.ts	CdcLoadUserTokenFailEvent, CdcReConsentEvent
integration-libs/cdc/root/guards/cdc-login.guard.ts	CdcLoginGuard extends LoginGuard
integration-libs/cdc/root/guards/cdc-logout.guard.ts	CdcLogoutGuard extends LogoutGuard
integration-libs/cdc/root/guards/cdc-login-as-guest.guard.ts	CdcLoginAsGuestGuard extends LoginAsGuestGuard
integration-libs/cdc/root/service/cdc-js.service.ts	CdcJsService: carica lo script Gigya e avvolge le sue API callback in Observable
integration-libs/cdc/root/consent-management/cdc-consent.module.ts	override UserConsentAdapter, ConsentManagementComponentService, serializer
integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts	CdcUserConsentAdapter extends OccUserConsentAdapter
integration-libs/cdc/root/consent-management/converters/cdc-preference.serializer.ts	CdcPreferenceSerializer (consensi → preferences annidate)
integration-libs/cdc/root/consent-management/converters/converter.ts	token multi CDC_PREFERENCE_SERIALIZER

PATH	RUOLO
integration-libs/cdc/root/consent-management/model/cdc-consent-management.model.ts	augmentation di `LAUNCH_CALLER` e `UserSignUp`
integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts	persistenza `cdc-consents-list`
integration-libs/cdc/root/consent-management/services/cdc-consent-management-component.service.ts	consensi obbligatori CDC + core
integration-libs/cdc/root/consent-management/services/cdc-user-consent.service.ts	`updateCdcUserPreferences`
integration-libs/cdc/core/public_api.ts	entry `@spartacus/cdc/core` (store non esportato)
integration-libs/cdc/core/cdc-core.module.ts	`CdcCoreModule`: auth + eventi + store + `facadeProviders`
integration-libs/cdc/core/auth/facade/facade-providers.ts	`CdcAuthFacade → CdcAuthService`
integration-libs/cdc/core/auth/facade/cdc-auth.service.ts	`CdcAuthService`: dispatch, check ASM, scrittura token
integration-libs/cdc/core/auth/services/user-authentication/cdc-user-authentication-token.service.ts	POST token endpoint `grant_type=custom`
integration-libs/cdc/core/store/actions/cdc-user-token.action.ts	`LOAD_CDC_USER_TOKEN(_FAIL)`
integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts	effect token
integration-libs/cdc/core/store/effects/cdc-user-addresses.effect.ts	sincronizzazione indirizzo di default verso CDC
integration-libs/cdc/core/events/cdc-event.builder.ts	azione `LOAD_CDC_USER_TOKEN_FAIL` → evento
integration-libs/cdc/core/models/cms.model.ts	`GigyaRaasComponentData`
integration-libs/cdc/components/gigya-raas/gigya-raas.module.ts	mapping CMS `GigyaRaasComponent` + layout `GigyaLoginPageTemplate`
integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts	componente che mostra gli screen-set Gigya
integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts	guard CMS (auth/not-auth per componente)
integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts	override scoped del login form + `customLoginPage: undefined`
integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts	login via `accounts.login`
integration-libs/cdc/user-account/login-form/reconsent/*	dialog di reconsent (`CDC_RECONSENT`)
integration-libs/cdc/user-profile/cdc-user-profile.module.ts	aggregatore degli override profilo
integration-libs/cdc/user-profile/register/cdc-register.module.ts	override scoped `RegisterComponentService`
integration-libs/cdc/user-profile/register/cdc-register-component.service.ts	registrazione via CDC + consensi aggiuntivi

PATH	RUOLO
integration-libs/cdc/user-profile/{forgot-password,update-email,update-password,update-profile}/*	override scoped dei rispettivi component service
integration-libs/cdc/organization/administration/cdc-administration.module.ts	override B2BUserService, OrgUnitService, UserListService
integration-libs/cdc/organization/administration/cdc-user-list.service.ts	apre il Delegated Admin CDC
integration-libs/cdc/organization/user-registration/cdc-b2b-register.module.ts	override UserRegistrationFormService
integration-libs/cdc/assets/translations/translations.ts	chunk i18n cdc: ['reconsent','cdcProfile']

2.2 integration-libs/cds

PATH	RUOLO
integration-libs/cds/public_api.ts	entry @spartacus/cds (no recent-searches, no services, no utils)
integration-libs/cds/src/cds.module.ts	CdsModule eager + forRoot(config?)
integration-libs/cds/src/config/cds-config.ts	CdsConfig
integration-libs/cds/src/config/default-cds-config.ts	default (tenant: '', consentTemplateId: 'PROFILE', soglia 80)
integration-libs/cds/src/config/cds-config-validator.ts	cdsConfigValidator
integration-libs/cds/src/config/profile-tag.config.ts	ProfileTagConfig
integration-libs/cds/src/cds-models/cds-endpoints.model.ts	CdsEndpoints (strategyProducts, searchIntelligence)
integration-libs/cds/src/services/cds-endpoints.service.ts	costruzione URL CDS
integration-libs/cds/src/utils/dynamic-template.ts	sostituzione ${placeholder}
integration-libs/cds/src/merchandising/merchandising.module.ts	MerchandisingStrategyAdapter → CdsMerchandisingStrategyAdapter
integration-libs/cds/src/merchandising/connectors/strategy/merchandising-strategy.adapter.ts	adapter astratto
integration-libs/cds/src/merchandising/connectors/strategy/merchandising-strategy.connector.ts	connector
integration-libs/cds/src/merchandising/adapters/strategy/cds-merchandising-strategy.adapter.ts	GET strategy products
integration-libs/cds/src/merchandising/facade/cds-merchandising-product.service.ts	composizione StrategyRequest
integration-libs/cds/src/merchandising/facade/cds-merchandising-user-context.service.ts	contesto utente (consenso, pagina, facet, ricerca)
integration-libs/cds/src/merchandising/facade/cds-merchandising-site-context.service.ts	contesto sito
integration-libs/cds/src/merchandising/cms-components/merchandising-carousel-cms.module.ts	mapping CMS MerchandisingCarouselComponent

PATH	RUOLO
integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.ts	componente carousel
integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.service.ts	modello, soglia viewport, eventi
integration-libs/cds/src/merchandising/cms-components/directives/attributes/attributes.directive.ts	`cxAttributes`
integration-libs/cds/src/profiletag/profile-tag.module.ts	interceptor, adapter notifica login, `provideLoginEventsTracking()`
integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts	header `X-Consent-Reference`
integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts	header `X-Profile-Tag-Debug`
integration-libs/cds/src/profiletag/cms-components/profile-tag.component.ts	componente CMS senza UI
integration-libs/cds/src/profiletag/services/profiletag-event.service.ts	script ProfileTag, `Y_TRACKING`, consent reference
integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts	eventi Spartacus → push event
integration-libs/cds/src/profiletag/services/profile-tag-lifecycle.service.ts	consenso e login
integration-libs/cds/src/profiletag/services/profile-tag.injector.service.ts	`track()`
integration-libs/cds/src/profiletag/providers/login-events.provider.ts	`LOGIN_EVENTS` + `provideAppInitializer`
integration-libs/cds/src/profiletag/adapters/occ-backend-notification-adapter.ts	POST `/users/{id}/loginnotification`
integration-libs/cds/src/profiletag/tracking/tracking.module.ts	`APP_INITIALIZER` del tracking
integration-libs/cds/src/profiletag/tracking/tracking.service.ts	filtro per consenso e inoltro
integration-libs/cds/src/recent-searches/recent-searches.module.ts	outlet recent searches
integration-libs/cds/src/trending-searches/trending-searches.module.ts	outlet trending searches
integration-libs/cds/src/trending-searches/trending-searches.service.ts	polling `searchIntelligence`

2.3 `integration-libs/cdp` e `integration-libs/segment-refs`

PATH	RUOLO
integration-libs/cdp/public_api.ts	entry `@spartacus/cdp` → solo `customer-ticketing`
integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts	override di 3 component service
integration-libs/cdp/customer-ticketing/cdp-customer-ticketing-close-component.service.ts	`enableCloseButton → of(false)`

PATH	RUOLO
integration-libs/cdp/customer-ticketing/cdp-customer-ticketing-reopen-component.service.ts	enableReopenButton → of(false)
integration-libs/cdp/customer-ticketing/cdp-customer-ticketing-messages-component.service.ts	displayAddMessageSection → false
integration-libs/segment-refs/public_api.ts	entry @spartacus/segment-refs
integration-libs/segment-refs/root/segment-refs-root.module.ts	interceptor + default config
integration-libs/segment-refs/root/config/segment-refs-config.ts	SegmentRefsConfig (non esportato)
integration-libs/segment-refs/root/config/default-segment-refs-config.ts	httpHeaderName: 'Segmentrefs'
integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts	OccSegmentRefsInterceptor
integration-libs/segment-refs/root/feature-name.ts	SEGMENT_REFS_FEATURE (inutilizzata)

2.4 Wiring (schematics e demo app)

PATH	RUOLO
core-libs/schematics/src/shared/lib-configs/integration-libs/cdc-schematics-config.ts	root module eager, chunk lazy, importAfter per user/organization
core-libs/schematics/src/shared/lib-configs/integration-libs/cds-schematics-config.ts	CdsModule.forRoot(), endpoint generati, lazy: false
core-libs/schematics/src/shared/lib-configs/integration-libs/cdp-schematics-config.ts	importAfter su CustomerTicketingModule
core-libs/schematics/src/shared/lib-configs/integration-libs/segment-refs-schematics-config.ts	dipendenza da personalization
projects/storefrontapp/src/app/spartacus/features/cdc/cdc-feature.module.ts	CdcRootModule + lazy CdcModule
projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts	factory config CDS per base site
projects/storefrontapp/src/app/spartacus/features/user/user-account-wrapper.module.ts	CDCUserAccountModule dopo UserAccountModule
projects/storefrontapp/src/app/spartacus/features/organization/administration-wrapper.module.ts	CdcAdministrationModule dopo AdministrationModule
projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts	CdpCustomerTicketingModule nel wrapper lazy

File aggiuntivi

PATH	RUOLO
integration-libs/cdc/schematics/add-cdc/index.ts	rule ng add di @spartacus/cdc (addCdcFeature)
integration-libs/cdc/schematics/add-cdc/index_spec.ts	test Jest dello schematic CDC: B2C e B2B, validazione di javascriptUrl, lazy/eager
integration-libs/cdc/vitest.config.ts	configurazione Vitest di cdc (restoreMocks: true, coverage solo lcov)
integration-libs/cdp/schematics/add-cdp/index.ts	rule ng add di @spartacus/cdp (addCdpFeature)

PATH	RUOLO
integration-libs/cdp/schematics/add-cdp/index_spec.ts	test Jest dello schematic CDP (richiede prima customer ticketing)
integration-libs/cdp/vitest.config.ts	configurazione Vitest di cdp
integration-libs/cds/vitest.config.ts	configurazione Vitest di cds (soglie 80%)
integration-libs/segment-refs/schematics/add-segment-refs/index.ts	rule `ng add` di `@spartacus/segment-refs` (`addSegmentRefsFeature`)
integration-libs/segment-refs/schematics/add-segment-refs/index_spec.ts	test Jest dello schematic segment-refs (richiede prima personalization)
integration-libs/segment-refs/vitest.config.ts	configurazione Vitest di segment-refs (soglie 90/90/75/90)

3. Come funziona passo per passo

3.1 CDC: struttura root/lazy

CDC segue la convenzione root/lazy descritta in 12 — Feature module e lazy loading, con una particolarità: il root module non è affatto "leggero".

1. **Nomi delle feature.** `CDC_FEATURE = 'cdc'` e `CDC_CORE_FEATURE = 'cdcCore'` — integration-libs/cdc/root/feature-name.ts:L7-L8.
2. **Config CMS di default.** `defaultCdcComponentsConfig` dichiara che la feature `cdc` possiede il componente CMS `GigyaRaasComponent` e che `cdcCore` è un **alias stringa** di `cdc` ("by default core is bundled together with components") — integration-libs/cdc/root/cdc-root.module.ts:L40-L51. L'alias è risolto da `FeatureModulesService.resolveFeatureAlias`, che segue la catena finché il valore è una stringa — core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100.
3. **Modulo lazy.** `CdcModule` importa `CdcComponentsModule` e `CdcCoreModule` — integration-libs/cdc/cdc.module.ts:L7-L14. Nella demo app `CdcFeatureModule` importa `CdcRootModule` e registra `module: () => import('@spartacus/cdc').then((m) => m.CdcModule)` — projects/storefrontapp/src/app/spartacus/features/cdc/cdc-feature.module.ts:L17-L40.
4. **Root module eager.** `CdcRootModule` importa `CdcConsentManagementModule` e fornisce: config CMS, tre override di guard con `useExisting`, un `APP_INITIALIZER` e la routing config — integration-libs/cdc/root/cdc-root.module.ts:L53-L69.
5. **Facade lazy.** `CdcAuthFacade` è una classe astratta `providedIn: 'root'` con `useFactory: () => facadeFactory({ facade, feature: CDC_CORE_FEATURE, methods: ['loginWithCustomCdcFlow','loginWithToken'], async: true })` — integration-libs/cdc/root/facade/cdc-auth.facade.ts:L11-L33. Con `async: true` il resolver aggiunge `delay(0)` e ogni chiamata di metodo è resa *eager* con `connectable(...).connect()` su `ReplaySubject`: il metodo reale viene invocato anche se il chiamante non si sottoscrive — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L53-L56, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L82. Quando il modulo lazy è caricato, `facadeProviders` lega la facade a `CdcAuthService` con `useExisting` — integration-libs/cdc/core/auth/facade/facade-providers.ts:L11-L17.
6. **Core lazy.** `CdcCoreModule` importa `CdcAuthModule`, `CdcEventModule`, `CdcStoreModule` — integration-libs/cdc/core/cdc-core.module.ts:L13-L17; lo store registra `EffectsModule.forFeature(effects)` — integration-libs/cdc/core/store/cdc-store.module.ts:L14-L18 — con `CdcUserTokenEffects` e `CdcUserAddressesEffects` — integration-libs/cdc/core/store/effects/index.ts:L10-L10.

Conseguenza: tutto ciò che deve esistere **prima** del primo login (guard, script Gigya, consensi, `CdcJsService`) sta nel root; la parte NgRx che produce il token Commerce è lazy e viene scaricata alla prima chiamata della facade.

3.2 CDC: bootstrap dello script Gigya

1. `cdcJsFactory` restituisce una funzione che ritorna `lastValueFrom(configInit.getStable('context', 'cdc').pipe(tap(() => cdcJsService.initialize())))` — integration-libs/cdc/root/cdc-root.module.ts:L26-L38. Il token `APP_INITIALIZER` è deprecato da Angular v19 a favore di `provideAppInitializer` — angular:packages/core/src/application/application_init.ts:L38-L38.
2. `getStable(...scopes)` emette subito se la config è già stabile, altrimenti attende che nessuno scope in corso si sovrapponga a quelli richiesti — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70. Tra gli initializer di core con scope dichiarati ci sono quelli di site context (`['context']`), i18n (`['i18n.fallbackLang']`), auth (`['authentication.OAuthLibConfig.redirectUri', 'authentication.client_id']`) e secure portal (`['routing']`); nessuno si sovrappone a `'cdc'` — core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L26-L26, core-libs/core/src/i18n/config/i18n-config-initializer.ts:L16-L16, core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L20-L23, core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L18-L18; `scopesOverlap` usa il confronto per prefisso puntato — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L117-L122. Quindi lo scope `'cdc'` serve solo se l'applicazione registra un proprio initializer con quello scope (nessuno nel repo: vedi §9).
3. `loadCdcJavascript` agisce **solo in browser** (`isPlatformBrowser`), prende il primo valore di base site e lingua attivi (`take(1)`) — integration-libs/cdc/root/service/cdc-js.service.ts:L94-L103.
4. La config per base site si sceglie filtrando `cdcConfig.cdc` per `baseSite` e prendendo il primo elemento; senza corrispondenza ritorna `''` e nessuno script viene caricato — integration-libs/cdc/root/service/cdc-js.service.ts:L141-L149.
5. L'URL è `javascriptUrl&lang=<lingua>`, iniettato con `ScriptLoader.embedScript`. Al `load` registra gli handler Gigya, scarica i consensi del sito persistendoli in local storage ed emette `loaded$ = true`; all'errore `errorLoading$ = true` — integration-libs/cdc/root/service/cdc-js.service.ts:L104-L122. Poi imposta `window.__gigyaConf = { include: 'id_token, missing-required-fields, preferences' }` — integration-libs/cdc/root/service/cdc-js.service.ts:L123-L129.
6. `ScriptLoader.embedScript` in SSR non accetta callback, e in browser non reinserisce uno script già presente — core-libs/core/src/util/script-loader.service.ts:L55-L61, core-libs/core/src/util/script-loader.service.ts:L84-L89.
7. `addCdcEventHandlers` legge `window.gigya` e registra `onLogin`, eseguito dentro `zone.run` (le callback del SDK arrivano fuori dalla zona Angular) — integration-libs/cdc/root/service/cdc-js.service.ts:L165-L173.

`CdcJsService` conserva `loaded$` ed `errorLoading$` come `ReplaySubject(1)` — integration-libs/cdc/root/service/cdc-js.service.ts:L43-L52 — esposti come `didLoad()` / `didScriptFailToLoad()` e consumati da componenti e component service (es. integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L27-L59).

3.3 CDC: il ponte `invokeAPI` verso il Gigya SDK

Tutte le operazioni "without screen-set" passano per un unico metodo:

1. `getSdkFunctionFromName` scende in `gigyaSDK` lungo i segmenti del nome puntato (`accounts.setAccountInfo`) con `hasOwnProperty`; un segmento mancante viene saltato — integration-libs/cdc/

root/service/cdc-js.service.ts:L713-L726.

2. `invokeAPI` crea un Observable **cold** (`new Observable`), chiama `actualAPI({...payload, callback})` e nella callback, dentro `zone.run`, emette `next + complete` se `status === 'OK'`, altrimenti `error(response)` — integration-libs/cdc/root/service/cdc-js.service.ts:L734-L755.

Su questo ponte sono costruiti:

METODO DI CDCJSSERVICE	API GIGYA	RIFERIMENTO
`registerUserWithoutScreenSet`	`accounts.initRegistration` → `accounts.register` (`finalizeRegistration: true`)	integration-libs/cdc/root/service/cdc-js.service.ts:L199-L243
`loginUserWithoutScreenSet`	`accounts.login` (`ignoreInterruptions` solo se canale `B2C`, `sessionExpiry`)	integration-libs/cdc/root/service/cdc-js.service.ts:L252-L292
`registerOrganisationWithoutScreenSet`	`accounts.b2b.registerOrganization`	integration-libs/cdc/root/service/cdc-js.service.ts:L299-L344
`getOrganizationContext`	`accounts.b2b.getOrganizationContext`	integration-libs/cdc/root/service/cdc-js.service.ts:L350-L352
`openDelegatedAdminLogin` (chiamata diretta al SDK dentro `zone.run`, **non** via `invokeAPI`)	`accounts.b2b.openDelegatedAdminLogin`	integration-libs/cdc/root/service/cdc-js.service.ts:L359-L365
`resetPasswordWithoutScreenSet`	`accounts.resetPassword`	integration-libs/cdc/root/service/cdc-js.service.ts:L460-L494
`updateProfileWithoutScreenSet`	`accounts.setAccountInfo` (+ update Commerce)	integration-libs/cdc/root/service/cdc-js.service.ts:L502-L528
`updateUserPasswordWithoutScreenSet`	`accounts.setAccountInfo` (`password`, `newPassword`)	integration-libs/cdc/root/service/cdc-js.service.ts:L536-L557
`updateUserEmailWithoutScreenSet`	`accounts.login` con `skipOccAuth` + `setAccountInfo`	integration-libs/cdc/root/service/cdc-js.service.ts:L635-L650
`updateAddressWithoutScreenSet`	`accounts.setAccountInfo` (`profile.address`, …)	integration-libs/cdc/root/service/cdc-js.service.ts:L687-L706
`getSiteConsentDetails`	`accounts.getSiteConsentDetails` (apikey dalla query di `javascriptUrl`)	integration-libs/cdc/root/service/cdc-js.service.ts:L762-L782
`setUserConsentPreferences`	`accounts.setAccountInfo` (`preferences`, `regToken`)	integration-libs/cdc/root/service/cdc-js.service.ts:L792-L812

Il caso più curioso è l'update email: "CDC doesn't have any specific api, for verifying a password", quindi il codice chiama `loginUserWithoutScreenSet(email, password, { skipOccAuth: true })` come verifica e poi `setAccountInfo` — integration-libs/cdc/root/service/cdc-js.service.ts:L635-L650. Il flag `skipOccAuth` arriva nel `context` della risposta `onLogin` ed evita il ri-login su Commerce — integration-libs/cdc/root/service/cdc-js.service.ts:L181-L192.

3.4 CDC: dal login Gigya al token OAuth Commerce

1. **Origine del login.** O lo screen-set Gigya (`GigyaRaasComponent.showScreenSet` → `gigya.accounts.showScreenSet`, integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L80-L105) o il form Spartacus (`CdcLoginFormComponentService.login` → `loginUserWithoutScreenSet`, integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L34-L66).
2. `onLogin` **di Gigya** → `onLoginEventHandler` chiama `cdcAuth.loginWithCustomCdcFlow(UID, UIDSignature, signatureTimestamp, id_token || '', baseSite)` salvo `skipOccAuth` — integration-libs/cdc/root/service/cdc-js.service.ts:L181-L192.
3. **Facade → lazy load.** La facade risolve `cdcCore` → alias `cdc` → carica `CdcModule` — integration-libs/cdc/root/facade/cdc-auth.facade.ts:L11-L33, core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100.
4. **Dispatch.** `CdcAuthService.loginWithCustomCdcFlow` dispatcha `LoadCdcUserToken` — integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L47-L63; azione `'[Auth] Load CDC User Token'` — integration-libs/cdc/core/store/actions/cdc-user-token.action.ts:L10-L41.
5. **Effect.** `CdcUserTokenEffects.loadCdcUserToken$` chiama `loadTokenUsingCustomFlow` — integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts:L25-L59 — che fa POST al token endpoint OAuth con `grant_type=custom`, `client_id`, `client_secret` e i campi CDC URL-encoded — integration-libs/cdc/core/auth/services/user-authentication/cdc-user-authentication-token.service.ts:L43-L60.
6. **Successo.** L'effect chiama `cdcAuthService.loginWithToken(token)` e ritorna `EMPTY` — integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts:L40-L43. `loginWithToken` verifica prima che non ci sia un agente ASM loggato (se lo storage è un `AsmAuthStorageService` con token di tipo `CSAgent`, mostra `asm.auth.agentLoggedInError`) — integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L68-L104; poi `setTokenData`, `setUserId(OCC_USER_ID_CURRENT)`, `AuthActions.Login()`, rimozione errori globali e `authRedirectService.redirect()` — integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L106-L120.
7. **Formato token.** `setTokenData` scrive nello `AuthStorageService` le stesse chiavi che userebbe la libreria OAuth: `access_token`, `granted_scopes`, `access_token_stored_at`, `expires_at`, `refresh_token` — integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L123-L145. Da qui in poi il resto dello storefront (interceptor, refresh) vede un utente normalmente autenticato: vedi 22 — Autenticazione.
8. **Errore.** Messaggio `httpHandlers.badGateway` e azione `LoadCdcUserTokenFail` — integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts:L44-L55 — che `CdcEventBuilder` trasforma in `CdcLoadUserTokenFailEvent` via `StateEventService.register` — integration-libs/cdc/core/events/cdc-event.builder.ts:L33-L38.

DA VERIFICARE: quale estensione lato Commerce accetti `grant_type=custom` con questi parametri; il codice Spartacus non lo dichiara (dossier Q1).

3.5 CDC: override dei guard di login/logout

GUARD CORE	OVERRIDE CDC	COSA CAMBIA	RIFERIMENTO
`LoginGuard` (storefront)	`CdcLoginGuard`	`shouldRenderCMSPage()` → sempre `of(true)`: la pagina CMS di login è sempre mostrata e non si fa mai `loginWithRedirect()` verso il server OAuth	integration-libs/cdc/root/guards/cdc-login.guard.ts:L12-L32; base core-libs/storefront/cms-components/user/login-route/login.guard.ts:L34-L65

GUARD CORE	OVERRIDE CDC	COSA CAMBIA	RIFERIMENTO
`LogoutGuard` (storefront)	`CdcLogoutGuard`	`logout()` esegue in parallelo `super.logout()` (`auth.coreLogout()`) e `gigya.accounts.logout()`	integration-libs/cdc/root/guards/cdc-logout.guard.ts:L43-L55; base core-libs/storefront/cms-components/user/logout/logout.guard.ts:L87-L89
`LoginAsGuestGuard` (user/account/components)	`CdcLoginAsGuestGuard`	`getPath()` usa la route `login` invece di `loginForm`	integration-libs/cdc/root/guards/cdc-login-as-guest.guard.ts:L10-L17; base feature-libs/user/account/components/guards/login-as-guest.guard.ts:L26-L28

Gli override funzionano perché le route non referenziano la classe concreta ma il token: `canActivate: [LoginGuard]` — core-libs/storefront/cms-components/user/login-route/login-route.module.ts:L29-L29; `canActivate: [LogoutGuard, CmsPageGuard]` — core-libs/storefront/cms-components/user/logout/logout.module.ts:L25-L25; `guards: [NotAuthGuard, LoginAsGuestGuard]` nel componente login-register — feature-libs/user/account/components/login-register/login-register.module.ts:L37-L37. Il cambio di path in `CdcLoginAsGuestGuard` è coerente con la routing config CDC, che svuota `loginForm.paths` "to avoid same path in different routes" — integration-libs/cdc/root/config/default-cdc-routing-config.ts:L9-L35.

3.6 CDC: override "scoped" dei component service

I moduli `CDCLoginFormModule`, `CDCRegisterModule`, `CDCForgotPasswordModule`, `CDCUpdateProfileModule`, `CDCUpdatePasswordModule`, `CDCUpdateEmailModule` non toccano il root injector: aggiungono `providers` alla mappatura CMS del componente.

1. `CDCLoginFormModule` imposta `authentication.customLoginPage: undefined` e ridefinisce `ReturningCustomerLoginComponent` con `guards: [NotAuthGuard]` e il provider `LoginFormComponentService → CdcLoginFormComponentService` con `deps` espliciti — integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L44-L68.
2. `CmsInjectorService.getInjector` crea per ogni istanza di componente un `Injector.create` figlio con quei provider (parent = injector del modulo/feature) — core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50. L'override vale **solo** nell'albero di quel componente.
3. Il default di user/account per lo stesso componente ha `guards: [NotAuthGuard, CustomLoginGuard]` — feature-libs/user/account/components/login-form/login-form.module.ts:L52-L59. Siccome `deepMerge` sostituisce gli array (ramo `else`) — core-libs/core/src/config/utils/deep-merge.ts:L36-L45 — l'override CDC rimuove `CustomLoginGuard`. Allo stesso modo `customLoginPage: undefined` sovrascrive il default `{ csrfEndpoint: '/csrf', loginFormEndpoint: '/login' }` e fa ritornare `false` a `customLoginEnabled()` — core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L65-L68, core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L105-L107.
4. `CdcLoginFormComponentService.login()` attende `didLoad()`, poi `loginUserWithoutScreenSet(userId.toLowerCase(), password)`; se lo script non c'è mostra `errorHandlers.scriptFailedToLoad` — integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L34-L66. `initCustomLogin()` è sovrascritto vuoto "so csrf token is not initialized in case of CDC" — integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L68-L70; nel servizio base è invocato dal costruttore con `authorizationCodeFlowByDefault` — feature-libs/user/account/components/login-form/login-form-component.service.ts:L89-L91.
5. `CDCRegisterModule` aggiunge a `RegisterCustomerComponent` solo `providers: [RegisterComponentService → CDCRegisterComponentService]`, lasciando `component`/`guards` del default — integration-libs/cdc/user-

profile/register/cdc-register.module.ts:L38-L51. Lo stesso schema per `ForgotPasswordComponent`, `UpdateProfileComponent`, `UpdatePasswordComponent`, `UpdateEmailComponent` — integration-libs/cdc/user-profile/forgot-password/cdc-forgot-password.module.ts:L35-L47, integration-libs/cdc/user-profile/update-profile/cdc-update-profile.module.ts:L39-L52, integration-libs/cdc/user-profile/update-password/cdc-update-password.module.ts:L37-L50, integration-libs/cdc/user-profile/update-email/cdc-update-email.module.ts:L37-L50.

6. `CDCRegisterComponentService.register` attende il load di CDC, esegue un `Command` che chiama `registerUserWithoutScreenSet`, attende `merge(loadUserTokenFailed$, isLoggedIn$)` e poi aggiorna il profilo Commerce — integration-libs/cdc/user-profile/register/cdc-register-component.service.ts:L43-L123. Il form di registrazione aggiunge i consensi CDC attivi (con `Validators.requiredTrue` per gli obbligatori) e li serializza in `preferences` — integration-libs/cdc/user-profile/register/cdc-register-component.service.ts:L134-L215. `postRegisterMessage()` è svuotato — integration-libs/cdc/user-profile/register/cdc-register-component.service.ts:L125-L128.

3.7 CDC: consensi

1. `CdcConsentManagementModule` (importato dal root, quindi eager) sostituisce `UserConsentAdapter` con `CdcUserConsentAdapter`, `ConsentManagementComponentService` con `CdcConsentManagementComponentService` e registra `CdcPreferenceSerializer` nel token multi `CDC_PREFERENCE_SERIALIZER` — integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L16-L31. Il default è `OccUserConsentAdapter` in `UserOccModule` — core-libs/core/src/occ/adapters/user/user-occ.module.ts:L39-L39.
2. All'avvio dello script, `getSiteConsentDetails(true)` persiste i consensi del sito: `CdcConsentsLocalStorageService` salva in `StatePersistenceService` (chiave `cdc-consents-list`) solo quelli `isActive === true`, con `required = isMandatory` — integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts:L15-L51.
3. `CdcUserConsentAdapter.giveConsent`: delega a OCC se la lista CDC in storage esiste e non contiene l'id; se la lista non è mai stata persistita, `readFromStorage` ritorna `undefined` e `checkIfConsentExists` lancia `TypeError` (né OCC né CDC vengono chiamati) — integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts:L57-L76, core-libs/core/src/state/utils/browser-storage.ts:L65-L68; altrimenti aggiorna prima le preferenze CDC e chiama l'OCC solo con `errorCode === 0`, altrimenti `EMPTY` — integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L35-L65. `withdrawConsent` usa il terzo parametro `consentId` — integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L67-L89 — che il connector core passa già, anche se `OccUserConsentAdapter.withdrawConsent` ne dichiara solo due — core-libs/core/src/user/connectors/consent/user-consent.connector.ts:L39-L39, core-libs/core/src/occ/adapters/user/occ-user-consent.adapter.ts:L69-L69.
4. `CdcUserConsentService.updateCdcUserPreferences` serializza con `CDC_PREFERENCE_SERIALIZER`, usa uid e lingua attivi, e chiama `setUserConsentPreferences` — integration-libs/cdc/root/consent-management/services/cdc-user-consent.service.ts:L34-L65. Il serializer trasforma `[{id:'terms.marketing', isConsentGranted:true}]` in `{ terms: { marketing: { isConsentGranted: true } } }` con `reduceRight` + `deepMerge` — integration-libs/cdc/root/consent-management/converters/cdc-preference.serializer.ts:L18-L38; `ConverterService.convert` applica in catena tutti i converter del token — core-libs/core/src/util/converter.service.ts:L106-L112, core-libs/core/src/util/converter.service.ts:L130-L137.
5. `CdcConsentManagementComponentService.getRequiredConsents` unisce gli obbligatori core con quelli CDC — integration-libs/cdc/root/consent-management/services/cdc-consent-management-component.service.ts:L20-L52.

Reconsent al login. Se `accounts.login` risponde con errore `206001`, `loginUserWithoutScreenSet` non mostra un errore ma chiama `raiseCdcReconsentEvent` — integration-libs/cdc/root/service/cdc-js.service.ts:L252-L292 — che normalizza gli id (`preferences.X.isConsentGranted` → `X`) e dispatcha `CdcReConsentEvent`

— integration-libs/cdc/root/service/cdc-js.service.ts:L823-L848. CdcReconsentModule istanzia nel costruttore CdcReconsentDialogEventListener — integration-libs/cdc/user-account/login-form/reconsent/cdc-reconsent.module.ts:L21-L41 — che apre il dialog LAUNCH_CALLER.CDC_RECONSENT — integration-libs/cdc/user-account/login-form/reconsent/cdc-reconsent-dialogue-event.listener.ts:L25-L51. Il valore dell'enum è aggiunto per augmentation e assegnazione runtime — integration-libs/cdc/root/consent-management/model/cdc-consent-management.model.ts:L9-L26 — e configurato come DIALOG_TYPE.DIALOG con inlineRoot: true — integration-libs/cdc/user-account/login-form/reconsent/default-cdc-reconsent-layout.config.ts:L10-L18. Alla conferma, savePreferencesAndLogin aggiorna le preferenze con regToken e, se errorCode === 0, ripete il login e chiude il dialog — integration-libs/cdc/user-account/login-form/reconsent/cdc-reconsent-component.service.ts:L32-L75.

3.8 CDC: componente GigyaRaas e guard

1. GigyaRaasModule mappa GigyaRaasComponent con guards: [GigyaRaasGuard] e definisce il layout GigyaLoginPageTemplate (BodyContent, BottomContent) — integration-libs/cdc/components/gigya-raas/gigya-raas.module.ts:L14-L35.
2. Il componente ri-renderizza lo screen-set ad ogni cambio lingua (distinctUntilChanged + renderScreenSet = true) — integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L27-L59. In modalità embed (embed === 'true' e containerID non vuoto) mostra lo screen-set inline, altrimenti un link popup — integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L141-L147, integration-libs/cdc/components/gigya-raas/gigya-raas.component.html:L1-L24.
3. showScreenSet passa sessionExpiration per gli screen-set di login, e per quelli di profile edit aggancia onSubmit (che calcola isPasswordReset) e onAfterSubmit, che passa i parametri a onProfileUpdateEventHandler aggiungendo in coda l'**oggetto** { passwordReset } — integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L80-L105, integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L97-L101. onProfileUpdateEventHandler aggiorna l'utente Commerce e fa logout se currentEmail !== userDetails.uid || isPasswordReset — integration-libs/cdc/root/service/cdc-js.service.ts:L564-L609. Siccome il secondo parametro è tipizzato boolean ma riceve un oggetto (sempre truthy), con un evento a un solo argomento il logout scatta a ogni profile edit, non solo quando cambia l'email o si resetta la password. DA VERIFICARE: il numero di argomenti passati da Gigya a onAfterSubmit dipende dal Gigya SDK, che non è nel repo (con più argomenti il secondo parametro sarebbe comunque un argomento Gigya, non il flag).
4. GigyaRaasGuard legge la prossima page context, trova i componenti GigyaRaasComponent della pagina e applica AuthGuard se showAnonymous === 'false' o NotAuthGuard se showLoggedIn === 'false' — integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts:L38-L78, integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts:L80-L122.

3.9 CDC: B2B (organization)

1. CdcAdministrationModule sostituisce B2BUserService e OrgUnitService sia come provider di modulo (useClass) sia nei provider CMS di ManageUsersListComponent (+ UserListService → CdcUserListService) e ManageUnitsListComponent, **riaccodando** i provider originali — integration-libs/cdc/organization/administration/cdc-administration.module.ts:L23-L62. Il riaccodamento serve perché l'array providers sostituirebbe quello di default (ListService, ItemService) — feature-libs/organization/administration/components/user/user.config.ts:L42-L53.
2. CdcB2BUserService.isUpdatingUserAllowed() e CdcOrgUnitService.isUpdatingUnitAllowed() ritornano false — integration-libs/cdc/organization/administration/cdc-b2b-user.service.ts:L12-L16, integration-libs/cdc/organization/administration/cdc-org-unit.service.ts:L12-L16 — e il flag è letto da UserGuard e UserDetailsComponent — feature-libs/organization/administration/core/guards/user.

guard.ts:L26-L27, feature-libs/organization/administration/components/user/details/user-details.component.ts:L62-L62.

3. CdcUserListService.onCreateButtonClick legge l'organizzazione con getOrganizationContext e apre il Delegated Admin CDC; il pulsante diventa CreateButtonType.BUTTON — integration-libs/cdc/organization/administration/cdc-user-list.service.ts:L39-L75.
4. CDCB2BRegisterModule sostituisce UserRegistrationFormService con CDCB2BRegisterComponentService — integration-libs/cdc/organization/user-registration/cdc-b2b-register.module.ts:L33-L38 — che registra l'organizzazione via CDC, mostra il messaggio, redirige al login e resetta il form — integration-libs/cdc/organization/user-registration/cdc-b2b-register-component.service.ts:L110-L131. Per il dominio B2B vedi 28 — B2B: organization.

3.10 CDC: wiring

Lo schematic installa CdcRootModule eager, usa CDC_FEATURE come chunk lazy e inserisce CDCUserAccountModule/CDCUserProfileModule (più CdcAdministrationModule/CDCB2BRegisterModule in B2B) con importAfter rispetto ai moduli originali — core-libs/schematics/src/shared/lib-configs/integration-libs/cdc-schematics-config.ts:L55-L101, core-libs/schematics/src/shared/lib-configs/integration-libs/cdc-schematics-config.ts:L126-L136; la config generata ha placeholder BASE_SITE_PLACEHOLDER, JS_SDK_URL_PLACEHOLDER e sessionExpiration 3600 — core-libs/schematics/src/shared/lib-configs/integration-libs/cdc-schematics-config.ts:L161-L171. Nella demo app le estensioni sono importate nei wrapper lazy *dopo* il modulo originale, condizionate da environment.cdc — projects/storefrontapp/src/app/spartacus/features/user/user-account-wrapper.module.ts:L13-L22, projects/storefrontapp/src/app/spartacus/features/organization/administration-wrapper.module.ts:L12-L20.

3.11 CDP: customer ticketing in sola lettura

1. Il pacchetto esporta solo customer-ticketing (tre servizi e un modulo; niente config/adapter) — integration-libs/cdp/public_api.ts:L7-L7, integration-libs/cdp/customer-ticketing/public_api.ts:L7-L10.
2. CdpCustomerTicketingModule sostituisce a livello di modulo CustomerTicketingCloseComponentService, CustomerTicketingReopenComponentService, CustomerTicketingMessagesComponentService — integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts:L17-L35.
3. Le varianti ritornano of(false) / false — integration-libs/cdp/customer-ticketing/cdp-customer-ticketing-close-component.service.ts:L12-L16, integration-libs/cdp/customer-ticketing/cdp-customer-ticketing-reopen-component.service.ts:L12-L16, integration-libs/cdp/customer-ticketing/cdp-customer-ticketing-messages-component.service.ts:L12-L16.
4. I servizi base sono @Injectable() forniti nei moduli dei componenti — feature-libs/customer-ticketing/components/details/customer-ticketing-close/customer-ticketing-close.module.ts:L44-L45 — e consumati dai componenti — feature-libs/customer-ticketing/components/details/customer-ticketing-close/customer-ticketing-close.component.ts:L36-L37.
5. Lo schematic importa CdpCustomerTicketingModule dopo CustomerTicketingModule (importAfter) — core-libs/schematics/src/shared/lib-configs/integration-libs/cdp-schematics-config.ts:L28-L40; nella demo app avviene nel wrapper lazy — projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts:L13-L22. Per la feature base vedi 31 — Quote, customer ticketing, subscription billing.

3.12 CDS: modulo, configurazione, endpoint

1. `CdsModule` importa **eager** `ProfileTagModule`, `TrackingModule`, `MerchandisingModule`, `RecentSearchesModule`, `TrendingSearchesModule`; `forRoot(config?)` aggiunge default config factory, la config passata, il validator e `ProfileTagPushEventsService`. Non esiste root module né feature lazy — integration-libs/cds/src/cds.module.ts:L26-L47. Lo schematic importa `CdsModule.forRoot()` e forza `lazy: false` — core-libs/schematics/src/shared/lib-configs/integration-libs/cds-schematics-config.ts:L46-L46, core-libs/schematics/src/shared/lib-configs/integration-libs/cds-schematics-config.ts:L120-L120.
2. `CdsConfig.cds` ha `baseSite?`, `tenant`, `baseUrl`, `consentTemplateId`, `endpoints`, `merchandising`, `profileTag` — integration-libs/cds/src/config/cds-config.ts:L13-L31; i default **non** includono `endpoints` — integration-libs/cds/src/config/default-cds-config.ts:L9-L23.
3. `cdsConfigValidator` richiede `tenant`, `baseUrl`, `endpoints.strategyProducts` e, se c'è `profileTag`, `configUrl`/`javascriptUrl` — integration-libs/cds/src/config/cds-config-validator.ts:L9-L44; i validator girano solo in dev mode e producono solo `logger.warn` — core-libs/core/src/config/config-validator/config-validator.module.ts:L29-L33, core-libs/core/src/config/config-validator/config-validator.ts:L43-L45.
4. `CdsEndpointsService.getUrl` risolve la chiave in `cds.endpoints`, aggiunge `tenant` se assente, URL-encoda, sostituisce `${...}` con `DynamicTemplate`, gestisce i query param ed infine prefissa `baseUrl` — integration-libs/cds/src/services/cds-endpoints.service.ts:L18-L65, senza duplicarlo — integration-libs/cds/src/services/cds-endpoints.service.ts:L67-L81. `DynamicTemplate.resolve` usa una RegExp globale per variabile — integration-libs/cds/src/utils/dynamic-template.ts:L7-L18.
5. Lo schematic genera `strategyProducts` (`/strategy/${tenant}/strategies/${strategyId}/products` o, con `sciEnabled`, `/strategy/v1/sites/${baseSite}/strategies/${strategyId}/products`) e `searchIntelligence` — core-libs/schematics/src/shared/lib-configs/integration-libs/cds-schematics-config.ts:L73-L79.

3.13 CDS: merchandising carousel

1. `MerchandisingModule` lega `MerchandisingStrategyAdapter → CdsMerchandisingStrategyAdapter` e importa il modulo CMS del carousel — integration-libs/cds/src/merchandising/merchandising.module.ts:L12-L21. L'adapter astratto (senza `@Injectable`) e il connector `providedIn: 'root'` seguono lo schema descritto in 21 — Livello OCC — integration-libs/cds/src/merchandising/connectors/strategy/merchandising-strategy.adapter.ts:L11-L21, integration-libs/cds/src/merchandising/connectors/strategy/merchandising-strategy.connector.ts:L13-L28.
2. Il componente CMS `MerchandisingCarouselComponent` — integration-libs/cds/src/merchandising/cms-components/merchandising-carousel-cms.module.ts:L25-L33 — carica i prodotti al cambio di `strategy`, imposta le CSS custom property di colore e con `using()` attiva il flusso di intersection — integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.ts:L68-L133.
3. `CdsMerchandisingProductService.loadProductsForStrategy` combina site context e user context (`debounceTime(0)`) in una `StrategyRequest` — integration-libs/cds/src/merchandising/facade/cds-merchandising-product.service.ts:L29-L66. Il site context è `{site, language}` — integration-libs/cds/src/merchandising/facade/cds-merchandising-site-context.service.ts:L22-L35; lo user context unisce consent reference (se consenso dato), prodotto (PDP) o categoria (PLP), facet e testo di ricerca — integration-libs/cds/src/merchandising/facade/cds-merchandising-user-context.service.ts:L40-L122.
4. L'adapter in SSR ritorna `{ products: [] }`; in browser fa GET su `strategyProducts` con header `consent-reference` — integration-libs/cds/src/merchandising/adapters/strategy/cds-merchandising-strategy.adapter.ts:L30-L61.

5. Ogni prodotto è caricato con `ProductService.get(id, [DETAILS, PRICE])` e arricchito con metadata `slot/id` — integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.service.ts:L152-L170, integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.service.ts:L193-L203.
6. La soglia di visibilità è `viewportPercentage` del CMS, poi `cds.merchandising.defaultCarouselViewportThreshold`, poi `80` — integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.service.ts:L44-L53. `CarouselViewed` (deduplicato per id modello) e `CarouselClicked` sono push event per ProfileTag — integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.service.ts:L93-L133, integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/model/merchandising-carousel-events.model.ts:L17-L44.
7. I metadata finiscono nel DOM come `data-cx-merchandising-*` tramite `AttributesDirective` — integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.html:L4-L21, integration-libs/cds/src/merchandising/cms-components/directives/attributes/attributes.directive.ts:L29-L47.

3.14 CDS: ProfileTag e tracking

1. `ProfileTagModule` aggiunge due `HTTP_INTERCEPTORS` (`ConsentReferenceInterceptor`, `DebugInterceptor`), lega `CdsBackendNotificationAdapter → OccBackendNotification` e include `provideLoginEventsTracking()` — integration-libs/cds/src/profiletag/profile-tag.module.ts:L16-L32.
2. `TrackingModule` registra un `APP_INITIALIZER` che chiama `TrackingService.trackEvents()` — integration-libs/cds/src/profiletag/tracking/tracking.module.ts:L10-L20, integration-libs/cds/src/profiletag/tracking/tracking.service.ts:L22-L27. `trackEvents` inoltra **sempre** i cambi di consenso e gli altri eventi solo se l'ultimo consenso è `granted` — integration-libs/cds/src/profiletag/tracking/tracking.service.ts:L28-L48. Il consenso osservato è `cds.consentTemplateId` (default `PROFILE`) — integration-libs/cds/src/profiletag/services/profile-tag-lifecycle.service.ts:L28-L54.
3. `ProfileTagPushEventsService` converte eventi Spartacus (pagine categoria/prodotto/ricerca/home/carrello, `OrderPlacedEvent`, modifiche al carrello, merge) in push event — integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L72-L84 — arricchiti con `segments/actions` dell'ultimo `PersonalizationContext` — integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L98-L113. La personalizzazione è descritta in 33 — Tracking, SmartEdit, Qualtrics; lo schematic CDS ne dichiara la dipendenza — core-libs/schematics/src/shared/lib-configs/integration-libs/cds-schematics-config.ts:L49-L49.
4. Il componente CMS `ProfileTagComponent` (`deferLoading: INSTANT`) non ha UI: sottoscrive `profileTagInjector.track()` con `async` — integration-libs/cds/src/profiletag/cms-components/profile-tag-cms.module.ts:L19-L26, integration-libs/cds/src/profiletag/cms-components/profile-tag.component.ts:L12-L21.
5. `track()` aggiunge il tracker e unisce gli eventi interni di ProfileTag con la notifica di login — integration-libs/cds/src/profiletag/services/profile-tag.injector.service.ts:L24-L45. `addTracker` (solo browser, base site truthy e distinto) inietta `profileTag.javascriptUrl` in `<head>` e scrive la config in `window.Y_TRACKING.q` una sola volta — integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L111-L119, integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L154-L211.
6. Gli eventi sono consegnati con `window.Y_TRACKING.eventLayer.push(event)` in try/catch — integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L121-L129.
7. Il consent reference iniziale si legge da `localStorage['profiletag']`; i successivi arrivano dall'evento window `profiletag_consentReferenceLoaded`; il debug da `profiletag_debugFlagChanged` — integration-

libs/cds/src/profiletag/services/profiletag-event.service.ts:L68-L105, integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L142-L152, integration-libs/cds/src/profiletag/model/profile-tag.model.ts:L50-L53.

8. `ConsentReferenceInterceptor` aggiunge `X-Consent-Reference` alle sole richieste OCC — integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts:L24-L45; `DebugInterceptor` aggiunge `X-Profile-Tag-Debug` — integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts:L24-L37.
9. **Login.** `provideLoginEventsTracking` crea un `ReplaySubject(1)` esposto con `LOGIN_EVENTS` e, con `provideAppInitializer`, ascolta `ActionsSubject` emettendo un envelope `{action, timestamp}` per ogni `AuthActions.LOGIN` — integration-libs/cds/src/profiletag/providers/login-events.provider.ts:L20-L48. `loginSuccessful()` deduplica per timestamp — integration-libs/cds/src/profiletag/services/profile-tag-lifecycle.service.ts:L28-L54 — e `OccBackendNotification` fa POST `/users/{userId}/loginnotification` — integration-libs/cds/src/profiletag/adapters/occ-backend-notification-adapter.ts:L22-L35. Poiché anche CDC dispatcha `AuthActions.Login()` (§3.4), un login CDC con CDS installato genera la stessa notifica.

3.15 CDS: recent e trending searches

1. `RecentSearchesModule` registra con `provideOutlet` `RecentSearchesComponent` (AFTER su `SearchBoxOutlets.RECENT_SEARCHES`) e `RecentSearchesHeaderComponent` (REPLACE su `RECENT_SEARCHES_HEADER`) — integration-libs/cds/src/recent-searches/recent-searches.module.ts:L34-L45. Per gli outlet vedi 17 — Outlet.
2. `RecentSearchesService` (solo browser) controlla ogni 250 ms fino a 100 volte se esiste `window.Y_TRACKING.recentSearches` — integration-libs/cds/src/recent-searches/recent-searches.service.ts:L17-L55. Il componente filtra e limita a `maxRecentSearches ?? 5` — integration-libs/cds/src/recent-searches/recent-searches.component.ts:L40-L90. Rimozione singola e "Clear" sono dietro il toggle `searchBoxRecentSearchesRemoval`, applicato nel template con `*cxFeature` (header con "Clear" e bottone di rimozione che chiama `removeFromRecentSearch`) — integration-libs/cds/src/recent-searches/recent-searches.component.html:L4-L9, integration-libs/cds/src/recent-searches/recent-searches.component.html:L40-L42, integration-libs/cds/src/recent-searches/recent-searches.component.ts:L124-L129; il toggle decide anche quali eventi inoltrare alla search box — integration-libs/cds/src/recent-searches/recent-searches.component.ts:L105-L108 — il cui default è `false` — core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1041-L1041.
3. `TrendingSearchesModule` registra `TrendingSearchesComponent` AFTER su `TRENDING_SEARCHES` — integration-libs/cds/src/trending-searches/trending-searches.module.ts:L29-L35. `TrendingSearchesService` attende `Y_TRACKING.config.cdsSiteId`, costruisce l'URL `searchIntelligence` e ripete la GET ogni 15 minuti con `shareReplay(1)` — integration-libs/cds/src/trending-searches/trending-searches.service.ts:L29-L89.

3.16 segment-refs

1. Il public API espone solo `SegmentRefsRootModule` — integration-libs/segment-refs/public_api.ts:L10-L10, integration-libs/segment-refs/root/public_api.ts:L7-L7 — che fornisce l'interceptor e la config di default — integration-libs/segment-refs/root/segment-refs-root.module.ts:L12-L20, integration-libs/segment-refs/root/http-interceptors/index.ts:L11-L17.
2. `segmentRefs.httpHeaderName` ha default `'Segmentrefs'` — integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13.
3. Nel costruttore l'interceptor legge `?segmentrefs=` da `winRef.location.href` e lo salva in localStorage (`segment-refs`), oppure lo rilegge; in browser calcola il nome dell'header in minuscolo — integration-

libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L39-L60.

4. `intercept` aggiunge l'header solo in browser, con header e valore presenti, per URL OCC — integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L69-L86.
5. Lo schematic aggiunge la dipendenza da personalization perché "backend occ api of Segment-refs requires Personalization to be enabled" — core-libs/schematics/src/shared/lib-configs/integration-libs/segment-refs-schematics-config.ts:L36-L39.

3.17 Schematic di installazione e configurazione dei test

`addCdcFeature`, `addCdpFeature` e `addSegmentRefsFeature` eseguono tutte `validateSpartacusInstallation`, `analyzeCrossFeatureDependencies` e poi `chain([analyzeApplication, addFeatures, addPackageJsonDependenciesForLibrary(peerDependencies), finalizeInstallation])` (integration-libs/cdc/schematics/add-cdc/index.ts:L25-L43, integration-libs/cdp/schematics/add-cdp/index.ts:L25-L43, integration-libs/segment-refs/schematics/add-segment-refs/index.ts:L25-L45); la logica specifica (moduli, `importAfter`, config generata) è nelle `*-schematics-config.ts` della sezione 2.4. Gli spec preparano prima la feature da cui dipendono: CDC installa User-Account e User-Profile e verifica che l'opzione `javascriptUrl` finisca nel feature module (integration-libs/cdc/schematics/add-cdc/index_spec.ts:L82-L104); CDP installa prima customer ticketing (integration-libs/cdp/schematics/add-cdp/index_spec.ts:L56-L73); segment-refs installa prima personalization (integration-libs/segment-refs/schematics/add-segment-refs/index_spec.ts:L56-L73).

Le configurazioni Vitest differiscono: `integration-libs/cdc/vitest.config.ts` ha un solo alias (feature toggles testing) e `restoreMocks: true` (integration-libs/cdc/vitest.config.ts:L14-L25); `integration-libs/cds/vitest.config.ts` abbassa le soglie a 80/80/65/80 (integration-libs/cds/vitest.config.ts:L34-L39); `integration-libs/segment-refs/vitest.config.ts` le alza a 90/90/75/90 (integration-libs/segment-refs/vitest.config.ts:L41-L46); `integration-libs/cdp/vitest.config.ts` non dichiara alias (integration-libs/cdp/vitest.config.ts:L13-L25).

---

4. Diagrammi

4.1 Mappa degli override DI

■ **TAVOLA 34.1** diagramma a tutta pagina, in fondo al capitolo

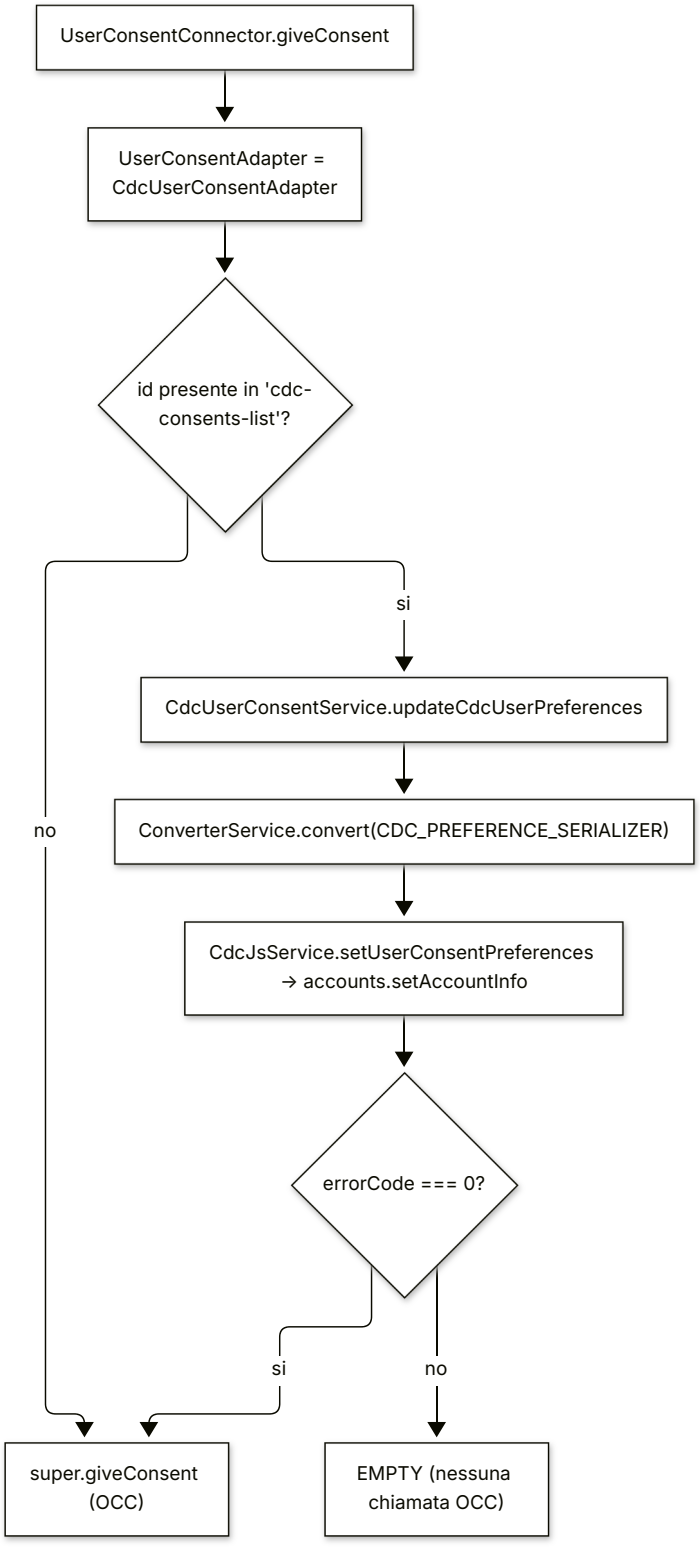
Riferimenti: integration-libs/cdc/root/cdc-root.module.ts:L57-L59, integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L16-L31, integration-libs/cds/src/profiletag/profile-tag.module.ts:L16-L32, integration-libs/segment-refs/root/http-interceptors/index.ts:L11-L17, integration-libs/cds/src/merchandising/merchandising.module.ts:L12-L21, integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L49-L68, integration-libs/cdc/user-profile/register/cdc-register.module.ts:L38-L51, integration-libs/cdc/organization/administration/cdc-administration.module.ts:L23-L62, integration-libs/cdc/organization/user-registration/cdc-b2b-register.module.ts:L33-L38, integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts:L17-L35.

4.2 Login CDC → token Commerce

■ **TAVOLA 34.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L34-L66, integration-libs/cdc/root/service/cdc-js.service.ts:L252-L292, integration-libs/cdc/root/service/cdc-js.service.ts:L165-L192, integration-libs/cdc/root/facade/cdc-auth.facade.ts:L11-L33, core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100, integration-libs/cdc/core/auth/facade/facade-providers.ts:L11-L17, integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts:L25-L59, integration-libs/cdc/core/auth/services/user-authentication/cdc-user-authentication-token.service.ts:L43-L60, integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L68-L145.

4.3 Give consent con CDC



Riferimenti: core-libs/core/src/user/connectors/consent/user-consent.connector.ts:L39-L39, integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L16-L31, integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L35-L65, integration-libs/cdc/root/consent-

management/services/cdc-user-consent.service.ts:L34-L65, integration-libs/cdc/root/service/cdc-js.service.ts:L792-L812.

4.4 ProfileTag: tracking e consent reference

■ **TAVOLA 34.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: integration-libs/cds/src/profiletag/tracking/tracking.module.ts:L10-L20, integration-libs/cds/src/profiletag/tracking/tracking.service.ts:L28-L48, integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L98-L113, integration-libs/cds/src/profiletag/services/profile-tag-lifecycle.service.ts:L28-L54, integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L121-L152, integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts:L24-L45.

---

5. Estratti di codice reale

5.1 `CdcRootModule`: initializer e override dei guard

integration-libs/cdc/root/cdc-root.module.ts:L26-L69

```ts
export function cdcJsFactory(
  cdcJsService: CdcJsService,
  configInit: ConfigInitializerService
): () => Promise<Config> {
  return () =>
    lastValueFrom(
      configInit.getStable('context', 'cdc').pipe(
        tap(() => {
          cdcJsService.initialize();
        })
      )
    );
}
// ...
@NgModule({
  imports: [CdcConsentManagementModule],
  providers: [
    provideDefaultConfigFactory(defaultCdcComponentsConfig),
    { provide: LogoutGuard, useExisting: CdcLogoutGuard },
    { provide: LoginGuard, useExisting: CdcLoginGuard },
    { provide: LoginAsGuestGuard, useExisting: CdcLoginAsGuestGuard },
    {
      provide: APP_INITIALIZER,
      useFactory: cdcJsFactory,
      deps: [CdcJsService, ConfigInitializerService],
      multi: true,
```

```
    },
    provideDefaultConfig(defaultCdcRoutingConfig),
  ],
})
export class CdcRootModule {}
```

5.2 Caricamento dello script Gigya

integration-libs/cdc/root/service/cdc-js.service.ts:L94-L122

```ts
  loadCdcJavascript(): void {
    // Only load the script on client side (no SSR)
    if (isPlatformBrowser(this.platform)) {
      this.subscription.add(
        combineLatest([
          this.baseSiteService.getActive(),
          this.languageService.getActive(),
        ])
          .pipe(take(1))
          .subscribe(([baseSite, language]) => {
            const scriptForBaseSite =
              this.getJavascriptUrlForCurrentSite(baseSite);
            if (scriptForBaseSite) {
              const javascriptUrl = `${scriptForBaseSite}&lang=${language}`;
              this.scriptLoader.embedScript({
                src: javascriptUrl,
                params: undefined,
                attributes: { type: 'text/javascript' },
                callback: () => {
                  this.registerEventListeners(baseSite);
                  this.getSiteConsentDetails(true).subscribe(); //fetch CDC consents and persist to
local storage
                  this.loaded$.next(true);
                  this.errorLoading$.next(false);
                },
                errorCallback: () => {
                  this.errorLoading$.next(true);
                  this.loaded$.next(false);
                },
              });
```

5.3 `onLogin` e il ponte callback → Observable

integration-libs/cdc/root/service/cdc-js.service.ts:L181-L192

```ts
  protected onLoginEventHandler(baseSite: string, response?: any) {
    if (response && !response?.context?.skipOccAuth) {
      //skip re-authentication during reset email
      this.cdcAuth.loginWithCustomCdcFlow(
        response.UID,
        response.UIDSignature,
        response.signatureTimestamp,
        response.id_token !== undefined ? response.id_token : '',
        baseSite
      );
    }
  }
```

integration-libs/cdc/root/service/cdc-js.service.ts:L734-L755

```ts
  protected invokeAPI(methodName: string, payload: Object): Observable<any> {
    return new Observable<any>((result) => {
      const actualAPI = this.getSdkFunctionFromName(methodName);
      if (typeof actualAPI != 'function') {
        result.error('CDC API name is incorrect');
        return;
      }
      actualAPI({
        ...payload,
        callback: (response: any) => {
          this.zone.run(() => {
            if (response?.status === 'OK') {
              result.next(response);
              result.complete();
            } else {
              result.error(response);
            }
          });
        },
      });
    });
  }
```

5.4 La facade lazy

integration-libs/cdc/root/facade/cdc-auth.facade.ts:L11-L33

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: CdcAuthFacade,
      feature: CDC_CORE_FEATURE,
```

```
      methods: ['loginWithCustomCdcFlow', 'loginWithToken'],
      async: true,
    }),
})
export abstract class CdcAuthFacade {
  abstract loginWithCustomCdcFlow(
    UID: string,
    UIDSignature: string,
    signatureTimestamp: string,
    idToken: string,
    baseSite: string
  ): void;

  abstract loginWithToken(
    token: Partial<AuthToken> & { expires_in?: number }
  ): void;
}
```

5.5 Token con `grant_type=custom`

integration-libs/cdc/core/auth/services/user-authentication/cdc-user-authentication-token.service.ts:L43-L60

```ts
    const url = this.authConfigService.getTokenEndpoint();
    const params = new HttpParams()
      .set('client_id', this.authConfigService.getClientId())
      .set('client_secret', this.authConfigService.getClientSecret())
      .set('grant_type', 'custom')
      .set('UID', encodeURIComponent(UID))
      .set('UIDSignature', encodeURIComponent(UIDSignature))
      .set('signatureTimestamp', encodeURIComponent(signatureTimestamp))
      .set('id_token', encodeURIComponent(idToken))
      .set('baseSite', encodeURIComponent(baseSite));

    return this.http
      .post<Partial<AuthToken> & { expires_in?: number }>(url, params)
      .pipe(
        catchError((error: any) => {
          throw tryNormalizeHttpError(error, this.logger);
        })
      );
```

5.6 Scrittura del token e login "a mano"

integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L106-L120

```ts
    stream$.pipe(take(1)).subscribe((canLogin) => {
      if (canLogin) {
        // Code mostly based on auth lib we use and the way it handles token properties
        this.setTokenData(token);

        // OCC specific code
        this.userIdService.setUserId(OCC_USER_ID_CURRENT);

        this.store.dispatch(new AuthActions.Login());

        // Remove any global errors and redirect user on successful login
        this.globalMessageService.remove(GlobalMessageType.MSG_TYPE_ERROR);
        this.authRedirectService.redirect();
      }
    });
```

5.7 Override per sottoclasse: guard

integration-libs/cdc/root/guards/cdc-login.guard.ts:L20-L32

```ts
export class CdcLoginGuard extends LoginGuard {
  constructor(
    protected authService: AuthService,
    protected authConfigService: AuthConfigService,
    protected cmsPageGuard: CmsPageGuard
  ) {
    super(authService, authConfigService, cmsPageGuard);
  }

  protected shouldRenderCMSPage(): Observable<boolean> {
    return of(true); // always load CMS Login Page
  }
}
```

integration-libs/cdc/root/guards/cdc-logout.guard.ts:L43-L55

```ts
  protected logoutFromCdc(): void {
    (this.winRef.nativeWindow as { [key: string]: any })?.[
      'gigya'
    ]?.accounts?.logout();
  }

  /**
   * @override
   * @returns promise to resolve after complete logout
   */
```

```
protected logout(): Promise<any> {
  return Promise.all([super.logout(), Promise.resolve(this.logoutFromCdc())]);
}
```

5.8 Override scoped del login form

integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L44-L68

```ts
  provideDefaultConfig(<AuthConfig>{
    authentication: {
      customLoginPage: undefined,
    },
  }),
  provideDefaultConfig(<CmsConfig>{
    cmsComponents: {
      ReturningCustomerLoginComponent: {
        component: LoginFormComponent,
        guards: [NotAuthGuard],
        providers: [
          {
            provide: LoginFormComponentService,
            useClass: CdcLoginFormComponentService,
            deps: [
              AuthService,
              GlobalMessageService,
              WindowRef,
              CdcJsService,
            ],
          },
        ],
      },
    },
  }),
```

E il meccanismo che lo rende possibile, core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50:

```ts
public getInjector(
  type: string,
  uid: string,
  parentInjector?: Injector
): Injector {
  const configProviders =
    this.cmsComponentsService.getMapping(type)?.providers ?? [];
  return Injector.create({
    providers: [
      {
        provide: CmsComponentData,
```

```
        useFactory: (dataProvider: ComponentDataProvider) => ({
          uid,
          data$: dataProvider.get(uid, type),
        }),
        deps: [ComponentDataProvider],
      },
      provideLcpPresenceForCmsComponent(),
      ...configProviders,
    ],
    parent: parentInjector ?? this.injector,
  });
}
```

5.9 Override con riaccodamento dei provider originali

integration-libs/cdc/organization/administration/cdc-administration.module.ts:L23-L43

```ts
@NgModule({
  providers: [
    //to override UserListService in ListComponent
    provideDefaultConfig(<CmsConfig>{
      cmsComponents: {
        ManageUsersListComponent: {
          providers: [
            {
              provide: UserListService,
              useExisting: CdcUserListService,
            },
            { provide: B2BUserService, useClass: CdcB2BUserService },
            userCmsConfig.cmsComponents?.ManageUsersListComponent?.providers ||
              [],
          ],
        },
      },
    }),
    //to override B2BUserService in UserDetailsComponent, UnitUserListComponent
    { provide: B2BUserService, useClass: CdcB2BUserService },
    //to override B2BUserService in UnitUserRolesCellComponent
```

5.10 Adapter consensi ibrido CDC/OCC

integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L35-L65

```ts
  giveConsent(
    userId: string,
    consentTemplateId: string,
    consentTemplateVersion: number
```

```
  ): Observable<ConsentTemplate> {
    if (!this.cdcConsentsStorage.checkIfConsentExists(consentTemplateId)) {
      return super.giveConsent(
        userId,
        consentTemplateId,
        consentTemplateVersion
      );
    } else {
      return this.cdcUserConsentService
        .updateCdcUserPreferences([
          { id: consentTemplateId, isConsentGranted: true },
        ])
        .pipe(
          catchError((error: any) => throwError(error)),
          switchMap((result) => {
            if (result?.errorCode === 0) {
              return super.giveConsent(
                userId,
                consentTemplateId,
                consentTemplateVersion
              );
            }
            return EMPTY;
          })
        );
    }
  }
```

5.11 CDP: tre righe di override

integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts:L17-L35

```ts
@NgModule({
  imports: [],
  providers: [
    {
      provide: CustomerTicketingCloseComponentService,
      useClass: CdpCustomerTicketingCloseComponentService,
    },
    {
      provide: CustomerTicketingReopenComponentService,
      useClass: CdpCustomerTicketingReopenComponentService,
    },
    {
      provide: CustomerTicketingMessagesComponentService,
      useClass: CdpCustomerTicketingMessagesComponentService,
    },
  ],
```

```
  declarations: [],
})
export class CdpCustomerTicketingModule {}
```

5.12 CDS: `forRoot` e validazione

integration-libs/cds/src/cds.module.ts:L26-L47

```ts
@NgModule({
  imports: [
    ProfileTagModule,
    TrackingModule,
    MerchandisingModule,
    RecentSearchesModule,
    TrendingSearchesModule,
  ],
})
export class CdsModule {
  static forRoot(config?: CdsConfig): ModuleWithProviders<CdsModule> {
    return {
      ngModule: CdsModule,
      providers: [
        provideDefaultConfigFactory(defaultCdsConfigFactory),
        provideDefaultConfig(config),
        provideConfigValidator(cdsConfigValidator),
        ProfileTagPushEventsService,
      ],
    };
  }
}
```

5.13 CDS: strategy adapter (SSR-safe)

integration-libs/cds/src/merchandising/adapters/strategy/cds-merchandising-strategy.adapter.ts:L30-L61

```ts
  loadProductsForStrategy(
    strategyId: string,
    strategyRequest: StrategyRequest = {}
  ): Observable<StrategyProducts> {
    if (!this.windowRef.isBrowser()) {
      return of({ products: [] });
    }

    let headers: HttpHeaders = new HttpHeaders();
    if (strategyRequest.headers && strategyRequest.headers.consentReference) {
      headers = headers.set(
```

```
        'consent-reference',
        strategyRequest.headers.consentReference
      );
    }
    return this.baseSiteService.getActive().pipe(
      take(1),
      switchMap((baseSite) =>
        this.http.get(
          this.cdsEndpointsService.getUrl(
            STRATEGY_PRODUCTS_ENDPOINT_KEY,
            {
              baseSite,
              strategyId,
            },
            strategyRequest.queryParams
          ),
          { headers }
        )
      )
    );
  }
```

5.14 CDS: filtro per consenso

integration-libs/cds/src/profiletag/tracking/tracking.service.ts:L28-L48

```ts
  trackEvents(): void {
    this.profileTagPushEventsService
      .getPushEvents()
      .pipe(
        withLatestFrom(
          this.profileTagLifecycleService.consentChanged().pipe(
            tap((event) => {
              // always notify of consent changes
              this.profileTagEventTracker.notifyProfileTagOfEventOccurrence(
                event
              );
            })
          )
        ),
        filter(([_event, consentChanged]) => consentChanged.data.granted), //don't notify other events
until consent is granted
        tap(([event]) => {
          this.profileTagEventTracker.notifyProfileTagOfEventOccurrence(event);
        })
      )
      .subscribe();
  }
```

5.15 CDS: login events con `provideAppInitializer`

integration-libs/cds/src/profiletag/providers/login-events.provider.ts:L20-L48

```ts
export function provideLoginEventsTracking() {
  const loginEvents$ = new ReplaySubject<LoginEventEnvelope>(1);

  return [
    {
      provide: LOGIN_EVENTS,
      useValue: loginEvents$.asObservable(),
    },
    provideAppInitializer(() => {
      const actionsSubject = inject(ActionsSubject);
      actionsSubject
        .pipe(
          filter(
            (action): action is AuthActions.Login =>
              action.type === AuthActions.LOGIN
          ),
          tap((action) => {
            const envelope: LoginEventEnvelope = {
              action,
              timestamp: Date.now(),
            };
            loginEvents$.next(envelope);
          }),
          takeUntilDestroyed()
        )
        .subscribe();
    }),
  ];
}
```

5.16 segment-refs: interceptor completo

integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L39-L60

```ts
  protected initialize() {
    const url = this.winRef.location.href ?? '';
    const queryParams = new URLSearchParams(url.substring(url.indexOf('?')));
    this.segmentRefs = queryParams.get(this.SEGMENT_REFS_QUERY_PARAM);
    if (this.segmentRefs) {
      this.winRef.localStorage?.setItem(
        this.SEGMENT_REFS_KEY,
        this.segmentRefs
      );
    } else {
      this.segmentRefs = this.winRef.localStorage?.getItem(
```

```
        this.SEGMENT_REFS_KEY
      );
    }
    if (this.winRef.isBrowser()) {
      if (!this.config.segmentRefs?.httpHeaderName && isDevMode()) {
        this.logger.warn(`There is no httpHeaderName configured in Segment`);
      }
      this.requestHeader =
        this.config.segmentRefs?.httpHeaderName?.toLowerCase?.();
    }
  }
```

integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L69-L86

TS

```
  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    if (
      this.winRef.isBrowser() &&
      this.requestHeader &&
      this.segmentRefs &&
      request.url.includes(this.occEndpoints.getBaseUrl())
    ) {
      request = request.clone({
        setHeaders: {
          [this.requestHeader]: this.segmentRefs,
        },
      });
    }
    return next.handle(request);
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cdc[].baseSite	string	nessuno (schematic: BASE_SITE_PLACEHOLDER)	integration-libs/cdc/root/config/cdc-config.ts:L10-L24	integration-libs/cdc/root/service/cdc-js.service.ts:L141-L149, integration-libs/cdc/root/service/cdc-js.service.ts:L406-L417
cdc[].javascriptUrl	string (deve contenere ?apikey=)	nessuno (schematic: JS_SDK_URL_PLACEHOLDER)	integration-libs/cdc/root/config/cdc-config.ts:L10-L24	integration-libs/cdc/root/service/cdc-js.service.ts:L104-L107, integration-libs/cdc/root/service/cdc-js.service.ts:L762-L782

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cdc[].sessionExpiration	number (secondi)	fallback 3600	integration-libs/cdc/root/config/cdc-config.ts:L10-L24	integration-libs/cdc/root/service/cdc-js.service.ts:L406-L417
featureModules.cdc.cmsComponents	string[]	['GigyaRaas Component']	integration-libs/cdc/root/cdc-root.module.ts:L40-L51	core-libs/storefront/cms-structure/services/cms-features.service.ts:L62-L74
featureModules.cdcCore	alias string	'cdc'	integration-libs/cdc/root/cdc-root.module.ts:L40-L51	core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100
routing.routes.{cdcLogin,cdcCheckoutLogin,cdcOrgRegistration,login,loginForm}	RoutesConfig	/cdc/login, /cdc/checkout-login, /cdc/register-org, login, loginForm.paths: []	integration-libs/cdc/root/config/default-cdc-routing-config.ts:L9-L35	routing core (14 — Routing)
authentication.customLoginPage	oggetto / undefined	core {csrfEndpoint:'/csrf', loginFormEndpoint:'/login'}; CDC undefined	integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L44-L48	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L105-L107
cmsComponents.{ReturningCustomerLogin,RegisterCustomer,ForgotPassword,UpdateProfile,UpdatePassword,UpdateEmail}Component.providers	Provider[]	override CDC	integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L49-L68, integration-libs/cdc/user-profile/register/cdc-register.module.ts:L38-L51	core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50
cmsComponents.{ManageUsersList,ManageUnitsList}Component.providers	Provider[]	override CDC + originali	integration-libs/cdc/organization/administration/cdc-administration.module.ts:L23-L62	idem
layoutSlots.GigyaLoginPageTemplate	slot	['BodyContent', 'BottomContent']	integration-libs/cdc/components/gigya-raas/gigya-raas.module.ts:L14-L35	layout storefront (18)
launch.CDC_RECONSENT	LaunchDialog	inlineRoot: true, DIALOG	integration-libs/cdc/user-account/login-form/reconsent/default-cdc-reconsent-layout.config.ts:L10-L18	LaunchDialogService
i18n.chunks.cdc	string[]	['reconsent','cdcProfile']	integration-libs/cdc/assets/translations/translations.ts:L9-L11	i18n (37)
cds.baseSite	string[]	—	integration-libs/cds/src/config/cds-config.ts:L13-L31	solo demo app projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts:L70-L82
cds.tenant	string	''	integration-libs/cds/src/config/default-cds-config.ts:L9-L23	integration-libs/cds/src/services/cds-endpoints.service.ts:L18-L65, integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L154-L211

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cds.baseUrl	string	''	integration-libs/cds/src/config/default-cds-config.ts:L9-L23	integration-libs/cds/src/services/cds-endpoints.service.ts:L67-L81
cds.consentTemplateId	string	'PROFILE'	integration-libs/cds/src/config/default-cds-config.ts:L9-L23	integration-libs/cds/src/profiletag/services/profile-tag-lifecycle.service.ts:L28-L54
cds.endpoints.strategyProducts	template string	nessuno (schematic: §3.12)	integration-libs/cds/src/cds-models/cds-endpoints.model.ts:L7-L15	integration-libs/cds/src/merchandising/adapters/strategy/cds-merchandising-strategy.adapter.ts:L30-L61
cds.endpoints.searchIntelligence	template con ${cdsSiteId}	nessuno	integration-libs/cds/src/cds-models/cds-endpoints.model.ts:L7-L15	integration-libs/cds/src/trending-searches/trending-searches.service.ts:L29-L89
cds.merchandising.defaultCarouselViewportThreshold	number (%)	80	integration-libs/cds/src/config/default-cds-config.ts:L9-L23	integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.service.ts:L44-L53
cds.profileTag.javascriptUrl	string	—	integration-libs/cds/src/config/profile-tag.config.ts:L7-L14	integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L154-L211
cds.profileTag.{configUrl,cdsSiteId,allowInsecureCookies,gtmId,sciEnabled}	vari	allowInsecureCookies: false	integration-libs/cds/src/config/profile-tag.config.ts:L7-L14	copiati in Y_TRACKING.q: integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L154-L211
i18n.chunks.cds	string[]	['cdsRecentSearches','cdsTrendingSearches']	integration-libs/cds/assets/translations/translations.ts:L9-L11	i18n
features.searchBoxRecentSearchesRemoval	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1041-L1041	integration-libs/cds/src/recent-searches/recent-searches.component.html:L4-L9, integration-libs/cds/src/recent-searches/recent-searches.component.html:L40-L42, integration-libs/cds/src/recent-searches/recent-searches.component.ts:L105-L108
segmentRefs.httpHeaderName	string	'Segmentrefs'	integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13	integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L39-L60
query param segmentrefs / localStorage segment-refs	—	—	integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L18-L33	integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L39-L60
localStorage spartacus⁞⁞cdc-consents-list (key cdc-consents-list di StatePersistenceService, contesto vuoto; core-libs/core/src/state/services/state-persistence.service.ts:L105-L109, core-libs/core/src/state/services/state-persistence.service.ts:L124-L131)	lista consensi	—	integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts:L15-L51	idem

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
localStorage `profiletag`	JSON `{cr: {...}}`	`{"cr":{}}`	scritto dallo script ProfileTag (esterno)	`integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L68-L105`

Esempio minimo (forma derivata dai default e dallo schematic, `core-libs/schematics/src/shared/lib-configs/integration-libs/cdc-schematics-config.ts:L161-L171`):

```ts
provideConfig(<CdcConfig>{
  cdc: [
    {
      baseSite: 'electronics-cdc',
      javascriptUrl: 'https://cdns.<region>.gigya.com/JS/gigya.js?apikey=<KEY>',
      sessionExpiration: 3600,
    },
  ],
});
```

DA VERIFICARE: l'URL di esempio è illustrativo (il dominio Gigya non è nel repo); il codice richiede solo che contenga `?apikey=` e accetti l'aggiunta di `&lang=` (`integration-libs/cdc/root/service/cdc-js.service.ts:L104-L107`, `integration-libs/cdc/root/service/cdc-js.service.ts:L762-L782`).

---

7. Commenti e note notevoli dal sorgente

- "@override CDC version of logout guard. In addition to token revocation we invoke logout method from CDC JS lib." — `integration-libs/cdc/root/guards/cdc-logout.guard.ts:L18-L22`. Il parametro `cms` del costruttore è `@deprecated since 2611`, usato solo nel percorso legacy quando `useConfigurableLogoutRedirect` è disabilitato — `integration-libs/cdc/root/guards/cdc-logout.guard.ts:L29-L30`.
- "// by default core is bundled together with components" — `integration-libs/cdc/root/cdc-root.module.ts:L46-L46`: spiega l'alias `cdcCore → cdc`; un'app può separare i chunk ridefinendo `featureModules.cdcCore`.
- "//overriding to avoid same path in different routes" — `integration-libs/cdc/root/config/default-cdc-routing-config.ts:L31-L31`.
- "// Only load the script on client side (no SSR)" — `integration-libs/cdc/root/service/cdc-js.service.ts:L95-L95`.
- Il workaround della verifica password con `skipOccAuth` è documentato estesamente — `integration-libs/cdc/root/service/cdc-js.service.ts:L635-L640`; "//skip re-authentication during reset email" — `integration-libs/cdc/root/service/cdc-js.service.ts:L183-L183`.
- "// Code mostly based on auth lib we use and the way it handles token properties" — `integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L108-L108`: CDC replica il formato di storage della libreria OAuth invece di passare da essa.
- "// Intentional empty method , so csrf token is not initialized in case of CDC" — `integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L69-L69`.
- `selector: 'cx-anonymous-consent-dialog', //reusing existing selector` — `integration-libs/cdc/user-account/login-form/reconsent/cdc-reconsent.component.ts:L38-L39`.

- I commenti di `CdcAdministrationModule` dicono esattamente quale componente ogni override deve raggiungere (`ListComponent`, `UserDetailsComponent`, `UnitUserListComponent`, `UnitUserRolesCellComponent`) — integration-libs/cdc/organization/administration/cdc-administration.module.ts:L25-L25, integration-libs/cdc/organization/administration/cdc-administration.module.ts:L41-L43.
- `// TODO:#cds - check the class name and the name property` su `BrandPageVisitedEvent` — integration-libs/cds/src/profiletag/model/profile-tag.model.ts:L111-L111.
- `startWith(<PageEvent>null), // https://github.com/ReactiveX/rxjs/issues/4772` — integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L137-L137.
- "Uses ReplaySubject to ensure no login events are missed. Provides timestamp to deduplicate login events." — integration-libs/cds/src/profiletag/providers/login-events.provider.ts:L15-L19.
- "If the endpoint to get the url for already has the configured base url appended, do not try and append it again" — integration-libs/cds/src/services/cds-endpoints.service.ts:L68-L71.
- Schematic segment-refs: "Through Spartacus Segment-refs code doesnot have a dependency on Personalization, backend occ api of Segment-refs requires Personalization to be enabled" — core-libs/schematics/src/shared/lib-configs/integration-libs/segment-refs-schematics-config.ts:L36-L39.

8. Extension point

EXTENSION POINT	COME USARLO	RIFERIMENTO	RICETTA
`CdcConfig`	config per base site (`cdc[]`)	integration-libs/cdc/root/config/cdc-config.ts:L10-L24	50 — Ricette config/DI
`CdcJsService` (metodi `protected`: `onLoginEventHandler`, `invokeAPI`, `handleLoginError`…)	sottoclasse + `{ provide: CdcJsService, useClass: MyCdcJsService }` nel root	integration-libs/cdc/root/service/cdc-js.service.ts:L43-L52, integration-libs/cdc/root/service/cdc-js.service.ts:L181-L192	50
`CdcLoginGuard`, `CdcLogoutGuard`, `CdcLoginAsGuestGuard`	ulteriore sottoclasse e override del token core (`LoginGuard`…)	integration-libs/cdc/root/cdc-root.module.ts:L57-L59	50, 51 — Ricette CMS/route
`CdcAuthFacade` / feature `cdcCore`	separare il chunk core ridefinendo `featureModules.cdcCore`	integration-libs/cdc/root/facade/cdc-auth.facade.ts:L11-L33, integration-libs/cdc/root/cdc-root.module.ts:L40-L51	53 — Ricette feature lazy
`CdcLoadUserTokenFailEvent`, `CdcReConsentEvent`	`EventService.get(...)`	integration-libs/cdc/root/events/cdc-event.ts:L13-L31	50
`CDC_PREFERENCE_SERIALIZER`	aggiungere un converter multi (attenzione all'ordine, §9)	integration-libs/cdc/root/consent-management/converters/converter.ts:L11-L13	52 — Ricette OCC/converter
`cmsComponents.<Tipo>.providers`	override scoped di component service (schema CDC)	core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50	51
`MerchandisingStrategyAdapter`	sostituire la sorgente delle strategie	integration-libs/cds/src/merchandising/connectors/strategy/merchandising-strategy.adapter.ts:L11-L21	52
`CdsBackendNotificationAdapter`	sostituire la notifica di login	integration-libs/cds/src/profiletag/profile-tag.module.ts:L16-L32	52

EXTENSION POINT	COME USARLO	RIFERIMENTO	RICETTA
`ProfileTagPushEventsService.addPushEvent(event$)`	aggiungere eventi custom a ProfileTag (prima di `trackEvents`, §9)	`integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L121-L123`	50
`CdsModule.forRoot(config?)` / `CdsConfig`	config CDS (anche via factory, come la demo app)	`integration-libs/cds/src/cds.module.ts:L26-L47`, `projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts:L70-L82`	50
Outlet `SearchBoxOutlets.RECENT_SEARCHES`, `RECENT_SEARCHES_HEADER`, `TRENDING_SEARCHES`	`provideOutlet` con componenti propri	`integration-libs/cds/src/recent-searches/recent-searches.module.ts:L34-L45`, `integration-libs/cds/src/trending-searches/trending-searches.module.ts:L29-L35`	51
`CdpCustomerTicketingModule`	esempio di override di component service in feature lazy	`integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts:L17-L35`	53
`segmentRefs.httpHeaderName`	rinominare l'header	`integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13`	50

Cosa **non** è API pubblica (e quindi a rischio negli upgrade, vedi 54 — API pubblica vs interna): lo store CDC (`integration-libs/cdc/core/public_api.ts:L7-L10`), `CdsEndpointsService` e `recent-searches` (`integration-libs/cds/public_api.ts:L10-L15`), il token `LOGIN_EVENTS` (`integration-libs/cds/src/profiletag/tokens/login-events.token.ts:L20-L22`), `SegmentRefsConfig` e `OccSegmentRefsInterceptor` (`integration-libs/segment-refs/root/public_api.ts:L7-L7`), e i moduli/servizi update-email/password/profile di `@spartacus/cdc/user-profile` (`integration-libs/cdc/user-profile/public_api.ts:L7-L9`).

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Logout Gigya mai eseguito da** `logoutUser()`. `invokeAPI` ritorna un Observable cold e `logoutUser` non si sottoscrive: `accounts.logout` non parte da questo percorso (il logout via route usa invece `CdcLogoutGuard`, che chiama direttamente `gigya.accounts.logout()`) — `integration-libs/cdc/root/service/cdc-js.service.ts:L850-L853`, `integration-libs/cdc/root/service/cdc-js.service.ts:L734-L755`, `integration-libs/cdc/root/guards/cdc-logout.guard.ts:L43-L55`.
2. **Update email senza logout.** In `updateUserEmailWithoutScreenSet` l'Observable con `tap({complete: logoutUser})` non viene sottoscritto: l'update Commerce parte comunque (facade eager) ma il logout no — `integration-libs/cdc/root/service/cdc-js.service.ts:L651-L661`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L82`.
3. **Messaggio di reset password irraggiungibile.** `handleResetPassResponse` gestisce anche `status === 'OK'`, ma è agganciato solo al ramo `error` del `tap` — `integration-libs/cdc/root/service/cdc-js.service.ts:L466-L494`. Il messaggio di successo arriva dal component service (`onSuccess()` se `status === 'OK'`) — `integration-libs/cdc/user-profile/forgot-password/cdc-forgot-password-component.service.ts:L44-L79`.
4. **Letture "sincrone" di stream.** `getCurrentBaseSite`, `getCurrentBaseSiteChannel`, `getUserID` sottoscrivono e ritornano subito: se il valore non è già emesso ottieni `''`/`undefined` (es. `ignoreInterruptions` resta `false`); `getUserID` non fa unsubscribe — `integration-libs/cdc/root/service/cdc-js.service.ts:L436-L453`, `integration-libs/cdc/root/consent-management/services/cdc-user-consent.service.ts:L71-L77`.

5. **Base site non configurato = attesa infinita.** Senza voce `cdc[]` per il sito (o in SSR) lo script non viene caricato e `loaded$` non emette mai: chi attende `didLoad()` resta bloccato (spinner del login) — integration-libs/cdc/root/service/cdc-js.service.ts:L94-L106, integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L34-L66.
6. `apikey` **obbligatorio nell'URL.** `getSiteConsentDetails` estrae `apikey` dalla query di `javascriptUrl`; se manca invia `apiKey: null` — integration-libs/cdc/root/service/cdc-js.service.ts:L762-L782.
7. **Nomi API errati risolti "a metà".** `getSdkFunctionFromName` salta i segmenti mancanti; il nome sbagliato è scartato solo dal controllo `typeof != 'function'` — integration-libs/cdc/root/service/cdc-js.service.ts:L713-L726, integration-libs/cdc/root/service/cdc-js.service.ts:L734-L740.
8. **Side effect nel template.** `GigyaRaasComponent` chiama `displayScreenSet` da un'interpolazione `{{ }}`, protetto solo dal flag `renderScreenSet` — integration-libs/cdc/components/gigya-raas/gigya-raas.component.html:L8-L8, integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L27-L59. Vedi 04 — Change detection.
9. `GigyaRaasGuard` **nega l'accesso se non riesce a leggere i dati del componente.** È un guard di componente CMS: `CmsPageGuardService` esegue solo i guard dei tipi di componente presenti nella pagina, quindi gira solo dove c'è `GigyaRaasComponent` — core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L71-L80, core-libs/storefront/cms-structure/services/cms-guards.service.ts:L41-L41. Lì restituisce `false` se la lookup dei componenti o dei loro dati fallisce o torna vuota (`!componentData.length`, `catchError → of([])`/`of(false)`, dati `{}`) — integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts:L41-L43, integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts:L66-L68, integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts:L95-L95.
10. **Gli array di config si sostituiscono, non si uniscono.** `deepMerge` assegna gli array così come sono: l'override CDC di `ReturningCustomerLoginComponent` elimina `CustomLoginGuard`, e `CdcAdministrationModule` deve riaccodare a mano i provider originali di `ManageUsersListComponent`. Chi aggiunge un proprio override sopra CDC deve fare lo stesso, e l'esito dipende dall'ordine dei chunk di default config (gli schematic usano `importAfter`) — core-libs/core/src/config/utils/deep-merge.ts:L36-L45, feature-libs/user/account/components/login-form/login-form.module.ts:L52-L59, integration-libs/cdc/organization/administration/cdc-administration.module.ts:L23-L62. Vedi 11 — Configurazione.
11. **Override "globali" che non sono globali.** `CdcAdministrationModule`, `CDCB2BRegisterModule` e `CdpCustomerTicketingModule` forniscono provider di NgModule: importati nel wrapper lazy valgono solo nell'injector della feature, e solo se importati **dopo** il modulo originale — projects/storefrontapp/src/app/spartacus/features/organization/administration-wrapper.module.ts:L12-L20, projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts:L13-L22. Importarli nel root non basterebbe se il modulo lazy rifornisce lo stesso token (il provider più vicino vince): `DA VERIFICARE` caso per caso nei moduli feature, perché dipende da dove la feature fornisce il servizio base (per customer ticketing è nel modulo componente, feature-libs/customer-ticketing/components/details/customer-ticketing-close/customer-ticketing-close.module.ts:L44-L45).
12. **Root "pesante".** `CdcRootModule` (eager) importa `LoginAsGuestGuard` da `@spartacus/user/account/components`, e `CdcJsService` dipende da `@spartacus/user/profile/root` e `@spartacus/organization/user-registration/root`: questi entry point finiscono nel bundle iniziale — integration-libs/cdc/root/cdc-root.module.ts:L24-L24, integration-libs/cdc/root/service/cdc-js.service.ts:L26-L27.
13. **Due provider di** `CdcAuthService`**.** È fornito sia da `CdcCoreModule` (`facadeProviders`) sia da `CdcStoreModule`; stessi injector lazy → un'istanza, ma l'effect inietta la classe concreta e non la facade: un override di `CdcAuthFacade` non intercetta `loginWithToken` chiamato dall'effect — integration-libs/cdc/core/cdc-core.module.ts:L13-L17, integration-libs/cdc/core/store/cdc-store.module.ts:L14-L18, integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts:L61-L65.

14. **Effect indirizzi che può terminare.** In `CdcUserAddressesEffects` il `tap({error})` sta dopo `mergeMap` sul flusso principale: un errore CDC si propaga nello stream dell'effect — integration-libs/cdc/core/store/effects/cdc-user-addresses.effect.ts:L27-L61. NgRx lo ri-sottoscrive (default `useEffectsErrorHandler: true`; Spartacus non ridefinisce `EFFECTS_ERROR_HANDLER`) fino a 10 volte, passando ogni errore all'`ErrorHandler`; dopo, un ulteriore errore lo termina definitivamente — deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L9-L13, deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L174-L185, deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L262-L262.
15. **Errore di update email sempre "password non valida"** — integration-libs/cdc/user-profile/update-email/cdc-update-email-component.service.ts:L38-L64.
16. **Serializer multi che ignora il target.** `ConverterService` passa a ogni converter il `target` del precedente (`reduce`), ma `CdcPreferenceSerializer.convert(source, _target)` ignora `_target` e riparte da `{}`: un serializer custom registrato **prima** di quello CDC viene di fatto scartato; uno registrato **dopo** riceve il risultato CDC — core-libs/core/src/util/converter.service.ts:L130-L137, integration-libs/cdc/root/consent-management/converters/cdc-preference.serializer.ts:L18-L38.
17. `ConsentManagementComponentService` **fornito due volte.** Sia da `ConsentManagementModule` (storefront) sia da `CdcConsentManagementModule`: se il modulo storefront è nell'injector più vicino al componente, il suo provider vince — core-libs/storefront/cms-components/myaccount/consent-management/consent-management.module.ts:L47-L48, integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L16-L31. `DA VERIFICARE` per app non generate dagli schematic (dipende da dove è importato `ConsentManagementModule`).
18. **Scope** `'cdc'` **fantasma.** Nessun initializer nel repo dichiara lo scope `cdc`, e `'cdc'` non si sovrappone a `'context'`: l'attesa di `cdcJsFactory` dipende dal solo scope `context`. Se un'app carica la config `cdc[]` da remoto con un `ConfigInitializer`, deve dichiarare `scopes: ['cdc']` perché l'initializer CDC la aspetti — integration-libs/cdc/root/cdc-root.module.ts:L26-L38, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L26-L26, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L117-L122.
19. **CDS senza** `endpoints` **lancia** `TypeError`**.** `getUrl` accede a `this.cdsConfig?.cds?.endpoints[endpoint]` senza optional chaining su `endpoints`, e i default non lo definiscono; il validator avvisa solo in dev mode — integration-libs/cds/src/services/cds-endpoints.service.ts:L18-L25, integration-libs/cds/src/config/default-cds-config.ts:L9-L23, core-libs/core/src/config/config-validator/config-validator.module.ts:L29-L33.
20. **Messaggio del validator fuorviante.** Se manca `profileTag.javascriptUrl` il messaggio parla di `configUrl` (copy-paste) — integration-libs/cds/src/config/cds-config-validator.ts:L20-L25.
21. `getConsentReference()` **può essere** `null`**.** Senza `nativeWindow` ritorna `null`, ma lo user context di merchandising chiama `.pipe` senza controllo quando il consenso è dato — integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L92-L105, integration-libs/cds/src/merchandising/facade/cds-merchandising-user-context.service.ts:L40-L122.
22. **Prodotti senza categorie.** `categoriesToIds(categories)` non gestisce `undefined` — integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L203-L209, integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L406-L408.
23. `addPushEvent` **troppo tardi non ha effetto.** `getPushEvents()` cattura `pushEvents$` al momento della chiamata, cioè nell'`APP_INITIALIZER` di `TrackingModule`: eventi aggiunti dopo non vengono tracciati — integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L98-L123, integration-libs/cds/src/profiletag/tracking/tracking.service.ts:L28-L31.
24. `ProfileTagPushEventsService` **doppio provider.** È `providedIn: 'root'` e fornito anche da `CdsModule.forRoot` — integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L68-L70, integration-libs/cds/src/cds.module.ts:L26-L47.

25. **Interceptor per sottostringa.** I due interceptor CDS e quello segment-refs agiscono su ogni URL che *contiene* `occEndpoints.getBaseUrl()`; `DebugInterceptor` aggiunge l'header anche con valore `false` — integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts:L24-L45, integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts:L24-L37, integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L69-L86.
26. **Config ProfileTag scritta una volta sola.** `exposeConfig` scrive in `Y_TRACKING.q` solo se vuoto: un cambio di base site non aggiorna la config — integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L154-L211.
27. **Polling su oggetti globali.** Recent/trending searches controllano ogni 250 ms fino a 100 volte (~25 s) e trending ripete la GET ogni 15 minuti finché il servizio root vive — integration-libs/cds/src/recent-searches/recent-searches.service.ts:L17-L55, integration-libs/cds/src/trending-searches/trending-searches.service.ts:L29-L89.
28. **Refuso nell'API pubblica:** `getMerchandisingCaourselViewportThreshold` — integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.service.ts:L44-L53.
29. `cds.baseSite` **non è letto dalla libreria**, solo dalla factory della demo app — integration-libs/cds/src/config/cds-config.ts:L13-L31, projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts:L70-L82.
30. **segment-refs in SSR.** L'interceptor legge `winRef.location.href` nel costruttore; in SSR senza `SERVER_REQUEST_URL` `WindowRef.location` lancia un errore — integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L39-L42, core-libs/core/src/window/window-ref.ts:L70-L82. Inoltre `SegmentRefsConfig` non è esportato: la config va scritta senza tipo — integration-libs/segment-refs/root/public_api.ts:L7-L7.
31. **Doppio sistema di login con ASM.** `loginWithToken` rifiuta il login cliente se un agente CS è già loggato — integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L68-L104; vedi 29 — ASM.
32. **Consensi senza lista CDC persistita → `TypeError`.** Se `cdc-consents-list` non è mai stata scritta (script Gigya non caricato, trappola 5, o callback di `getSiteConsentDetails(true)` non ancora arrivato), `readStateFromStorage` ritorna `undefined` e `checkIfConsentExists` chiama `undefined.forEach`: `giveConsent` e `withdrawConsent` lanciano `TypeError` senza chiamare né OCC né CDC — integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts:L57-L76, core-libs/core/src/state/utils/browser-storage.ts:L65-L68, integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L67-L75.
33. **Coverage non abilitata di default in cdc, cdp, cds.** Il blocco `coverage` di queste tre librerie non contiene `enabled: true`, presente invece in segment-refs e nelle feature lib (es. feature-libs/product/vitest.config.ts:L33-L34), e ha il solo reporter `lcov` — integration-libs/cdc/vitest.config.ts:L30-L39, integration-libs/cdp/vitest.config.ts:L26-L35, integration-libs/cds/vitest.config.ts:L24-L33. `DA VERIFICARE`: che le soglie vengano applicate solo con `--coverage` dipende dal default di Vitest, non verificabile senza `node_modules`.
34. **Exclude verso un file inesistente.** cdc, cdp e segment-refs escludono dalla coverage `../../testing/setup-test.ts`, ma il setup effettivo è `testing/setup-vitest.ts` (unico file di setup presente in `testing/` insieme a `testing/patch-object-define-property.ts`) — integration-libs/cdc/vitest.config.ts:L25-L25, integration-libs/cdc/vitest.config.ts:L38-L38, integration-libs/cdp/vitest.config.ts:L34-L34, integration-libs/segment-refs/vitest.config.ts:L38-L38.

10. Come fare debug

- **Lo script Gigya è caricato?** In DevTools verificare che esista `window.gigya` e uno `<script>` con l'URL `javascriptUrl&lang=...` — integration-libs/cdc/root/service/cdc-js.service.ts:L104-L122. Se manca: controllare che `cdc[].baseSite` coincida con il base site attivo — integration-libs/cdc/root/service/cdc-js.service.ts:L141-L149 — e che si sia in browser (non SSR) — integration-libs/cdc/root/service/cdc-js.service.ts:L94-L96.
- `didLoad()` **non emette.** Sottoscrivere `CdcJsService.didLoad()` e `didScriptFailToLoad()` (es. da un componente di debug): `errorLoading$ = true` indica errore di rete, nessuna emissione indica che lo script non è mai stato richiesto (trappola 5).
- **Login CDC non produce sessione Commerce.** Nel tab Network cercare la POST al token endpoint con `grant_type=custom` — integration-libs/cdc/core/auth/services/user-authentication/cdc-user-authentication-token.service.ts:L43-L60. Assente: la facade non ha caricato il chunk (verificare `featureModules.cdc.module`) — projects/storefrontapp/src/app/spartacus/features/cdc/cdc-feature.module.ts:L17-L40. Presente con errore: compare `httpHandlers.badGateway` e viene emesso `CdcLoadUserTokenFailEvent` — integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts:L44-L55. Con NgRx DevTools osservare `[Auth] Load CDC User Token` seguita (in caso di successo) da `AuthActions.Login` — integration-libs/cdc/core/store/actions/cdc-user-token.action.ts:L10-L41.
- **Successo token ma nessun login.** Verificare se c'è un agente ASM loggato (`asm.auth.agentLoggedInError`) — integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L68-L104.
- **Quale servizio è attivo?** Per gli override scoped, ispezionare l'injector del componente CMS (Angular DevTools → Injector tree): `LoginFormComponentService` deve risolvere a `CdcLoginFormComponentService` — core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50. Per gli override nelle feature lazy, verificare l'ordine degli import nel wrapper — projects/storefrontapp/src/app/spartacus/features/user/user-account-wrapper.module.ts:L13-L22.
- **Consensi.** Leggere in localStorage la chiave `spartacus⚿⚿cdc-consents-list` (prefisso `spartacus⚿`, contesto vuoto, separatore ⚿ — core-libs/core/src/state/services/state-persistence.service.ts:L124-L131): se un consenso manca lì, l'adapter usa solo OCC; se la chiave manca del tutto, `giveConsent`/`withdrawConsent` lanciano `TypeError` (§9) — integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L35-L65, integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts:L15-L51.
- **CDS, config.** In dev mode la console mostra i warning del validator (`Please configure cds.tenant...`) — integration-libs/cds/src/config/cds-config-validator.ts:L9-L44. Un `TypeError` in `getUrl` indica `cds.endpoints` mancante (trappola 19).
- **CDS, eventi.** Ispezionare `window.Y_TRACKING.eventLayer` e `window.Y_TRACKING.q` — integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L121-L129, integration-libs/cds/src/profiletag/services/profiletag-event.service.ts:L154-L211. Se compaiono solo eventi `ConsentChanged`, il consenso `PROFILE` non è dato (filtro di `trackEvents`) — integration-libs/cds/src/profiletag/tracking/tracking.service.ts:L28-L48.
- **CDS, header.** Sulle chiamate OCC controllare `X-Consent-Reference` e `X-Profile-Tag-Debug` — integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts:L24-L45, integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts:L24-L37; il carousel in SSR è sempre vuoto per design — integration-libs/cds/src/merchandising/adapters/strategy/cds-merchandising-strategy.adapter.ts:L30-L36.
- **segment-refs.** Aprire una pagina con `?segmentrefs=XYZ`, verificare la chiave `segment-refs` in localStorage e l'header `segmentrefs` (minuscolo) sulle chiamate OCC — integration-libs/segment-refs/root/http-

interceptors/occ-segment-refs.interceptor.ts:L39-L60, integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L69-L86.

- **CDP.** Se i pulsanti di chiusura/riapertura sono ancora visibili, `CdpCustomerTicketingModule` non è nell'injector della feature o è importato prima di `CustomerTicketingModule` — core-libs/schematics/src/shared/lib-configs/integration-libs/cdp-schematics-config.ts:L28-L40.

Per gli interceptor HTTP in generale vedi 06 — HttpClient e interceptor.

---

11. Domande di autoverifica

1. **Perché un login fatto con il form Spartacus in un'app CDC non chiama mai** `AuthService.loginWithCredentials`**?** Perché `CDCLoginFormModule` sostituisce, nell'injector del componente `ReturningCustomerLoginComponent`, `LoginFormComponentService` con `CdcLoginFormComponentService`, il cui `login()` usa `loginUserWithoutScreenSet` (Gigya `accounts.login`); la sessione Commerce nasce poi dall'evento `onLogin`. — integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L49-L68, integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L34-L66, integration-libs/cdc/root/service/cdc-js.service.ts:L181-L192.
2. **Come fa CDC a ottenere un token OAuth Commerce senza passare da** `angular-oauth2-oidc`**?** Con una POST al token endpoint con `grant_type=custom` e i dati firmati di Gigya, poi scrivendo direttamente le chiavi (`access_token`, `expires_at`, …) nell'`AuthStorageService` e dispatchando `AuthActions.Login()`. — integration-libs/cdc/core/auth/services/user-authentication/cdc-user-authentication-token.service.ts:L43-L60, integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L106-L145.
3. **Che cosa carica la prima chiamata a** `CdcAuthFacade.loginWithCustomCdcFlow` **e perché funziona anche senza** `subscribe`**?** Risolve la feature `cdcCore`, alias di `cdc`, e carica `CdcModule`; `facadeFactory` rende la chiamata eager con `connectable(...).connect()`. — integration-libs/cdc/root/facade/cdc-auth.facade.ts:L11-L33, integration-libs/cdc/root/cdc-root.module.ts:L40-L51, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L82.
4. **Perché in un'app CDC la route di login non reindirizza mai al server OAuth anche con Authorization Code Flow?** Perché `LoginGuard` è sovrascritto con `CdcLoginGuard`, il cui `shouldRenderCMSPage()` ritorna sempre `true`, quindi il ramo `loginWithRedirect()` del guard base non si attiva. — integration-libs/cdc/root/cdc-root.module.ts:L57-L59, integration-libs/cdc/root/guards/cdc-login.guard.ts:L20-L32, core-libs/storefront/cms-components/user/login-route/login.guard.ts:L34-L65.
5. **Dare un consenso che esiste solo in OCC fa chiamate a CDC?** No: `CdcUserConsentAdapter.giveConsent` verifica la presenza dell'id in `cdc-consents-list` e, se la lista esiste e non lo contiene, delega a `super.giveConsent` (OCC); se la lista non è mai stata persistita, `checkIfConsentExists` lancia `TypeError` e nessuna delle due chiamate parte (integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts:L69-L76). Se presente, chiama prima CDC e poi OCC solo con `errorCode === 0`. — integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L35-L65.
6. **Perché** `CdcAdministrationModule` **include** `userCmsConfig.cmsComponents?.ManageUsersListComponent?.providers` **nel proprio array?** Perché `deepMerge` sostituisce gli array: senza riaccodarli si perderebbero i provider originali `ListService → UserListService` e `ItemService → UserItemService`. — integration-libs/cdc/organization/administration/cdc-administration.module.ts:L23-L43, feature-libs/organization/administration/components/user/user.config.ts:L42-L53, core-libs/core/src/config/utils/deep-merge.ts:L36-L45.
7. **Un evento di navigazione su PDP viene inviato a ProfileTag se l'utente non ha dato il consenso** `PROFILE`**?** No: `trackEvents` inoltra sempre `ConsentChanged`, ma gli altri eventi passano il `filter` solo se l'ul-

timo consenso è `granted`. — integration-libs/cds/src/profiletag/tracking/tracking.service.ts:L28-L48, integration-libs/cds/src/profiletag/services/profile-tag-lifecycle.service.ts:L28-L54.

8. **Perché il merchandising carousel è vuoto nell'HTML renderizzato lato server?** Perché `CdsMerchandisingStrategyAdapter.loadProductsForStrategy` ritorna `{ products: [] }` quando `windowRef.isBrowser()` è falso. — integration-libs/cds/src/merchandising/adapters/strategy/cds-merchandising-strategy.adapter.ts:L30-L36.
9. **Come arriva l'header `Segmentrefs` alle chiamate OCC e quale nome ha realmente?** `OccSegmentRefsInterceptor` legge `?segmentrefs=` (o localStorage `segment-refs`) nel costruttore e, solo in browser e per URL OCC, aggiunge l'header col nome `segmentRefs.httpHeaderName` in minuscolo (`segmentrefs`). — integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L39-L60, integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L69-L86, integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13.
10. **Che cosa cambia CDP nel customer ticketing e con quale meccanismo?** Sostituisce tre component service con varianti che disabilitano chiusura, riapertura e invio messaggi, tramite provider di NgModule importato dopo `CustomerTicketingModule`. — integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts:L17-L35, integration-libs/cdp/customer-ticketing/cdp-customer-ticketing-close-component.service.ts:L12-L16, core-libs/schematics/src/shared/lib-configs/integration-libs/cdp-schematics-config.ts:L28-L40.

TAVOLA 34.1 **4.1 Mappa degli override DI**

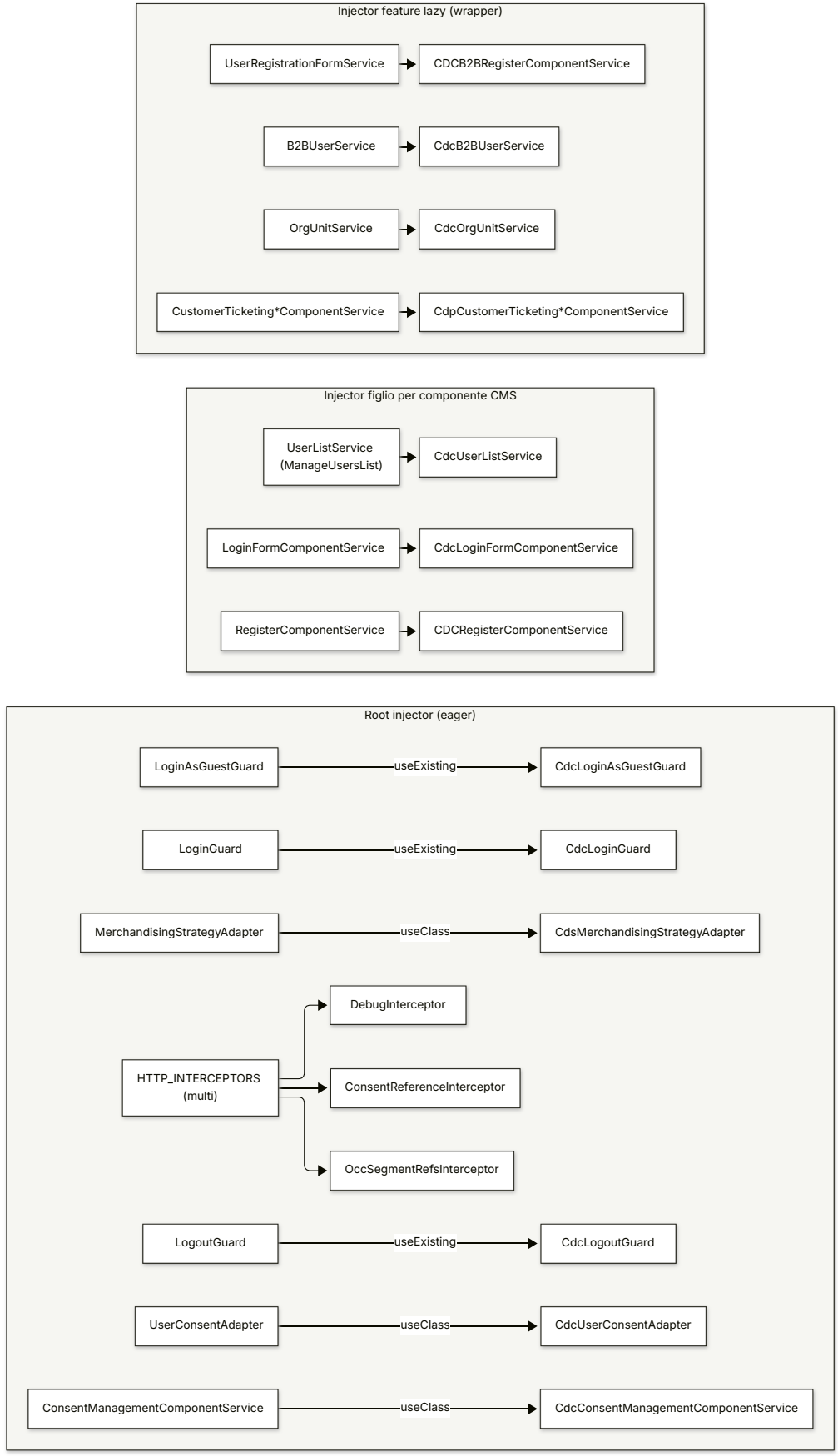


TAVOLA 34.2 **4.2 Login CDC → token Commerce**

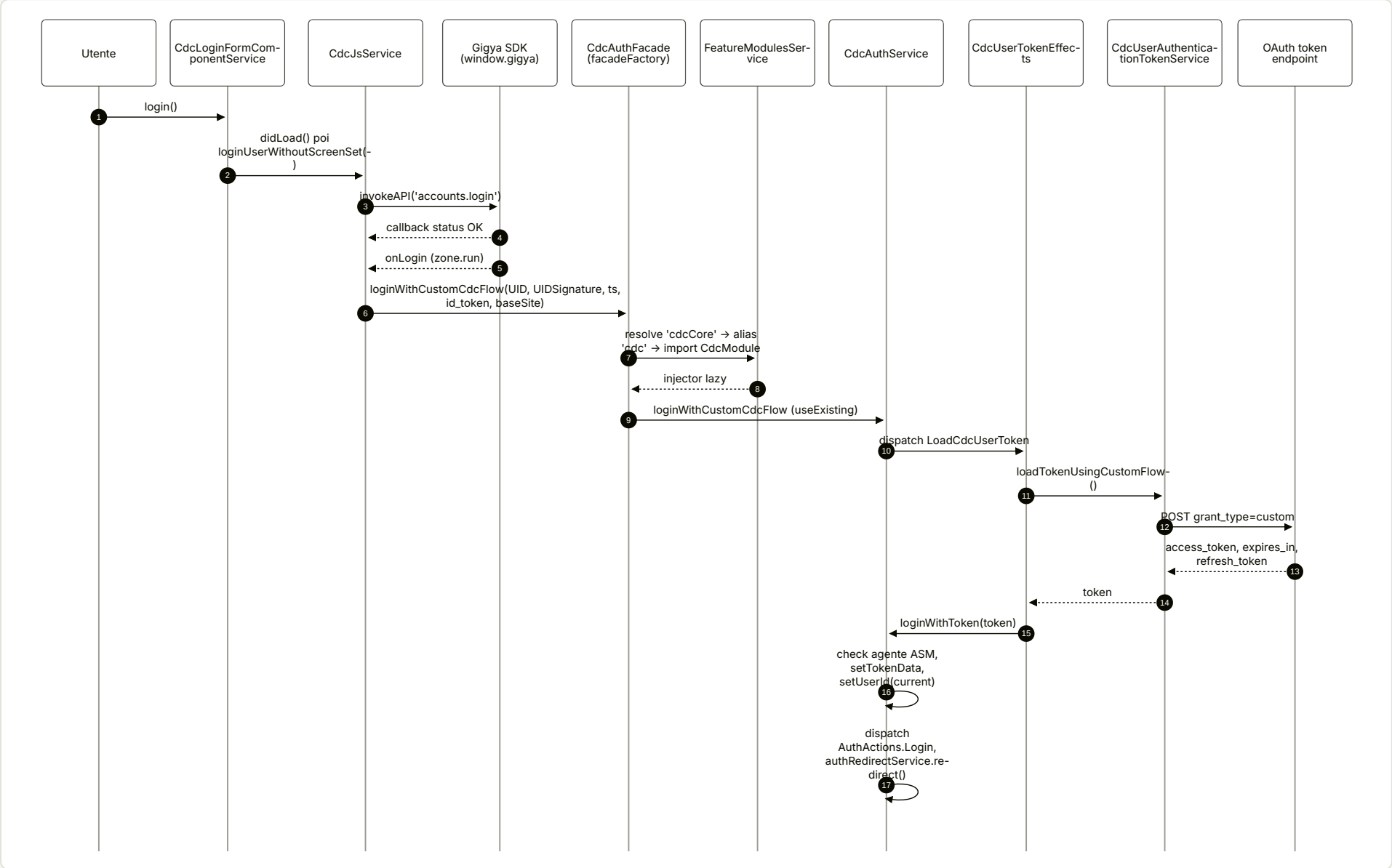
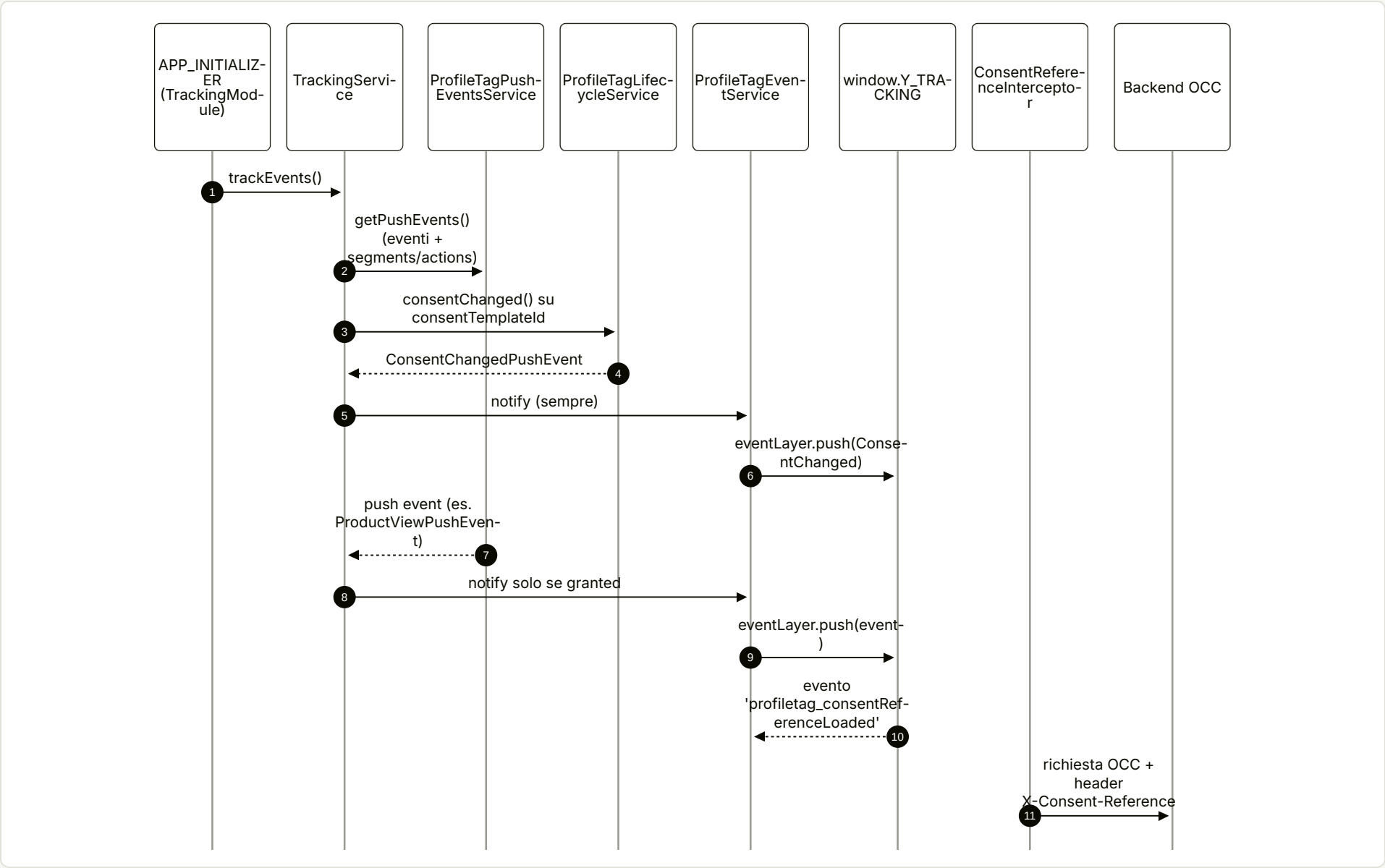


TAVOLA 34.3 **4.4 ProfileTag: tracking e consent reference**



35

Integration libs: OPF, digital payments, OPPS

Open Payment Framework, Digital payments, Omnichannel Promotion Pricing.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Prerequisiti consigliati: 12 — Feature module e lazy loading (featureModules, `facadeFactory`), 20 — Facade, command & query, eventi, 26 — Checkout (flow, step, guard: OPF li sostituisce), 21 — Livello OCC (adapter, converter, endpoint). Collegati: 25 — Carrello, 27 — Ordini (`placePaymentAuthorizedOrder`), 17 — Outlet, 41 — Errori e global message, 42 — Feature toggle, 01 — Dependency injection.

1. A cosa serve e perché esiste

Questo capitolo copre tre librerie di `integration-libs/`, cioè funzionalità che richiedono un backend o un add-on specifico oltre a SAP Commerce standard:

LIBRERIA	COS'È (DAL README)	FORMA NEL CODICE
`@spartacus/opf`	integrazione del "SAP Commerce Open Payment Framework" (`integration-libs/opf/README.md:L1-L3`)	10 sotto-cartelle (base, payment, checkout, b2b-checkout, cta, global-functions, quick-buy, gift-card, tokenisation, order), ciascuna con secondary entry point; il `public_api.ts` radice è vuoto (`integration-libs/opf/public_api.ts:L14-L16`)
`@spartacus/digital-payments`	integrazione di "SAP Digital payments" (`integration-libs/digital-payments/README.md:L1-L3`)	un solo modulo che si innesta nel checkout standard (`integration-libs/digital-payments/src/digital-payments.module.ts:L10-L13`)
`@spartacus/opps`	"Omnichannel Promotion Pricing Service", estensione di Personalization (`integration-libs/opps/README.md:L1-L4`)	solo un root module eager: coupon da URL + guard di login (`integration-libs/opps/root/opps-root.module.ts:L13-L17`)

OPF è di gran lunga la parte più grande ed è il cuore del capitolo. Il suo problema di fondo: il pagamento non passa da OCC ma da un servizio esterno (il "commerce cloud adapter" OPF, a cui si arriva via `opf.opfBaseUrl`, `projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L186-L193`), che restituisce le **configurazioni di pagamento attive** e, per ciascuna, uno di tre *render pattern*: `IFRAME`, `FULL_PAGE`, `HOSTED_FIELDS` (`integration-libs/opf/payment/root/model/opf-payment.model.ts:L250-L254`). Lo storefront deve quindi:

1. leggere le configurazioni attive e mostrarle come opzioni di pagamento (`integration-libs/opf/base/opf-api/adapters/opf-api-base.adapter.ts:L65-L80`);
2. "iniziare" una sessione di pagamento sul servizio OPF e renderizzare quello che il PSP restituisce: un iframe, un redirect/form a pagina intera, oppure script e HTML del PSP (`integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L284-L331`);
3. esporre agli script del PSP un'**API JavaScript globale** (`window.Opf.payments[domain]`) per fare submit, mostrare spinner, ricevere il 3DS (`integration-libs/opf/global-functions/core/facade/opf-global-functions.service.ts:L73-L88`);
4. gestire il **ritorno dal PSP** su una route dedicata che verifica il pagamento e piazza l'ordine (`integration-libs/opf/payment/root/opf-payment-root.module.ts:L18-L39`);
5. piazzare l'ordine con `OrderFacade.placePaymentAuthorizedOrder` invece del place order classico (`integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L229-L231`).

Tre scelte progettuali dominano tutto OPF:

- **Ogni sotto-feature è una mini feature lib.** Root module eager + modulo lazy registrato in `featureModules`; facade astratte in `root` con `facadeFactory`, implementazioni in `core` legate con `useExisting` (`integration-`

libs/opf/base/root/facade/opf-base.facade.ts:L16-L25, integration-libs/opf/base/core/facade/facade-providers.ts:L11-L17). È lo schema del capitolo 12.

- **Override via DI di pezzi di core e di altre feature.** OPF sostituisce `GlobalMessageService`, `CheckoutAuthGuard`, `CheckoutB2BStepsSetGuard`, `CheckoutPaymentTypeFacade`, `UserPaymentAdapter`, e aggiunge normalizer e interceptor multi-provider (tabella in §8). È il tema "override via DI" richiesto dall'outline e lo trattiamo con attenzione perché ogni override ha uno *scope* (root o lazy) diverso.
- **Stato senza NgRx.** Lo stato OPF è un `BehaviorSubject` (`OpfMetadataStoreService`) sincronizzato in storage (integration-libs/opf/base/root/services/opf-metadata-store.service.ts:L11-L39, integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L34-L47); le chiamate backend sono `Query`/`Command` (integration-libs/opf/payment/core/facade/opf-payment.service.ts:L36-L117).

Digital Payments e OPPS sono molto più piccoli e li trattiamo in fondo a §3.

---

2. Mappa dei file

OPF (radice `integration-libs/opf/`)

PATH	RUOLO
`integration-libs/opf/public_api.ts`	Barrel vuoto: solo `types.d.ts` + `export {}` (:L14-L16).
`integration-libs/opf/_index.scss`	Importa gli SCSS delle sotto-feature con stili (base, checkout, b2b-checkout, cta, payment, quick-buy, gift-card, tokenisation; non `global-functions` né `order`) e definisce `$opf-components-allowlist` (:L1-L24).
`integration-libs/opf/base/opf-base.module.ts`	Modulo lazy `OpfBaseModule` = api + core + components (:L12-L15).
`integration-libs/opf/base/root/opf-base-root.module.ts`	App initializer di persistenza, override `GlobalMessageService`, `defaultOpfConfig` (:L16-L32).
`integration-libs/opf/base/root/config/opf-config.ts`	`OpfConfig` (abstract, `useExisting: Config`) + augmentation di `Config` (:L10-L94).
`integration-libs/opf/base/root/config/constants.ts`	Header `sap-commerce-cloud-public-key` / `sap-commerce-cloud-access-code` (:L7-L8).
`integration-libs/opf/base/root/services/opf-metadata-store.service.ts`	Stato OPF in un `BehaviorSubject` (:L11-L39).
`integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts`	Sync con storage, chiave `'opf'` (:L34-L93).
`integration-libs/opf/base/root/services/opf-resource-loader.service.ts`	Caricamento script/CSS del PSP, esecuzione HTML/JS (:L100-L436).
`integration-libs/opf/base/root/services/opf-global-message.service.ts`	`GlobalMessageService` con chiavi silenziabili (:L19-L60).

PATH	RUOLO
integration-libs/opf/base/root/events/opf-event.listener.ts	Reset metadata su login/ordine/nuovo carrello (:L27-L35).
integration-libs/opf/base/core/services/opf-endpoints.service.ts	Costruzione URL OPF API (:L40-L96).
integration-libs/opf/base/opf-api/adapters/opf-api-base.adapter.ts	GET active configurations (:L65-L118).
integration-libs/opf/payment/root/opf-payment-root.module.ts	Route di verifica + location initializer (:L18-L49).
integration-libs/opf/payment/root/facade/opf-payment.facade.ts	OpfPaymentFacade (7 metodi, feature opfPayment) (:L22-L39).
integration-libs/opf/payment/root/components/opf-payment-verification/*	Componente e servizio della pagina di ritorno dal PSP.
integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts	Submit / submit-complete e gestione dello status (:L62-L235).
integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts	verify/submit/submitComplete/afterRedirect/initiate/update (:L85-L308).
integration-libs/opf/payment/occ/adapters/occ-opf-payment.adapter.ts	PUT paymentOption sul carrello OCC (:L29-L71).
integration-libs/opf/checkout/root/opf-checkout-root.module.ts	Override CheckoutAuthGuard, flow OPF, routing, app initializer orchestratore (:L26-L63).
integration-libs/opf/checkout/components/opf-checkout-payments/*	Lista opzioni di pagamento.
integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/*	Iniziazione pagamento e rendering PSP (service, component, directive sandbox).
integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts	Override CheckoutB2BStepsSetGuard, flow OPF B2B (:L19-L48).
integration-libs/opf/b2b-checkout/core/facade/facade-providers.ts	Override CheckoutPaymentTypeFacade (:L11-L17).
integration-libs/opf/cta/**	Script "call to action" del PSP su PDP/carrello/ordine.
integration-libs/opf/global-functions/core/facade/opf-global-functions.service.ts	Registrazione window.Opf.payments[domain] (:L32-L102).
integration-libs/opf/global-functions/core/services/**	Implementazione delle funzioni per i domini checkout/global/redirect.

PATH	RUOLO
integration-libs/opf/quick-buy/**	Google Pay / Apple Pay "quick buy" (anche da PDP con carrello dedicato).
integration-libs/opf/gift-card/**	Gift card come metodo STORED_VALUE_PAYMENT, normalizer e interceptor.
integration-libs/opf/tokenisation/**	Carte salvate, override UserPaymentAdapter.
integration-libs/opf/order/opf-order.module.ts	Normalizer ordine (billing address) (:L11-L20).

Digital Payments e OPPS

PATH	RUOLO
integration-libs/digital-payments/public_api.ts	Esporta src/checkout/index e DigitalPaymentsModule (:L10-L11).
integration-libs/digital-payments/src/checkout/dp-checkout.module.ts	Adapter, normalizer, endpoint, DpCheckoutPaymentService (:L21-L42).
integration-libs/digital-payments/src/checkout/adapters/occ-digital-payments.adapter.ts	POST request/response con query param sid/sign (:L36-L95).
integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts	Rimappa il CMS CheckoutPaymentDetails (:L23-L45).
integration-libs/digital-payments/src/checkout/facade/dp-checkout-payment.service.ts	Query request + Command details (:L36-L114).
integration-libs/opps/root/opps-root.module.ts	OppsRootModule (:L13-L17).
integration-libs/opps/root/config/default-opps-config.ts	Default nomi header/param/storage (:L9-L20).
integration-libs/opps/root/coupon-codes/**	Location initializer + interceptor coupon.
integration-libs/opps/root/login-required/*	OppsLoginRequiredGuard su BEFORE_CMS_PAGE_GUARD.

Wiring nell'app demo

PATH	RUOLO
projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts	9 root module eager, 9 featureModules, config OPF, scelta B2B/B2C (:L73-L197).
projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts	OpfOrderModule dentro il chunk order (:L21-L26).
projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts	DigitalPaymentsModule dentro il chunk checkout (:L22-L33).
projects/storefrontapp/src/app/spartacus/features/opps/opps-feature.module.ts	Import eager di OppsRootModule (:L10-L13).

File aggiuntivi

PATH	RUOLO
`integration-libs/digital-payments/assets/public_api.ts`	entry point `assets` di digital-payments: riesporta `translations/translations` (chunk `dpPaymentForm` e traduzioni per lingua)
`integration-libs/digital-payments/schematics/add-digital-payments/index.ts`	rule `ng add` di `@spartacus/digital-payments` (`addDigitalPaymentsFeature`)
`integration-libs/digital-payments/schematics/add-digital-payments/index_spec.ts`	test Jest dello schematic DP (registra la collection checkout e installa prima checkout base)
`integration-libs/opf/schematics/add-opf/index.ts`	rule `ng add` di `@spartacus/opf` (`addOpfFeature`)
`integration-libs/opf/schematics/add-opf/index_spec.ts`	test Jest dello schematic OPF per le feature base, checkout, payment, CTA, global functions, quick buy, gift card, tokenisation
`integration-libs/opps/public_api.ts`	entry point primario `@spartacus/opps`: riesporta `root/public_api`
`integration-libs/opps/schematics/add-opps/index.ts`	rule `ng add` di `@spartacus/opps` (`addOppsFeature`)
`integration-libs/opps/schematics/add-opps/index_spec.ts`	test Jest dello schematic OPPS (richiede prima personalization)

3. Come funziona passo per passo

3.1 Wiring: root module eager, moduli lazy, scelta B2B/B2C

1. **Nove root module eager.** L'app importa `OpfBaseRootModule`, `OpfPaymentRootModule`, `OpfCheckoutRootModule`, `OpfB2bCheckoutRootModule`, `OpfCtaRootModule`, `OpfGlobalFunctionsRootModule`, `OpfQuickBuyRootModule`, `OpfGiftCardRootModule`, `OpfTokenisationRootModule` (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L92-L103). Questi contengono solo config di default, facade proxy, route, override DI e i componenti che devono esistere subito.
2. **Nove moduli lazy.** Ogni feature name punta a un `import()` dinamico in `featureModules`; per `OPF_QUICK_BUY_FEATURE` l'app carica la variante `@spartacus/opf/quick-buy/pickup-in-store` (`OpfQuickBuyPickupInStoreModule`) e non `OpfQuickBuyModule` (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L105-L152).
3. **Facade proxy.** Una facade OPF è una classe astratta `providedIn: 'root'` con `useFactory: () => facadeFactory({facade, feature, methods})` (integration-libs/opf/base/root/facade/opf-base.facade.ts:L16-L25). `facadeFactory` delega a `FacadeFactoryService.create`, che restituisce un proxy che carica la feature "at first method call" (core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L10-L23). Dentro il modulo lazy, `facadeProviders` lega la facade al servizio concreto con `useExisting` (integration-libs/opf/base/core/facade/facade-providers.ts:L11-L17).
4. **Config di base.** `defaultOpfConfig` mette `opfBaseUrl: ''` e `paymentOption.enableInfoMessage: true` (integration-libs/opf/base/root/config/default-opf-config.ts:L9-L16); l'app fornisce URL reale, una public key segnaposto e `enableGetActiveConfigurationsAccessCodeHeader: true` (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L186-L193).
5. **B2B o B2C a build time.** Sia `OpfCheckoutRootModule` sia `OpfB2bCheckoutRootModule` forniscono come *default* un `checkout.flows.OPF` (3 step il primo, 5 il secondo) (integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L56-L56, integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.

module.ts:L43-L43). Poiché deepMerge sostituisce i valori non oggetto, array di step compresi (core-libs/core/src/config/utils/deep-merge.ts:L31-L46), i due default si sovrascrivono; l'app decide esplicitamente con provideConfig in base a environment.b2b (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L73-L81). Allo stesso modo le config gift card sono aggiunte solo se !environment.b2b && environment.opf (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L85-L91).

6. **Order fuori da OPF lazy.** OpfOrderModule non è una feature: l'app lo mette accanto a OrderModule dentro OrderWrapperModule (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L21-L26), e il modulo aggiunge OpfOccOrderNormalizer a ORDER_NORMALIZER, che copia sapBillingAddress in paymentInfo.billingAddress se assente (integration-libs/opf/order/opf-order.module.ts:L11-L20, integration-libs/opf/order/opf-occ-order-normalizer.ts:L13-L28).
7. **Feature toggle.** Tre toggle di core, default false, cambiano il comportamento OPF: opfPaymentVerificationCheckProcessingCartOnErrorOnly, opfUseDestroyRef, opfCheckoutUseUpdatePaymentTransaction; l'app demo li abilita tutti (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1074-L1083, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L364-L371).

3.2 OPF base: stato, persistenza, URL, active configurations

1. **All'avvio** OpfBaseRootModule registra con provideAppInitializer una funzione che chiama OpfMetadataStatePersistanceService.initSync() (integration-libs/opf/base/root/opf-base-root.module.ts:L19-L24); in Angular provideAppInitializer è solo un valore del multi-token APP_INITIALIZER (angular:packages/core/src/application/application_init.ts:L192-L202).
2. initSync collega lo stato a StatePersistenceService.syncWithStorage con chiave 'opf'; alla lettura chiama updateOpfMetadata (integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L34-L47, integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L89-L93).
3. **Cosa viene persistito.** Tutto il metadata tranne opfPaymentSessionId e opfPaymentSessionConfigurationId, che sono salvati solo se is3DSRedirect è vero, perché devono sopravvivere al reload della pagina del PSP (integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L62-L83).
4. **Lo stato** è un BehaviorSubject<OpfMetadataModel> con stato iniziale congelato (termsAndConditionsChecked, selectedPaymentOptionId, isPaymentInProgress, opfPaymentSessionId, is3DSRedirect, opf3DSRedirectReturnPath, ...) e tre operazioni: get, update parziale, clear (integration-libs/opf/base/root/services/opf-metadata-store.service.ts:L11-L39).
5. **Reset.** OpfEventListenerService azzera lo stato su LoginEvent, OrderPlacedEvent, CreateCartEvent; viene istanziato perché OpfEventModule lo inietta nel costruttore (integration-libs/opf/base/root/events/opf-event.listener.ts:L27-L35, integration-libs/opf/base/root/events/opf-event.module.ts:L10-L15).
6. **Override di** GlobalMessageService**.** Lo stesso root module fornisce { provide: GlobalMessageService, useExisting: OpfGlobalMessageService } (integration-libs/opf/base/root/opf-base-root.module.ts:L25-L28). OpfGlobalMessageService estende il servizio di core e scarta i messaggi la cui key è in disabledKeys mentre il blocco è attivo; disableGlobalMessage(keys, timeout) lo attiva per timeout ?? 2000 ms (integration-libs/opf/base/root/services/opf-global-message.service.ts:L19-L60). Effetto: **tutta l'app** (non solo OPF) usa questa sottoclasse.
7. **URL OPF.** OpfEndpointsService.buildUrl compone ${opfBaseUrl}/${activeBaseSite}/${endpoint}, risolvendo urlParams con StringTemplate.resolve e aggiungendo queryParams con HttpParamsURIEncoder (integration-libs/opf/base/core/services/opf-endpoints.service.ts:L40-L75). L'endpoint viene da backend.opfApi.endpoints, che OpfApiConfig aggiunge a BackendConfig per augmentation (integration-

libs/opf/base/core/services/opf-endpoints.service.ts:L77-L88, integration-libs/opf/base/root/config/opf-api-config.ts:L11-L21). L'interfaccia `OpfApiEndpoints` è vuota e ogni sotto-feature la estende (integration-libs/opf/base/root/model/opf-api-endpoints.model.ts:L7-L7, integration-libs/opf/base/opf-api/model/opf-api-base-endpoints.model.ts:L9-L16).

8. **Base site.** `_activeBaseSite` è riempito in modo asincrono dal costruttore con `baseSiteService.getActive().subscribe(...)` (integration-libs/opf/base/core/services/opf-endpoints.service.ts:L26-L38).
9. **Active configurations.** `OpfBaseService.getActiveConfigurationsState(query)` crea **a ogni chiamata** una nuova `Query` su `OpfBaseConnector.getActiveConfigurations(query)` e ne restituisce `getState()` (integration-libs/opf/base/core/facade/opf-base.service.ts:L23-L34). L'adapter fa GET con header JSON + `Accept-Language`, la public key e, se `enableGetActiveConfigurationsAccessCodeHeader`, l'access code del carrello ottenuto da `CartAccessCodeFacade.getCartAccessCode(userId, cartId)`; normalizza l'errore con `tryNormalizeHttpError` e applica `OPF_ACTIVE_CONFIGURATIONS_NORMALIZER` (integration-libs/opf/base/opf-api/adapters/opf-api-base.adapter.ts:L65-L118). Endpoint di default: `active-configurations` (integration-libs/opf/base/opf-api/config/default-opf-api-base-config.ts:L9-L17).
10. **Modello.** Una `OpfActiveConfiguration` ha `id`, `providerType`, `paymentType`, `displayName`, `logoUrl`, `dynamicScript`…; `OpfPaymentProviderType` = `PAYMENT_GATEWAY` | `PAYMENT_METHOD` | `STORED_VALUE_PAYMENT` (integration-libs/opf/base/root/model/opf-base.model.ts:L12-L23, integration-libs/opf/base/root/model/opf-base.model.ts:L84-L88).

3.3 Il resource loader: come arriva il codice del PSP nella pagina

`OpfResourceLoaderService` è il punto in cui codice di terze parti entra nello storefront.

1. **SSR: niente.** `loadResources` in server risolve subito, con il commento "SSR mode not supported for security concerns" (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L397-L406).
2. **Locale o esterno.** Se `opf.localPspResources[paymentOptionId]` esiste si usano quei file JS/CSS, altrimenti gli URL ricevuti dal backend (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L408-L416, integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L316-L345).
3. **Script.** Iniettati con `ScriptLoader.embedScript` e `disableKeyRestriction: true`; con `sri` si impostano `integrity` e `crossOrigin` (default `'anonymous'`) (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L124-L149).
4. **Marcatura per la pulizia.** Ogni risorsa riceve `data-opf-resource="true"`, salvo attributo `opf-load-once: 'true'`; `clearAllResources` rimuove tutti i nodi marcati (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L100-L115, integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L186-L194).
5. **Modalità `SEPARATE`.** Se `dynamicScript.htmlContentMode === SEPARATE`, il JS arriva in `jsContent` e il contesto in `jsContext` (JSON); il CSS `cssUrl+cssHash` è aggiunto agli stili con SRI (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L277-L311). Dopo il caricamento, `executeScriptWithContext` imposta `window.OpfContext`, inserisce uno `<script data-opf-script>` **fuori dalla zone** e lo rimuove dopo 100 ms (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L232-L261, integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L429-L435).
6. **Modalità HTML classica.** `executeScriptFromHtml` (solo browser) estrae il primo `<script>` dall'HTML con `DOMParser` ed esegue il contenuto con `Function(...)()` (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L174-L184).

3.4 Checkout OPF B2C: dal flow alla scelta dell'opzione

1. **Flow per payment provider.** `CheckoutFlowOrchestratorService` (lib checkout) legge `baseSite.baseStore.paymentProvider` e, se esiste `checkout.flows[paymentProvider]`, restituisce quel flow invece della config standard (feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L20-L44); `CheckoutStepService.resetSteps` usa gli step di quel flow (feature-libs/checkout/base/components/services/checkout-step.service.ts:L88-L94). OPF definisce il flow `OPF` (integration-libs/opf/checkout/root/model/opf-checkout.model.ts:L13-L13).
2. **Istanziazione precoce.** `OpfCheckoutRootModule` forza la creazione dell'orchestratore con `provideAppInitializer(() => { inject(CheckoutFlowOrchestratorService); })` (integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L58-L60): il costruttore sottoscrive `getPaymentProvider()` (con `take(1)`) e memorizza il nome (feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L20-L33).
3. **Step B2C.** `deliveryAddress` (route standard `checkoutDeliveryAddress`), `deliveryMode`, `opfPaymentAndReview` (route `opfCheckoutPaymentAndReview`, tipo `PAYMENT_DETAILS`) con `guest: false` (integration-libs/opf/checkout/root/config/default-opf-checkout-config.ts:L12-L45). Route semantiche OPF: `checkout/opf-payment-type`, `checkout/opf-delivery-address`, `checkout/opf-payment-and-review`, `checkout/opf-review`, `opf-checkout-email` (integration-libs/opf/checkout/root/config/default-opf-checkout-routing-config.ts:L9-L29).
4. **Guard sostituito.** Il root module fornisce `{ provide: CheckoutAuthGuard, useClass: OpfCheckoutAuthGuard }` (integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L44-L47). `OpfCheckoutAuthGuard` estende il guard standard: carrello non guest → `super.canActivate()`; carrello guest → verifica via OCC (`users/${userId}/carts/${cartId}?fields=sapCustomerEmail`) che il guest abbia un'email, altrimenti `UrlTree` verso `opfCheckoutEmail` (integration-libs/opf/checkout/root/checkout-guard/opf-checkout-auth.guard.ts:L33-L62, integration-libs/opf/checkout/root/config/default-occ-opf-checkout-config.ts:L9-L18). La pagina email aggiorna il guest con `CartGuestUserFacade.updateCartGuestUser`, ricarica il carrello e va a `checkoutDeliveryAddress` (integration-libs/opf/checkout/components/opf-checkout-email-update/opf-checkout-email-update.component.ts:L75-L91).
5. **Componente CMS.** `OpfCheckoutPaymentAndReview` (feature lazy `opfCheckout`) è mappato su `OpfCheckoutPaymentAndReviewComponent` con guard `[CheckoutAuthGuard, CartNotEmptyGuard]` (integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L26-L40, integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.module.ts:L57-L66). Il componente estende `CheckoutReviewSubmitComponent`, gestisce il form T&C con `Validators.requiredTrue` e scrive `termsAndConditionsChecked` nel metadata store (integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L74-L141).
6. **T&C esplicito.** È attivo se il JSON della pagina CMS contiene la stringa `OpfExplicitTermsAndConditionsComponent` (integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L103-L131, integration-libs/opf/checkout/root/model/opf-checkout.model.ts:L10-L11). Con T&C esplicito non accettato la preselezione dell'opzione di pagamento viene azzerata (integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L266-L322).
7. **Carrello solo pickup.** Se non ci sono delivery items il componente cancella l'indirizzo di consegna e imposta delivery mode `'pickup'` (integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L147-L161).
8. **Lista opzioni.** `OpfCheckoutPaymentsComponent` chiede le configurazioni paginando (`pageSize`, `pageNumber: index+1`), esclude `STORED_VALUE_PAYMENT` (gestite dalla gift card) e, in modalità "solo wrapper", tiene solo quella selezionata (integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L199-L238). Il messaggio informativo viene da `opf.paymentOption.`

`paymentInfoMessagesMap[paymentId]` con fallback `opfCheckout.defaultPaymentInfoMessage` (integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L244-L260). La selezione scrive `selectedPaymentOptionId` nel metadata store (integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L350-L357).

3.5 Iniziazione del pagamento e rendering del PSP

1. **Iniziazione.** `OpfCheckoutPaymentWrapperService.initiatePayment(paymentOptionId)` emette lo stato "loading", pulisce le risorse, imposta `isPaymentInProgress: true`, ottiene l'OTP con `CartAccessCodeFacade.getCartAccessCode` e chiama `OpfPaymentFacade.initiatePayment` (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L103-L168).
2. **Config di iniziazione.** `configurationId = String(paymentOptionId)`, `channel: BROWSER`, `browserInfo`, e `resultURL`/`cancelURL` calcolati con `routingService.getFullUrl` sulle route di verifica (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L399-L418). `getBrowserInfo` raccoglie colorDepth, lingua, schermo, userAgent, origin e timezone (integration-libs/opf/payment/core/utils/opf-payment-utils.ts:L9-L24).
3. **Update invece di initiate.** Con il toggle `opfCheckoutUseUpdatePaymentTransaction` e una sessione salvata per la *stessa* configurazione, si chiama `updatePaymentTransaction`; un 409 ricade su `initiatePayment` (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L137-L168, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L239-L255). La sessione viene salvata solo con il toggle o per pattern `FULL_PAGE` (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L192-L208).
4. **HTTP.** `OpfApiPaymentAdapter.initiatePayment` passa la config per `OPF_PAYMENT_CONFIG_SERIALIZER` e fa POST di `paymentConfig.config` su `payments`, con public key e `otpKey` nell'header access code; `updatePaymentTransaction` fa PUT su `payments/${paymentSessionId}` (integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts:L196-L256, integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L9-L23).
5. **Rendering.** `renderPaymentGateway`: se c'è `dynamicScript` carica le risorse (vedi §3.3), emette l'evento di rendering e, salvo `SEPARATE`, esegue l'HTML; altrimenti se c'è `destination` emette l'evento con l'URL; altrimenti errore generico. `dynamicScript` ha la precedenza (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L275-L331). L'HTML viene eseguito solo se l'utente è ancora sulla route `opfCheckoutPaymentAndReview` (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L74-L93).
6. **Template.** `FULL_PAGE` diventa link o form `_self`, `IFRAME` un `<iframe>` (eventualmente alimentato da un form POST con target `cx-payment-iframe`) con direttiva `cxOpfSandbox` (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.html:L15-L69). La direttiva imposta o rimuove `sandbox`; il valore viene da `opf.paymentOption.iframeSandboxMap[paymentOptionId]` (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-iframe-sandbox.directive.ts:L9-L25, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.ts:L111-L117).
7. **Hosted fields.** Se la risposta ha `paymentSessionId` e pattern `HOSTED_FIELDS`, il componente registra le global functions del dominio `checkout`, altrimenti le rimuove; in `ngOnDestroy` le rimuove e pulisce le risorse (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.ts:L208-L253, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.ts:L125-L131).
8. **Re-iniziazione.** Al cambio di lingua o valuta (saltato il valore iniziale, atteso il carrello stabile) il wrapper emette `reinitiatePaymentEvent$` (integration-libs/opf/checkout/components/opf-checkout-payment-

wrapper/opf-checkout-payment-wrapper.component.ts:L164-L199). Lo stesso canale è usato dalla gift card e dallo script PSP (`reinitiatePaymentForm`) (integration-libs/opf/payment/root/services/opf-payment-events.service.ts:L14-L50).

9. **409 all'iniziazione** = "pagamento già fatto": si piazza l'ordine e si va a conferma (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L333-L367).

3.6 Global functions: l'API JavaScript per gli script del PSP

1. **Domini.** `OpfGlobalFunctionsDomain = checkout | global | redirect` (integration-libs/opf/global-functions/root/model/opf-global-functions.model.ts:L10-L14).
2. **Registrazione.** `OpfGlobalFunctionsService.registerGlobalFunctions` (solo browser) crea se serve `window.Opf.payments[domain]` e delega al servizio di registrazione del dominio; `unregisterGlobalFunctions` imposta il contenitore a `undefined` (integration-libs/opf/global-functions/core/facade/opf-global-functions.service.ts:L32-L102).
3. **Dominio** `checkout` (wrapper hosted fields e ritorno 3DS): `submit`, `submitComplete`, `throwPaymentError`, start/stop load indicator, `reinitiatePaymentForm`, `handle3DSRedirect`, `updatePaymentTransaction` (integration-libs/opf/global-functions/core/services/domains/checkout/opf-global-functions-checkout-domain-registrations.service.ts:L21-L45).
4. **Dominio** `redirect` (pagina di verifica, pattern after-redirect): registra solo se c'è `paymentSessionId`: `submit`, `submitCompleteRedirect` (con returnPath `opfCheckoutPaymentAndReview`), `getRedirectParams` (integration-libs/opf/global-functions/core/services/domains/redirect/opf-global-functions-redirect-domain-registrations.service.ts:L24-L38, integration-libs/opf/global-functions/core/services/domains/redirect/opf-global-functions-redirect-domain.service.ts:L22-L42).
5. **Dominio** `global` (CTA e quick buy): `scriptReady`, `getCart`, indirizzi, delivery mode, guest user, `initiatePayment`, `verifyPayment`, `getApplePayWebSession`, ecc. (integration-libs/opf/global-functions/core/services/domains/global/opf-global-functions-global-domain-registrations.service.ts:L33-L69). Le funzioni di carrello delegano a `OpfQuickBuyTransactionService` (integration-libs/opf/global-functions/core/services/domains/global/opf-global-functions-global-domain.service.ts:L179-L266).
6. **Promise + zone.** Ogni funzione restituisce una Promise e gira in `ngZone.run`; `submit` converte l'Observable con `lastValueFrom(... last(() => true, true))` e mostra lo spinner `PLACE_ORDER_SPINNER` se ha un `ViewContainerRef` (integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L106-L170, integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L268-L291). Il `paymentSessionId` è quello passato o quello del metadata store, altrimenti eccezione `'paymentSessionId is required'` (integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L86-L104).
7. **3DS.** `handle3DSRedirect(url)` salva `opfPaymentSessionId`, `is3DSRedirect: true` e `opf3DSRedirectReturnPath` (URL completo della route result), poi naviga a `threeDsURL` (integration-libs/opf/global-functions/core/services/domains/checkout/opf-global-functions-checkout-domain.service.ts:L91-L118). Grazie alla regola di persistenza di §3.2 la sessione sopravvive al reload.

3.7 Submit hosted fields e place order

1. Lo script PSP chiama `window.Opf.payments.checkout.submit(options)` (integration-libs/opf/global-functions/core/services/opf-global-functions-shared-registrations.service.ts:L23-L32) → `OpfGlobalFunctionsSharedService.submit` → `OpfPaymentFacade.submitPayment` (integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L150-L168).
2. `OpfPaymentService` incapsula ogni metodo in un `Command`; submit e submit-complete delegano a `OpfPaymentHostedFieldsService` (integration-libs/opf/payment/core/facade/opf-payment.service.

ts:L49-L69).

3. `submitPayment` costruisce la richiesta (`channel: BROWSER`, `browserInfo`, `encryptedToken` svuotato se il metodo non è `CREDIT_CARD`), ottiene l'OTP e chiama il connector (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L62-L93); l'adapter fa POST su `payments/${paymentSessionId}/submit` con public key, access code e `Accept-Language`, retry su errori 5xx e errori normalizzati con il body copiato (integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts:L114-L140, integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts:L296-L308).
4. `paymentResponseHandler`: `ACCEPTED`/`DELAYED` → callback `onSuccess` poi `placePaymentAuthorizedOrder(true, cartId)`; `CANCELLED` con `onCancel` e `PENDING` → `EMPTY`; `REJECTED` → errore `PAYMENT_REJECTED`; altro → `STATUS_NOT_RECOGNIZED` (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L180-L227). Con l'ordine piazzato si naviga a `orderConfirmation` (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L233-L235).
5. Errori: `OpfPaymentErrorHandlerService` traduce gli errori 400 per tipo (scaduta, fondi insufficienti, carta/CVV non validi, carta smarrita) in chiavi `opfPayment.errors.*`, chiama `onFailure` e naviga a `returnPath` se presente (integration-libs/opf/payment/core/services/opf-payment-error-handler.service.ts:L36-L77). Un 401 (OTP scaduto) fa ripetere **l'intera sequenza**, OTP compreso, con `backOff({ shouldRetry: isAuthorizationError, maxTries: DEFAULT_AUTHORIZATION_ERROR_RETRIES_COUNT })` (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L115-L122, core-libs/core/src/util/occ-http-error-handlers.ts:L26-L31).

3.8 Ritorno dal PSP: la pagina di verifica

1. **Route senza path.** `OpfPaymentRootModule` registra due route figlie con `path: null` (con `@ts-ignore`) e `data.cxRoute` `paymentVerificationResult`/`paymentVerificationCancel`, entrambe su `OpfPaymentVerificationComponent`; il path reale viene dalla config di routing semantico, di default `opf/payment-verification-redirect/result|cancel` (integration-libs/opf/payment/root/opf-payment-root.module.ts:L18-L39, integration-libs/opf/payment/root/config/default-opf-payment-routing-config.ts:L9-L20).
2. **Cattura della query prima di OAuth.** Un location initializer salva `location.search` in `sessionStorage` (chiave `opfPaymentVerificationQuerySearch`) se il pathname termina con uno dei path di verifica, "before OAuth bootstrap mutates `location.search`" (integration-libs/opf/payment/root/opf-payment-root.module.ts:L42-L46, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L71-L98). `provideLocationInitializerFactory` contribuisce al multi-token `LOCATION_INITIALIZED_MULTI`, i cui initializer sono eseguiti in parallelo (core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L43-L51, core-libs/core/src/routing/routing.module.ts:L66-L71). La lettura successiva rimuove la chiave (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L169-L193).
3. **Uscita dall'iframe.** `ngOnInit` per prima cosa, se la pagina è dentro un iframe, ricarica l'URL nella finestra top (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L85-L101).
4. **Carrello in elaborazione.** Senza il toggle `opfPaymentVerificationCheckProcessingCartOnErrorOnly` chiama subito `checkIfProcessingCartIdExist()`; con il toggle solo in caso di errore (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L45-L76). Se `isPaymentInProgress === false`, va al carrello con l'errore `httpHandlers.cartNotFound` (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L266-L283).
5. `verifyResultUrl`. Accetta solo la route result (sulla cancel lancia `opfPayment.errors.cancelPayment`); il `paymentSessionId` viene, in ordine, dallo stato 3DS persistito, dal parametro `opfPaymentSessionId` del-

l'URL, dal metadata store (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L117-L167, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L195-L215).

6. **Scelta del pattern** (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L103-L137):
   - `is3DSRedirect` → `run3DSRedirectPattern`: registra il dominio `checkout`, chiama `submitCompletePayment` con i parametri dell'URL e il `returnPath` salvato, in `finalize` azzera lo stato 3DS (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L377-L406);
   - `opfAfterRedirectScriptFlag === 'true'` → `runHostedFieldsPattern`: registra il dominio `redirect`, scarica gli after-redirect scripts ed esegue con il resource loader (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L299-L341);
   - altrimenti `runHostedPagePattern`: `verifyPayment` (successo = `AUTHORIZED`/`DELAYED`) → `placePaymentAuthorizedOrder(true)` → conferma ordine (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L236-L252, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L285-L297).
7. **Errore.** Messaggio globale, cancellazione della sessione per la reiniziazione, ritorno a `opfCheckoutPaymentAndReview` (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L139-L143).

3.9 B2B checkout OPF

1. Flow `OPF` B2B a 5 step (paymentType, deliveryAddress, deliveryMode, paymentAndReview, review) (integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-config.ts:L13-L55); in B2B l'endpoint `placePaymentAuthorizedOrder` diventa `orgUsers/${userId}/orders?fields=FULL` (integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-occ-endpoints-config.ts:L10-L22).
2. `OpfB2bCheckoutRootModule` sostituisce `CheckoutB2BStepsSetGuard` con `OpfB2bCheckoutStepsSetGuard` e dichiara 4 componenti CMS lazy (integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts:L19-L48). Il guard disabilita `PAYMENT_DETAILS` per l'account payment e, solo se il provider è `OPF`, `REVIEW_ORDER` per i non-account (integration-libs/opf/b2b-checkout/root/guards/opf-b2b-checkout-steps-set.guard.ts:L70-L117).
3. Nel modulo lazy, `CheckoutPaymentTypeFacade` di `@spartacus/checkout/b2b/root` è rilegata a `OpfB2bCheckoutPaymentTypeService` (integration-libs/opf/b2b-checkout/core/facade/facade-providers.ts:L11-L17), che imposta il tipo di pagamento con `OpfPaymentFacade.setCartPaymentOption` (codice = `sapPaymentOptionId`), `CommandStrategy.CancelPrevious`, e dispatcha `CheckoutPaymentTypeSetEvent` (integration-libs/opf/b2b-checkout/core/facade/opf-b2b-checkout-payment-type.service.ts:L37-L72). L'OCC adapter fa PUT `users/${userId}/carts/${cartId}/paymentOption?fields=DEFAULT` (integration-libs/opf/payment/occ/adapters/occ-opf-payment.adapter.ts:L29-L71).
4. Il componente payment type usa l'`id` della configurazione OPF come codice payment type e abilita/disabilita gli step in base a `ACCOUNT_PAYMENT`: `handlePaymentChange` chiama `adaptCheckoutSteps`, che abilita/disabilita `PAYMENT_DETAILS` e `REVIEW_ORDER` (integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-payment-type/opf-b2b-checkout-payment-type.component.ts:L164-L179, integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-payment-type/opf-b2b-checkout-payment-type.component.ts:L110-L121). Gli altri componenti B2B estendono quelli standard (cost center, delivery ad-

dress, place order) o quello OPF B2C (integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-payment-and-review/opf-b2b-checkout-payment-and-review.component.ts:L45-L46).

3.10 CTA, quick buy, gift card, tokenisation

- **CTA.** `OpfCtaRootModule` dichiara lazy `OpfCtaScriptsComponent` e `OpfCtaQuickBuyButtons` (integration-libs/opf/cta/root/opf-cta-root.module.ts:L11-L25). `OpfCtaScriptsService` mappa il `pageId` CMS a una location (`order`, `orderConfirmationPage`, `productDetails`, `cartPage`) e chiede gli script con POST `payments/cta-scripts-rendering` (integration-libs/opf/cta/components/opf-cta-scripts/opf-cta-scripts.service.ts:L215-L254, integration-libs/opf/cta/opf-api/config/default-opf-api-cta-config.ts:L9-L17). Per le location dinamiche registra il dominio `global` e avvia eventi `CustomEvent` (`opfCartChanged`, `opfProductAmountChanged`) sulla window (integration-libs/opf/cta/components/opf-cta-scripts/opf-cta-scripts.service.ts:L70-L93, integration-libs/opf/cta/core/services/opf-dynamic-cta.service.ts:L180-L234). `OpfCtaElementComponent` toglie i tag `<script>` dall'HTML prima di renderlo (integration-libs/opf/cta/components/opf-cta-element/opf-cta-element.component.ts:L36-L50).
- **Quick buy.** `OpfQuickBuyConfig.providers` contiene `googlePay` e `applePay` con gli URL degli SDK (integration-libs/opf/quick-buy/root/config/default-opf-quick-buy-config.ts:L10-L63). I bottoni appaiono per le configurazioni `PAYMENT_GATEWAY` con `digitalWalletQuickBuy` abilitato (integration-libs/opf/quick-buy/components/opf-quick-buy-buttons/opf-quick-buy-buttons.service.ts:L37-L101). `OpfQuickBuyTransactionService` sceglie il contesto (PDP = singolo prodotto, altrimenti carrello attivo) dalla route semantica (integration-libs/opf/quick-buy/core/services/opf-quick-buy-transaction.service.ts:L36-L47, integration-libs/opf/quick-buy/core/services/opf-quick-buy-transaction.service.ts:L173-L197); sul PDP crea un nuovo carrello non attivo con il solo prodotto (integration-libs/opf/quick-buy/core/services/context/opf-quick-buy-single-product-transaction.service.ts:L307-L341). Il pagamento passa per `OpfPaymentFacade.submitPayment` con `paymentSessionId: ''` e token `btoa` (integration-libs/opf/quick-buy/components/opf-quick-buy-buttons/google-pay/google-pay.service.ts:L497-L516). La variante pickup-in-store rilega `OpfQuickBuySingleProductCartOptionsFacade` per aggiungere `pickupStore` (integration-libs/opf/quick-buy/pickup-in-store/opf-quick-buy-pickup-in-store.module.ts:L12-L22).
- **Gift card.** Il root module aggiunge `OpfGiftCardOrderOccNormalizer` a `ORDER_NORMALIZER`, `OpfGiftCardCartOccNormalizer` a `CART_NORMALIZER` e `OpfGiftCardPaymentApiInterceptor` a `HTTP_INTERCEPTORS` (integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L46-L63). L'interceptor agisce solo su `/orders/paymentAuthorizedOrderPlacement`: in errore ricarica il carrello e, se il tipo è `SAPCartStoredValueAuthorizationError`, azzera la sessione e reinizializza il pagamento (integration-libs/opf/gift-card/root/http-interceptors/opf-gift-card-payment-api.interceptor.ts:L38-L65). La gift card è una configurazione `STORED_VALUE_PAYMENT`; la copertura totale è `cart.opfGiftCardSummary.giftCardsCoverFullAmount` (integration-libs/opf/gift-card/core/facade/opf-gift-card.service.ts:L38-L62).
- **Tokenisation.** `OpfTokenisationRootModule` sostituisce `UserPaymentAdapter` di core con `OccOpfTokenisationUserPaymentAdapter` e mette tre componenti nell'outlet `OPF_CHECKOUT_BEFORE_PAYMENT_OPTIONS` (integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L33-L58). L'adapter carica solo i metodi `saved: true` e in `setDefault` fa PATCH dell'intero dettaglio (integration-libs/opf/tokenisation/root/adapters/occ-opf-tokenisation-user-payment.adapter.ts:L40-L127).

3.11 Digital Payments

1. `DigitalPaymentsModule` importa `DpCheckoutModule` (integration-libs/digital-payments/src/digital-payments.module.ts:L10-L13); l'app lo aggiunge a `CheckoutWrapperModule`, il modulo lazy del checkout, se `environment.digitalPayments` (projects/storefrontapp/src/app/spartacus/features/checkout/

checkout-wrapper.module.ts:L22-L33). DigitalPaymentsFeatureModule fornisce solo le traduzioni (projects/storefrontapp/src/app/spartacus/features/digital-payments/digital-payments-feature.module.ts:L17-L33).

2. DpCheckoutModule lega DigitalPaymentsAdapter → OccDigitalPaymentsAdapter, registra DP_DETAILS_NORMALIZER/DP_REQUEST_NORMALIZER e gli endpoint users/${userId}/carts/${cartId}/payment/digitalPayments/request|response (integration-libs/digital-payments/src/checkout/dp-checkout.module.ts:L21-L42, integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L11-L25).
3. DpPaymentMethodModule estende CheckoutPaymentMethodModule e sostituisce il componente CMS CheckoutPaymentDetails con ConfigModule.withConfig, cioè provideConfig e non un default (integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L23-L45, core-libs/core/src/config/config.module.ts:L28-L34).
4. Flusso: DpPaymentFormComponent chiede i dati di registrazione carta (POST .../request), li salva in storage con chiave digital-payment.checkout.request e reindirizza al PSP (integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-form/dp-payment-form.component.ts:L34-L57, integration-libs/digital-payments/src/checkout/facade/dp-local-storage.service.ts:L12-L46). OccDpRequestNormalizer estrae postUrl, session_id e signature (integration-libs/digital-payments/src/checkout/adapters/occ-digital-payment-request.normalizer.ts:L18-L33).
5. Al ritorno, con il query param x-card-registration-status, DpPaymentMethodComponent mostra il callback (integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.component.ts:L46-L93); se lo status è successful si chiede l'indirizzo e si chiama createPaymentDetails, che fa POST con body null e sid/sign/indirizzo come query param (integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-callback/dp-payment-callback.component.ts:L57-L131, integration-libs/digital-payments/src/checkout/adapters/occ-digital-payments.adapter.ts:L48-L84). I nomi dei parametri vengono da digitalPayments.occQueryParams (integration-libs/digital-payments/src/checkout/adapters/config/default-digital-payments-endpoint.config.ts:L9-L25).

3.12 OPPS

1. OppsRootModule importa OppsCouponCodesModule e OppsLoginRequiredModule e fornisce defaultOppsConfig (integration-libs/opps/root/opps-root.module.ts:L13-L17); default: header e parametro couponcodes, storage opps-couponcodes, parametro login loginRequired (integration-libs/opps/root/config/default-opps-config.ts:L9-L20).
2. **Coupon.** Un location initializer legge couponcodes dall'URL e lo salva in localStorage (integration-libs/opps/root/coupon-codes/opps-coupon-codes.module.ts:L12-L29, integration-libs/opps/root/coupon-codes/opps-coupon-codes.service.ts:L18-L31). OccOppsCouponCodesInterceptor (registrato con useExisting) legge il coupon nel costruttore e, nel browser, aggiunge l'header alle richieste verso la base URL OCC (integration-libs/opps/root/coupon-codes/http-interceptors/index.ts:L11-L17, integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L28-L64).
3. **Login richiesto.** OppsLoginRequiredGuard è aggiunto al multi-token BEFORE_CMS_PAGE_GUARD (root, factory []): se l'utente è anonimo e c'è ?loginRequired=true, salva l'URL di redirect e rimanda alla pagina di login (integration-libs/opps/root/login-required/opps-login-required.module.ts:L11-L20, integration-libs/opps/root/login-required/opps-login-required.guard.ts:L29-L61, core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L22-L24).

3.13 Schematic di installazione

`addOpfFeature`, `addDigitalPaymentsFeature` e `addOppsFeature` seguono la sequenza comune: `validateSpartacusInstallation`, `analyzeCrossFeatureDependencies`, poi `chain([analyzeApplication, addFeatures, addPackageJsonDependenciesForLibrary(peerDependencies), finalizeInstallation])` (integration-libs/opf/schematics/add-opf/index.ts:L25-L43, integration-libs/digital-payments/schematics/add-digital-payments/index.ts:L25-L45, integration-libs/opps/schematics/add-opps/index.ts:L25-L43). Gli spec mostrano le dipendenze tra librerie: digital-payments registra la collection di `@spartacus/checkout` prima di generare il workspace (integration-libs/digital-payments/schematics/add-digital-payments/index_spec.ts:L50-L59); OPF registra le collection Spartacus e checkout e ricrea workspace, applicazione e `ng-add` base in un `beforeEach` (integration-libs/opf/schematics/add-opf/index_spec.ts:L111-L147); OPPS installa prima la feature personalization (integration-libs/opps/schematics/add-opps/index_spec.ts:L56-L73). L'entry point primario di OPPS espone solo il root (integration-libs/opps/public_api.ts:L7-L7), mentre `integration-libs/digital-payments/assets/public_api.ts` riesporta le traduzioni (integration-libs/digital-payments/assets/public_api.ts:L7-L7).

---

4. Diagrammi

4.1 Checkout OPF: dalla lista opzioni al rendering del PSP

■ **TAVOLA 35.1** diagramma a tutta pagina, in fondo al capitolo

Fonti: §3.4 e §3.5 (integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L199-L238, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L103-L190, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L284-L331).

4.2 Ritorno dal PSP: scelta del pattern

■ **TAVOLA 35.2** diagramma a tutta pagina, in fondo al capitolo

Fonti: §3.8 (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L42-L143, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L71-L406).

4.3 Hosted fields e 3DS: lo script PSP pilota Spartacus

■ **TAVOLA 35.3** diagramma a tutta pagina, in fondo al capitolo

Fonti: §3.6 e §3.7 (integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L106-L170, integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L62-L235, integration-libs/opf/global-functions/core/services/domains/checkout/opf-global-functions-checkout-domain.service.ts:L91-L118, integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L62-L83).

5. Estratti di codice reale

Facade proxy in `root`, implementazione in `core` — integration-libs/opf/base/root/facade/opf-base.facade.ts:L16-L25 e integration-libs/opf/base/core/facade/facade-providers.ts:L11-L17:

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: OpfBaseFacade,
      feature: OPF_BASE_FEATURE,
      methods: ['getActiveConfigurationsState'],
    }),
})
export abstract class OpfBaseFacade {
```

```ts
export const facadeProviders: Provider[] = [
  OpfBaseService,
  {
    provide: OpfBaseFacade,
    useExisting: OpfBaseService,
  },
];
```

Override eager di un servizio di core — integration-libs/opf/base/root/opf-base-root.module.ts:L16-L32:

```ts
@NgModule({
  imports: [OpfEventModule],
  providers: [
    provideAppInitializer(() => {
      const opfStatePersistenceService = inject(
        OpfMetadataStatePersistanceService
      );
      opfStatePersistenceService.initSync();
    }),
    {
      provide: GlobalMessageService,
      useExisting: OpfGlobalMessageService,
    },
    provideDefaultConfig(defaultOpfConfig),
  ],
})
export class OpfBaseRootModule {}
```

Persistenza selettiva della sessione — integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L62-L83:

```ts
  protected getOpfState(): Observable<SyncedOpfState> {
    return this.opfMetadataStoreService.getOpfMetadataState().pipe(
      map((metadata: OpfMetadataModel) => {
        const {
          opfPaymentSessionId,
          opfPaymentSessionConfigurationId,
          ...persistedMetadata
        } = metadata;
        return {
          metadata: {
            ...persistedMetadata,
            // Persist session ID only for 3DS redirect — it must survive the PSP page reload.
            // For other patterns, keeping it in-memory only prevents stale sessions from being reused after a page refresh.
            ...(metadata.is3DSRedirect && {
              opfPaymentSessionId,
              opfPaymentSessionConfigurationId,
            }),
          } as OpfMetadataModel,
        };
      })
    );
  }
```

Override del guard del checkout e istanziazione dell'orchestratore — integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L42-L63:

```ts
@NgModule({
  providers: [
    {
      provide: CheckoutAuthGuard,
      useClass: OpfCheckoutAuthGuard,
    },
    {
      provide: OpfCheckoutAdapter,
      useClass: OpfApiCheckoutAdapter,
    },
    OpfCheckoutConnector,
    OpfCartUserEmailCheckerService,
    provideDefaultConfig(defaultOpfCheckoutRoutingConfig),
    provideDefaultConfig(defaultOccOpfCheckoutConfig),
    provideDefaultConfig(defaultOpfCheckoutConfig),
    provideDefaultConfigFactory(defaultOpfCheckoutComponentsConfig),
    provideAppInitializer(() => {
      inject(CheckoutFlowOrchestratorService);
    }),
```

```
  ],
})
export class OpfCheckoutRootModule {}
```

Scelta del flow per payment provider (lib checkout) — feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L35-L44:

```ts
getCheckoutFlow(): CheckoutFlow | undefined {
  if (this.paymentProviderName) {
    const flow =
      this.checkoutConfig.checkout?.flows?.[this.paymentProviderName];
    if (flow) {
      return flow;
    }
  }
  return this.checkoutConfig.checkout;
}
```

URL OPF — integration-libs/opf/base/core/services/opf-endpoints.service.ts:L40-L48 e :L74-L74:

```ts
buildUrl(endpoint: string, attributes?: DynamicAttributes): string {
  const baseUrl = this.getBaseEndpoint();
  let opfEndpoint = this.getEndpointFromContext(endpoint);
  if (attributes) {
    const { urlParams, queryParams } = attributes;

    if (urlParams && opfEndpoint) {
      opfEndpoint = StringTemplate.resolve(opfEndpoint, urlParams, true);
    }
    // ...
  return `${baseUrl}/${this._activeBaseSite}/${opfEndpoint}`;
```

Gestione dello status di submit — integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L190-L207:

```ts
  if (
    response.status === OpfPaymentSubmitStatus.ACCEPTED ||
    response.status === OpfPaymentSubmitStatus.DELAYED
  ) {
    return from(Promise.resolve(callbacks.onSuccess(response))).pipe(
      concatMap(() => this.placeOrder(cartId))
    );
  } else if (
    callbacks?.onCancel &&
    response.status === OpfPaymentSubmitStatus.CANCELLED
  ) {
    return from(Promise.resolve(callbacks.onCancel(response))).pipe(
```

```
      concatMap(() => EMPTY)
    );
  } else if (response.status === OpfPaymentSubmitStatus.PENDING) {
    return from(Promise.resolve(callbacks.onPending(response))).pipe(
      concatMap(() => EMPTY)
    );
```

Il contenitore globale per gli script del PSP — integration-libs/opf/global-functions/core/facade/opf-global-functions.service.ts:L73-L88:

```ts
protected getContainer(
  domain: OpfGlobalFunctionsDomain
): OpfPaymentGlobalMethods {
  const window = this.winRef.nativeWindow as any;
  if (!window) {
    return {};
  }

  if (!window.Opf?.payments?.[domain]) {
    window.Opf = window?.Opf ?? {};
    window.Opf.payments = window.Opf.payments ?? {};
    window.Opf.payments[domain] = {};
  }

  return window.Opf.payments[domain];
}
```

Route di verifica senza path e location initializer — integration-libs/opf/payment/root/opf-payment-root.module.ts:L21-L46:

```ts
  RouterModule.forChild([
    {
      // @ts-ignore
      path: null,
      component: OpfPaymentVerificationComponent,
      data: {
        cxRoute: 'paymentVerificationResult',
      },
    },
    // ...
  ]),
],
providers: [
  provideDefaultConfig(defaultOpfPaymentRoutingConfig),
  provideLocationInitializerFactory(() => {
    // convert incorrect type `() => void` to `() => Promise<void>`
    const locationInitializer = captureOpfPaymentVerificationQueryFactory();
```

```
      return () => Promise.resolve(locationInitializer());
    }),
  ],
```

Esecuzione di script PSP fuori dalla zone — integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L237-L259:

```ts
  this.ngZone.runOutsideAngular(() => {
    if (this.windowRef.isBrowser() && this.windowRef.nativeWindow) {
      const nativeWindow = this.windowRef.nativeWindow as any;
      if (nativeWindow.OpfContext) {
        delete nativeWindow.OpfContext;
      }

      nativeWindow.OpfContext = contextData ?? {};
      nativeWindow.OpfContext.additionalData ??= {};
    }

    const scriptElement = this.document.createElement('script');
    scriptElement.type = 'text/javascript';
    scriptElement.textContent = originalScript;
    scriptElement.setAttribute('data-opf-script', 'true');
    this.document.head.appendChild(scriptElement);

    // Clean up after execution
    setTimeout(() => {
      if (scriptElement.parentNode) {
        scriptElement.parentNode.removeChild(scriptElement);
      }
    }, 100);
```

Multi-provider della gift card — integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L46-L61:

```ts
providers: [
  {
    provide: ORDER_NORMALIZER,
    useExisting: OpfGiftCardOrderOccNormalizer,
    multi: true,
  },
  {
    provide: CART_NORMALIZER,
    useExisting: OpfGiftCardCartOccNormalizer,
    multi: true,
  },
  {
    provide: HTTP_INTERCEPTORS,
```

```
      useClass: OpfGiftCardPaymentApiInterceptor,
      multi: true,
    },
```

Digital Payments: override CMS con `withConfig` — integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L32-L45:

```ts
    ConfigModule.withConfig({
      cmsComponents: {
        CheckoutPaymentDetails: {
          component: DpPaymentMethodComponent,
        },
      },
    }),
    FeaturesConfigModule,
    DpPaymentMethodComponent,
  ],
  exports: [DpPaymentMethodComponent],
  providers: [provideDefaultConfig(defaultDigitalPaymentsConfig)],
})
export class DpPaymentMethodModule extends CorePaymentMethodModule {}
```

OPPS: interceptor del coupon — integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L28-L38 e :L51-L63:

```ts
  constructor() {
    this.initialize();
  }

  protected initialize() {
    this.oppsCoupon = this.service.getCouponCodes();
    if (this.winRef.isBrowser()) {
      this.requestHeader =
        this.config.opps?.couponcodes?.httpHeaderName?.toLowerCase?.();
    }
  }
  // ...
    if (
      this.winRef.isBrowser() &&
      this.requestHeader &&
      this.oppsCoupon &&
      request.url.includes(this.occEndpoints.getBaseUrl())
    ) {
      request = request.clone({
        setHeaders: {
          [this.requestHeader]: this.oppsCoupon,
```

```
        },
      });
    }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
opf.opfBaseUrl	string	''	integration-libs/opf/base/root/config/default-opf-config.ts:L11-L11	integration-libs/opf/base/core/services/opf-endpoints.service.ts:L90-L96
opf.commerceCloudPublicKey	string	—	integration-libs/opf/base/root/config/opf-config.ts:L17-L17	integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts:L89-L92, integration-libs/opf/base/opf-api/adapters/opf-api-base.adapter.ts:L90-L94
opf.paymentOption.enableInfoMessage	boolean	true	integration-libs/opf/base/root/config/default-opf-config.ts:L12-L14	integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L255-L260
opf.paymentOption.paymentInfoMessagesMap	Record<number, string>	—	integration-libs/opf/base/root/config/opf-config.ts:L32-L32	integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L244-L253

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
opf.paymentOption.iframeSandboxMap	Record<number, string>	—	integration-libs/opf/base/root/config/opf-config.ts:L54-L54	integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.ts:L111-L117
opf.localPspResources	Record<number, {jsFiles, cssFiles}>	—	integration-libs/opf/base/root/config/opf-config.ts:L76-L82	integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L266-L345
opf.enableGetActiveConfigurationsAccessCodeHeader	boolean	— (app: true)	integration-libs/opf/base/root/config/opf-config.ts:L88-L88	integration-libs/opf/base/opf-api/adapters/opf-api-base.adapter.ts:L96-L100
backend.opfApi.endpoints.*	string	per sotto-feature (es. active-configurations, payments/${paymentSessionId}/submit)	integration-libs/opf/base/opf-api/config/default-opf-api-base-config.ts:L9-L17, integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L9-L23	integration-libs/opf/base/core/services/opf-endpoints.service.ts:L77-L88
backend.occ.endpoints.setCartPaymentOption	string	users/${userId}/carts/${cartId}/paymentOption	integration-libs/opf/payment/occ/config/default-occ-opf-payment-config.ts:L13-L13	integration-libs/opf/payment/occ/adapters/occ-opf-payment.adapter.ts:L58-L71
backend.occ.endpoints.cartUserEmail	string	...?fields=sapCustomerEmail	integration-libs/opf/checkout/root/config/default-occ-opf-checkout-config.ts:L13-L14	integration-libs/opf/checkout/root/adapters/opf-api-checkout.adapter.ts:L42-L44

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.endpoints.cart (OPF)	string	campi estesi + sapBillingAddress(FULL)	integration-libs/opf/global-functions/root/config/default-occ-opf-cart-config.ts:L10-L12	adapter OCC del carrello (25)
backend.occ.endpoints.placePaymentAuthorizedOrder	string	B2B orgUsers/${userId}/orders?fields=FULL; gift card con campi propri	integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-occ-endpoints-config.ts:L10-L12, integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-order-config.ts:L13-L14	adapter OCC ordini (27)
backend.occ.endpoints.applyGiftCard / removeGiftCard	string	.../giftCards[/${giftCardId}]	integration-libs/opf/gift-card/occ/config/default-occ-opf-gift-card-config.ts:L13-L15	integration-libs/opf/gift-card/occ/adapters/occ-opf-gift-card.adapter.ts:L68-L82
backend.occ.endpoints.quickBuy*	string	5 endpoint	integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts:L13-L20	integration-libs/opf/quick-buy/occ/adapters/occ-opf-quick-buy-cart.adapter.ts:L143-L190
checkout.flows.OPF	CheckoutFlow	B2C 3 step / B2B 5 step	integration-libs/opf/checkout/root/config/default-opf-checkout-config.ts:L36-L45, integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-config.ts:L46-L55	feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L35-L44

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes.paymentVerificationResult / Cancel	RoutingConfig	opf/payment-verification-redirect/result / cancel	integration-libs/opf/payment/root/config/default-opf-payment-routing-config.ts:L9-L20	integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L80-L85
routing.routes.opfCheckout*	RoutingConfig	5 route	integration-libs/opf/checkout/root/config/default-opf-checkout-routing-config.ts:L9-L29	route semantiche
providers.googlePay / providers.applePay	config provider	URL SDK Google (TEST) e Apple	integration-libs/opf/quick-buy/root/config/default-opf-quick-buy-config.ts:L10-L63	integration-libs/opf/quick-buy/components/opf-quick-buy-buttons/google-pay/google-pay.service.ts:L134-L148
cart.validation / selectiveCart / showRealTimeStockInPDP (gift card)	CartConfig	true / false / false	integration-libs/opf/gift-card/root/config/default-gift-card-cart-config.ts:L9-L21	lib cart

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
features.opfUseDestroyRef, features.opfPaymentVerificationCheckProcessingCartOnErrorOnly, features.opfCheckoutUseUpdatePaymentTransaction	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1074-L1083	integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.ts:L147-L161, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L45-L50, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L106-L107
digitalPayments.occQueryParams.*	string	sid, sign, campi indirizzo	integration-libs/digital-payments/src/checkout/adapters/config/default-digital-payments-endpoint.config.ts:L9-L25	integration-libs/digital-payments/src/checkout/adapters/occ-digital-payments.adapter.ts:L64-L84
backend.occ.endpoints.paymentRequest / paymentDetails	string	.../digitalPayments/request / response	integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L11-L16	integration-libs/digital-payments/src/checkout/adapters/occ-digital-payments.adapter.ts:L40-L58

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
opps.couponcodes.{httpHeaderName, urlParameter, localStorageKey}	string	couponcodes / couponcodes / opps-couponcodes	integration-libs/opps/root/config/default-opps-config.ts:L11-L15	integration-libs/opps/root/coupon-codes/opps-coupon-codes.service.ts:L15-L31, integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L32-L38
opps.loginRequired.urlParameter	string	loginRequired	integration-libs/opps/root/config/default-opps-config.ts:L16-L18	integration-libs/opps/root/login-required/opps-login-required.guard.ts:L38-L48

Note: la chiave dei toggle va in `features` della config (vedi 42); l'app li imposta a `true` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L364-L371).

---

7. Commenti e note notevoli dal sorgente

- **"SSR mode not supported for security concerns"** — ripetuto in `executeScriptFromHtml` e `loadResources` (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L175-L175, integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L403-L403). Il codice del PSP non gira mai lato server.
- **Persistenza della sessione solo per 3DS**: "Persist session ID only for 3DS redirect — it must survive the PSP page reload. For other patterns, keeping it in-memory only prevents stale sessions from being reused after a page refresh." (integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L73-L74).
- **"Stores the redirect query string before OAuth bootstrap mutates `location.search`."** — motivo del location initializer di payment (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L71-L73).
- **Location initializer**: "convert incorrect type `() => void` to `() => Promise<void>`", identico in OPF e OPPS (integration-libs/opf/payment/root/opf-payment-root.module.ts:L43-L43, integration-libs/opps/root/coupon-codes/opps-coupon-codes.module.ts:L23-L23).
- **Saferpay e iframe**: "Some PSPs render verification/success pages inside an iframe; in such cases we reload the current URL in the top window to preserve full-page UX." con link alla documentazione Saferpay (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L85-L91).
- **Retry su OTP**: "It means that `accessCode` (OTP signature) is not valid or expired and we need to refresh it." (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L116-L119).
- `cartId` **deprecato**: "@deprecated Property no longer used. Cart ID is currently derived from the OTP access code header." (integration-libs/opf/global-functions/core/services/opf-global-functions-

shared.service.ts:L46-L46; anche integration-libs/opf/payment/root/model/opf-payment.model.ts:L26-L27).

- **Precedenza** `dynamicScript`: "if `dynamicScript` and `destination` are present in same config, dynamicScript takes precendence." (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L275-L283).
- **JSDoc contraddittorio** di `loadResources`: "The returned Promise is also resolved (not rejected!) immediately when any loading error occurs." (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L395-L395) — vedi §9.
- **TODO tokenisation**: "TO DO: Unify this adapter with Core once Core fixes the paymentDetail PATCH request payload handling." (integration-libs/opf/tokenisation/root/adapters/occ-opf-tokenisation-user-payment.adapter.ts:L67-L67); e i segnaposto "Currently unused as all payment operations are delegated to the core UserPaymentService" (integration-libs/opf/tokenisation/core/connectors/opf-tokenisation.adapter.ts:L9-L16).
- **Workaround CTA**: "Removing script tags on FE until BE fix: CXSPA-8572" (integration-libs/opf/cta/components/opf-cta-element/opf-cta-element.component.ts:L43-L43).
- **Commento sul carrello quick buy**: "`createCart` initially replays the previously created (NEW_CREATED) cart from the store. Skip it so we operate on the freshly created..." (integration-libs/opf/quick-buy/core/services/context/opf-quick-buy-single-product-transaction.service.ts:L307-L341).
- **Deprecazione in B2C**: `@deprecated since 221121.21. No longer used internally.` su `opfBaseFacade` di `OpfCheckoutPaymentAndReviewComponent` (integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L82-L86).
- **Digital Payments**: `// TODO:#checkout - handle breaking changes` (integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.component.ts:L70-L70).
- `BEFORE_CMS_PAGE_GUARD`: "**CAUTION**: Suitable for global guards ONLY!" (core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L10-L21) — OPPS lo usa correttamente per un controllo globale.

8. Extension point

8.1 Override via DI fatti da OPF (e da imitare)

TOKEN SOSTITUITO	CON	TIPO PROVIDER	SCOPE	DOVE
`GlobalMessageService` (core)	`OpfGlobalMessageService`	`useExisting`	root (eager)	integration-libs/opf/base/root/opf-base-root.module.ts:L25-L28
`CheckoutAuthGuard` (checkout)	`OpfCheckoutAuthGuard`	`useClass`	root (eager)	integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L44-L47
`CheckoutB2BStepsSetGuard` (checkout b2b)	`OpfB2bCheckoutStepsSetGuard`	`useClass`	root (eager)	integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts:L39-L42

TOKEN SOSTITUITO	CON	TIPO PROVIDER	SCOPE	DOVE
`CheckoutPaymentTypeFacade` (checkout b2b)	`OpfB2bCheckoutPaymentTypeService`	`useExisting`	modulo lazy OPF B2B	`integration-libs/opf/b2b-checkout/core/facade/facade-providers.ts:L11-L17`
`UserPaymentAdapter` (core)	`OccOpfTokenisationUserPaymentAdapter`	`useClass`	root (eager)	`integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L37-L40`
`ORDER_NORMALIZER`, `CART_NORMALIZER`, `HTTP_INTERCEPTORS`	normalizer/interceptor gift card	`multi: true`	root (eager)	`integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L46-L61`
`ORDER_NORMALIZER`	`OpfOccOrderNormalizer`	`multi: true`	chunk order	`integration-libs/opf/order/opf-order.module.ts:L11-L20`
`OpfQuickBuySingleProductCartOptionsFacade`	servizio pickup-in-store	provider	modulo lazy quick buy	`integration-libs/opf/quick-buy/pickup-in-store/opf-quick-buy-pickup-in-store.module.ts:L12-L22`
`cmsComponents.CheckoutPaymentDetails` (DP)	`DpPaymentMethodComponent`	`provideConfig`	chunk checkout	`integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L32-L38`
`HTTP_INTERCEPTORS`, `BEFORE_CMS_PAGE_GUARD` (OPPS)	interceptor coupon, guard login	`multi: true`	root (eager)	`integration-libs/opps/root/coupon-codes/http-interceptors/index.ts:L11-L17`, `integration-libs/opps/root/login-required/opps-login-required.module.ts:L11-L20`

Regola pratica che emerge: OPF mette nel **root module** gli override di guard, adapter di core e interceptor, che così valgono nell'injector root per tutta l'app indipendentemente da quali chunk sono stati caricati; nel **modulo lazy** mette solo gli override consumati dentro lo stesso chunk (con il rischio di scope descritto in §9.17). Per ricette generiche di override di servizi, facade e guard: 50 — Ricette: configurazione, DI, interceptor, eventi.

8.2 Punti di estensione esportati da OPF

PUNTO	TIPO	DOVE	ENTRY POINT
`OpfConfig`, `OpfApiConfig`, `OpfApiEndpoints`	config / augmentation	integration-libs/opf/base/root/config/opf-config.ts:L14-L14, integration-libs/opf/base/root/config/opf-api-config.ts:L15-L17, integration-libs/opf/base/root/model/opf-api-endpoints.model.ts:L7-L7	`@spartacus/opf/base/root` (integration-libs/opf/base/root/public_api.ts:L7-L13)
`OpfBaseFacade`, `OpfBaseAdapter`, `OPF_ACTIVE_CONFIGURATIONS_NORMALIZER`, `OpfEndpointsService`	facade / adapter / token	integration-libs/opf/base/core/connectors/opf-base.adapter.ts:L13-L13, integration-libs/opf/base/core/tokens/tokens.ts:L11-L13	`@spartacus/opf/base/root`, `@spartacus/opf/base/core`
`OpfPaymentFacade`, `OpfPaymentAdapter`, normalizer `OPF_PAYMENT_*`, `OPF_PAYMENT_CONFIG_SERIALIZER`	facade / adapter / token	integration-libs/opf/payment/core/connectors/opf-payment.adapter.ts:L20-L20, integration-libs/opf/payment/core/tokens/tokens.ts:L16-L30, integration-libs/opf/payment/core/connectors/converters.ts:L11-L13	`@spartacus/opf/payment/root` (integration-libs/opf/payment/root/public_api.ts:L7-L13), `/core`
`OpfCheckoutOutlets` (3 outlet) e `CartOutlets.OPF_CHECKOUT_PICKUP_ITEMS`	outlet	integration-libs/opf/checkout/root/model/opf-checkout-outlets.model.ts:L7-L11, integration-libs/opf/checkout/root/model/augmented-cart-outlets.model.ts:L9-L16	`@spartacus/opf/checkout/root`
`OpfCtaFacade`, `OpfCtaAdapter`, `OPF_CTA_SCRIPTS_NORMALIZER`	facade / adapter / token	integration-libs/opf/cta/root/facade/opf-cta.facade.ts:L26-L26, integration-libs/opf/cta/core/connectors/opf-cta.adapter.ts:L13-L13	`@spartacus/opf/cta/root`, `/core`
`OpfGlobalFunctionsFacade`, `window.Opf.payments[domain]`	facade / API JS	integration-libs/opf/global-functions/root/facade/opf-global-functions.facade.ts:L24-L24	`@spartacus/opf/global-functions/root` (integration-libs/opf/global-functions/root/public_api.ts:L7-L11)
`OpfQuickBuyConfig`, `OpfQuickBuySingleProductCartOptionsFacade`, `OpfQuickBuyCartAdapter`	config / facade / adapter	integration-libs/opf/quick-buy/root/config/opf-quick-buy-config.ts:L18-L18, integration-libs/opf/quick-buy/root/facade/opf-quick-buy-single-product-cart-options.facade.ts:L22-L22, integration-libs/opf/quick-buy/core/connectors/opf-quick-buy-cart.adapter.ts:L11-L11	`@spartacus/opf/quick-buy/root`, `/core`, `/pickup-in-store`
`LAUNCH_CALLER.OPF_ERROR`	enum augmentation + assegnazione runtime	integration-libs/opf/base/root/model/augmented-core.model.ts:L9-L15	`@spartacus/opf/base/root`

PUNTO	TIPO	DOVE	ENTRY POINT
`DigitalPaymentsModule`, `DpCheckoutModule`, `DpCheckoutPaymentService`	modulo / facade	`integration-libs/digital-payments/public_api.ts:L10-L11`, `integration-libs/digital-payments/src/checkout/index.ts:L7-L9`, `integration-libs/digital-payments/src/checkout/facade/index.ts:L7-L7`	`@spartacus/digital-payments` (adapter, token e config **non** sono esportati da quel barrel)
`OppsConfig`, `OccOppsCouponCodesInterceptor`, `OppsLoginRequiredGuard`	config / interceptor / guard	`integration-libs/opps/root/public_api.ts:L7-L10`	`@spartacus/opps/root`

8.3 Come estendere (con le ricette di Parte III)

- **Aggiungere un endpoint OPF API o cambiarne il path**: estendere `OpfApiEndpoints` per augmentation (come `integration-libs/opf/base/opf-api/model/opf-api-base-endpoints.model.ts:L9-L16`) e fornire `backend.opfApi.endpoints` con `provideConfig`; costruire l'URL con `OpfEndpointsService.buildUrl`. Il pattern è quello di OCC: 52 — Ricette: OCC, adapter, converter, endpoint.
- **Trasformare le risposte OPF**: i token `OPF_*_NORMALIZER` non hanno converter di default, quindi un normalizer aggiunto con `multi: true` è l'unico a operare (`core-libs/core/src/util/converter.service.ts:L106-L111`). Vedi 52.
- **Sostituire un componente OPF** (es. `OpfCheckoutPaymentAndReview`) o aggiungere contenuto negli outlet `OpfCheckoutOutlets`: 51 — Ricette: componenti CMS, outlet, route e layout. Esempio reale: la tokenisation usa `provideOutlet` su `OPF_CHECKOUT_BEFORE_PAYMENT_OPTIONS` (`integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L41-L55`).
- **Flow di checkout per payment provider**: definire `checkout.flows[<paymentProvider>]` (con l'intera lista di step, perché gli array si sostituiscono): 53 — Ricette: feature lib lazy, carrello/checkout.
- **Override di facade/guard/adapter**: 50 — Ricette: configurazione, DI, interceptor, eventi.
- **Importare solo dagli entry point pubblici** (`@spartacus/opf/<feature>/root|core|...`) e non da path interni: 54 — API pubblica vs interna, upgrade.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `GlobalMessageService` **cambia per tutta l'app.** Importare `OpfBaseRootModule` sostituisce il servizio globale con la sottoclasse OPF, che può silenziare chiavi per 2 s (`integration-libs/opf/base/root/opf-base-root.module.ts:L25-L28`, `integration-libs/opf/base/root/services/opf-global-message.service.ts:L51-L60`). Un vostro override di `GlobalMessageService` entra in conflitto con questo.
2. **Due default per** `checkout.flows.OPF`. B2C e B2B root module forniscono entrambi il flow come default; con entrambi importati (come nell'app) bisogna scegliere esplicitamente con `provideConfig` (`integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L56-L56`, `integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts:L43-L43`, `projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L73-L81`).
3. **Il flow dipende dal backend.** Il flow `OPF` si attiva solo se `baseStore.paymentProvider === 'OPF'`, letto una volta con `take(1)` (`feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L20-L44`). `BaseSiteService.get()` non emette prima del caricamento dei base site: passa da `getActive()`, che filtra i valori falsy, e da `getAll()`, che filtra `null` e lancia `LoadBaseSites` (`core-libs/core/src/site-context/facade/base-site.service.ts:L31-L35`, `core-libs/core/src/site-context/`

facade/base-site.service.ts:L41-L50, core-libs/core/src/site-context/facade/base-site.service.ts:L63-L69). Il nome resta `undefined` solo se il base site caricato non contiene `baseStore.paymentProvider`.

4. **URL con `undefined`.** `_activeBaseSite` è riempito in modo asincrono: un `buildUrl` chiamato prima della prima emissione produce `.../undefined/...` (integration-libs/opf/base/core/services/opf-endpoints.service.ts:L26-L38, integration-libs/opf/base/core/services/opf-endpoints.service.ts:L74-L74).
5. **Endpoint come oggetto.** `getEndpointFromContext` restituisce il valore di config così com'è e `buildUrl` lo tratta come stringa (`opfEndpoint.includes('?')`): un `OpfApiEndpoint` oggetto, ammesso dal tipo, non viene risolto (integration-libs/opf/base/core/services/opf-endpoints.service.ts:L53-L53, integration-libs/opf/base/core/services/opf-endpoints.service.ts:L77-L88, integration-libs/opf/base/root/model/opf-api-endpoint.model.ts:L7-L10).
6. **Header di lingua congelati.** `OpfApiBaseAdapter` e `OpfApiCtaAdapter` calcolano `Accept-Language` una volta, come campi; `OpfApiQuickBuyAdapter` fa lo stesso con `Content-Language`; `OpfApiPaymentAdapter` usa getter (integration-libs/opf/base/opf-api/adapters/opf-api-base.adapter.ts:L53-L63, integration-libs/opf/cta/opf-api/adapters/opf-api-cta.adapter.ts:L50-L54, integration-libs/opf/quick-buy/opf-api/adapters/opf-api-quick-buy.adapter.ts:L49-L53, integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts:L62-L83).
7. **Nessuna cache delle active configurations.** Ogni chiamata a `getActiveConfigurationsState` crea una nuova Query (integration-libs/opf/base/core/facade/opf-base.service.ts:L23-L34); inoltre `OpfCheckoutPaymentsComponent` filtra **in place** `state.data.value` (integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L219-L228).
8. `CANCELLED` **senza** `onCancel` **diventa "status non riconosciuto".** Il ramo cancel richiede `callbacks?.onCancel`; senza, uno status `CANCELLED` cade nel ramo finale con `STATUS_NOT_RECOGNIZED` (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L197-L226). Le global functions `submit`/`submitComplete` forniscono un `submitCancel` di default (integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L121-L148), ma il ritorno 3DS e Google Pay quick buy chiamano la facade senza `onCancel`, quindi per loro un `CANCELLED` diventa `STATUS_NOT_RECOGNIZED` (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L390-L400, integration-libs/opf/quick-buy/components/opf-quick-buy-buttons/google-pay/google-pay.service.ts:L34-L38, integration-libs/opf/quick-buy/components/opf-quick-buy-buttons/google-pay/google-pay.service.ts:L506-L513). Lo stesso vale per chi chiama `OpfPaymentFacade.submitPayment` direttamente.
9. **Il retry su 401 del wrapper non scatta.** In `initiatePayment` il `catchError(handlePaymentInitiationError)` precede il `backOff`; per un 401 restituisce `throwError(() => 'Payment failed')`, una stringa, mentre `isAuthorizationError` controlla `err.status === 401` (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L179-L188, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L333-L343, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L379-L388, core-libs/core/src/util/occ-http-error-handlers.ts:L26-L28). Nel submit hosted fields invece l'errore originale viene rilanciato, il retry funziona, ma `handlePaymentError` viene invocato a ogni tentativo (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L107-L122).
10. **409 non gestito se il place order fallisce.** In `handlePaymentAlreadyDoneError`, se il place order fallisce `catchError` restituisce `of()`, che completa senza emettere: lo `switchMap` successivo non parte (integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L345-L359).
11. **Verifica che parte "carrello non trovato".** Senza il toggle, `checkIfProcessingCartIdExist` è chiamato subito: se `isPaymentInProgress === false` (ad esempio perché il metadata è stato azzerato) si va al carrello

con errore anche se la verifica riuscirebbe (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L45-L50, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L266-L283).

12. **La sessione non sopravvive al refresh** (salvo 3DS): è un comportamento voluto (§7), ma significa che con `opfCheckoutUseUpdatePaymentTransaction` dopo un reload si torna a `initiatePayment` (integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L62-L83, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L239-L255).
13. **CSP.** `executeScriptFromHtml` usa `Function()` (equivalente a eval) e i template usano `bypassSecurityTrustHtml`/`bypassSecurityTrustResourceUrl`: una Content Security Policy restrittiva rompe i PSP (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L182-L182, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.ts:L103-L109).
14. `loadResources` **rifiuta, nonostante il JSDoc.** Il commento dice che la Promise è risolta anche in errore, ma `loadScript`/`loadStyles` chiamano `reject()` e `Promise.all` rifiuta; i chiamanti infatti hanno `.catch` (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L395-L395, integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L141-L142, integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L429-L435).
15. **Risorse locali e CSS** `SEPARATE`**.** Con `localPspResources` configurato per quell'opzione, il CSS `SEPARATE` aggiunto a `styles` viene ignorato perché le risorse locali sostituiscono quelle esterne (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L408-L416).
16. `OPF_ERROR` **configurato due volte**, `inline` nel modulo error modal e `inlineRoot` nel core delle global functions, entrambi come default (integration-libs/opf/base/components/opf-error-modal/default-opf-error-modal.layout.config.ts:L10-L18, integration-libs/opf/global-functions/core/config/default-opf-global-layout.config.ts:L14-L27). DA VERIFICARE quale prevale: dipende dall'ordine con cui i due moduli lazy vengono caricati e i default fusi.
17. **Override con scope lazy.** `CheckoutPaymentTypeFacade` è rilegata solo nel modulo lazy OPF B2B, mentre la facade standard è una `facadeFactory` di `CHECKOUT_CORE_FEATURE`; consumatori fuori da quell'injector (es. il guard root `OpfB2bCheckoutStepsSetGuard`) potrebbero ricevere l'implementazione standard (integration-libs/opf/b2b-checkout/core/facade/facade-providers.ts:L11-L17, feature-libs/checkout/b2b/root/facade/checkout-payment-type.facade.ts:L17-L18, integration-libs/opf/b2b-checkout/root/guards/opf-b2b-checkout-steps-set.guard.ts:L39-L56). DA VERIFICARE a runtime (risoluzione proxy vs injector lazy). Inoltre `getPaymentTypesState()` restituisce `of()` e `getPaymentTypes()` non emette mai (integration-libs/opf/b2b-checkout/core/facade/opf-b2b-checkout-payment-type.service.ts:L103-L109).
18. **Il B2C OPF dipende dal pacchetto checkout B2B**: `OpfCheckoutPaymentAndReviewComponent` inietta `CheckoutPaymentTypeFacade` di `@spartacus/checkout/b2b/root` (integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L23-L23, integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L81-L81).
19. **T&C esplicito per sottostringa**: `JSON.stringify(page).includes(uid)` attiva il T&C con qualunque occorrenza della stringa nella pagina (integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L129-L131).
20. **Moduli lazy che reimportano il root.** `OpfGiftCardModule` e `OpfTokenisationModule` reimportano i rispettivi root module: interceptor, normalizer e override vengono registrati anche nell'injector lazy (integration-libs/opf/gift-card/opf-gift-card.module.ts:L13-L21, integration-libs/opf/tokenisation/opf-tokenisation.module.ts:L11-L18). DA VERIFICARE l'effetto della doppia registrazione degli `HTTP_INTERCEPTORS` (l'`HttpClient` root dovrebbe usare quelli root, non confermato a runtime).

21. **Interceptor gift card su ogni errore di place order**: ricarica il carrello per qualunque errore di `paymentAuthorizedOrderPlacement`, non solo gift card, ed è registrato eager anche quando la gift card non è configurata (integration-libs/opf/gift-card/root/http-interceptors/opf-gift-card-payment-api.interceptor.ts:L43-L50, projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L85-L103).
22. **Doppio slash corretto a posteriori**: con `paymentSessionId` vuoto (quick buy) il path diventa `payments//submit`, poi corretto con `.replace('//submit', '/submit')` (integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts:L264-L270).
23. **Stato condiviso nel quick buy.** `OpfQuickBuyTransactionService` memorizza `transactionContext` e non lo azzera (integration-libs/opf/quick-buy/core/services/opf-quick-buy-transaction.service.ts:L34-L47, integration-libs/opf/quick-buy/core/services/opf-quick-buy-transaction.service.ts:L173-L179); `OpfDynamicCtaService` è condiviso tra CTA scripts e CTA quick buy nello stesso injector lazy: è `@Injectable()` senza `providedIn` ed è fornito da due NgModule importati entrambi da `OpfCtaComponentsModule`, quindi finisce in un unico module injector (integration-libs/opf/cta/core/services/opf-dynamic-cta.service.ts:L46-L47, integration-libs/opf/cta/components/opf-cta-scripts/opf-cta-scripts.module.ts:L20-L22, integration-libs/opf/cta/components/opf-cta-quick-buy-buttons/opf-cta-quick-buy-buttons.module.ts:L19-L21, integration-libs/opf/cta/components/opf-cta-components.module.ts:L13-L16); lo stato interno condiviso è in integration-libs/opf/cta/core/services/opf-dynamic-cta.service.ts:L57-L60.
24. **Digital Payments richiede il chunk del checkout.** `DpPaymentMethodComponent` inietta le classi concrete `CheckoutDeliveryAddressService`/`CheckoutPaymentService` di `checkout/base/core`, non le facade: funziona solo nello stesso injector lazy del checkout core, come in `CheckoutWrapperModule` (integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.component.ts:L15-L18, integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.component.ts:L71-L80). E `defaultDigitalPaymentsConfig` è fornito dal modulo CMS, non da `DpCheckoutModule`: senza di esso `sid`/`sign` non vengono aggiunti (integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L43-L43, integration-libs/digital-payments/src/checkout/adapters/occ-digital-payments.adapter.ts:L86-L95).
25. `getCardRegistrationDetails` **lancia dentro** `map` se la query è in errore, rompendo lo stream (integration-libs/digital-payments/src/checkout/facade/dp-checkout-payment.service.ts:L44-L54).
26. **OPPS legge il coupon una sola volta.** L'interceptor lo legge nel costruttore: un coupon salvato dopo l'istanziazione non viene inviato fino a nuova istanza (integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L28-L38). `DA VERIFICARE` l'ordine reale tra il location initializer e la prima istanza dell'interceptor (richiede esecuzione). Inoltre, con un URL senza `?`, `url.substring(url.indexOf('?'))` passa l'URL intero a `URLSearchParams` e il coupon risulta `null` (integration-libs/opps/root/coupon-codes/opps-coupon-codes.service.ts:L18-L26).

---

10. Come fare debug

- **La feature OPF non si carica / "Feature ... is not configured properly".** Il proxy `facadeFactory` richiede che la feature sia in `featureModules` (core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L10-L23): controllare la mappa nell'app (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L105-L152) e il nome della feature nella facade (es. `OPF_BASE_FEATURE`, integration-libs/opf/base/root/facade/opf-base.facade.ts:L21-L21).

- **Il checkout non usa gli step OPF.** Controllare `baseSite.baseStore.paymentProvider` (deve essere `OPF`) e che `checkout.flows.OPF` sia quello atteso (B2C/B2B) (feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L28-L44, projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L73-L81).
- **Network.** Le chiamate OPF vanno a `${opfBaseUrl}/${baseSite}/...` (§3.2); verificare gli header `sap-commerce-cloud-public-key` e `sap-commerce-cloud-access-code` (integration-libs/opf/base/root/config/constants.ts:L7-L8). Un `.../undefined/...` nell'URL indica il problema di §9.4. Un 401 su submit indica OTP scaduto (retry automatico, §3.7).
- **Stato OPF.** Non c'è NgRx: in DevTools ispezionare in `localStorage` la chiave `spartacus⚿⚿opf` (default di `StatePersistenceService`: `storageType = LOCAL_STORAGE`, `context$ = of('')`, prefisso `spartacus⚿<context>⚿`; OPF passa solo `key`, `state$` e `onRead`) (integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L34-L47, integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L41-L45, core-libs/core/src/state/services/state-persistence.service.ts:L44-L48, core-libs/core/src/state/services/state-persistence.service.ts:L124-L130) e, durante il ritorno dal PSP, `sessionStorage['opfPaymentVerificationQuerySearch']` (integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L40-L41). Vedi anche 19.
- **Script PSP.** In console: `window.Opf.payments` mostra i domini registrati (integration-libs/opf/global-functions/core/facade/opf-global-functions.service.ts:L81-L87); `window.OpfContext` il contesto della modalità `SEPARATE` (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L244-L245); `document.querySelectorAll('[data-opf-resource]')` le risorse caricate (integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L186-L194). In SSR nulla di tutto questo esiste (§7).
- **Messaggi globali che "spariscono".** Possono essere stati silenziati da `OpfGlobalMessageService.disableGlobalMessage` (integration-libs/opf/base/root/services/opf-global-message.service.ts:L51-L60).
- **Ritorno dal PSP che finisce sul carrello.** Verificare il toggle `opfPaymentVerificationCheckProcessingCartOnErrorOnly` e `isPaymentInProgress` (§9.11).
- **OPPS.** Verificare in `localStorage` la chiave `opps-couponcodes` e l'header `couponcodes` sulle richieste OCC (integration-libs/opps/root/config/default-opps-config.ts:L11-L15).
- **Test.** OPF ha un `karma.conf.js` e un `test.ts` (Jasmine) (integration-libs/opf/test.ts), mentre digital-payments e opps hanno `vitest.config.ts` (integration-libs/digital-payments/vitest.config.ts, integration-libs/opps/vitest.config.ts). Comandi da `.claude/CLAUDE.md`: `nx run <library-name>:test` (Jasmine) e `nx test-vitest <library-name>`. I nomi dei progetti Nx sono `opf` (target `test`, Jasmine), `digital-payments` e `opps` (target `test-vitest`) (integration-libs/opf/project.json:L2-L2, integration-libs/opf/project.json:L20-L20, integration-libs/digital-payments/project.json:L2-L2, integration-libs/digital-payments/project.json:L20-L20, integration-libs/opps/project.json:L2-L2, integration-libs/opps/project.json:L23-L23): quindi `nx run opf:test`, `nx test-vitest digital-payments`, `nx test-vitest opps`.

---

11. Domande di autoverifica

1. **Perché il `public_api.ts` di `@spartacus/opf` è vuoto e come si importano i simboli?** Perché ogni sotto-feature ha i propri secondary entry point (`@spartacus/opf/base/root`, `/payment/root`, ...); il barrel radice esporta solo `{}`. — integration-libs/opf/public_api.ts:L14-L16, projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L105-L152

2. **Come fa il checkout standard a usare gli step OPF senza che OPF modifichi** `checkout.steps`**?** `CheckoutFlowOrchestratorService` sceglie `checkout.flows[paymentProvider]` in base al base store; OPF fornisce `checkout.flows.OPF` e forza l'istanziazione dell'orchestratore con un app initializer. — feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L20-L44, integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L58-L60, integration-libs/opf/checkout/root/config/default-opf-checkout-config.ts:L36-L45
3. **Perché l'app usa** `provideConfig` **per il flow OPF anziché affidarsi ai default?** Perché B2C e B2B root module forniscono entrambi un default per `checkout.flows.OPF` e gli array di step si sostituiscono nel merge. — integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L56-L56, integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts:L43-L43, core-libs/core/src/config/utils/deep-merge.ts:L31-L46, projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L73-L81
4. **Quali sono i tre render pattern e come vengono resi?** `IFRAME` (iframe con sandbox configurabile), `FULL_PAGE` (link o form `_self`), `HOSTED_FIELDS` (script/HTML del PSP + global functions). `dynamicScript` ha la precedenza su `destination`. — integration-libs/opf/payment/root/model/opf-payment.model.ts:L250-L254, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.component.html:L15-L69, integration-libs/opf/checkout/components/opf-checkout-payment-wrapper/opf-checkout-payment-wrapper.service.ts:L275-L331
5. **Perché esiste un location initializer in** `OpfPaymentRootModule`**?** Per salvare in `sessionStorage` la query string del redirect del PSP prima che il bootstrap OAuth modifichi `location.search`. — integration-libs/opf/payment/root/opf-payment-root.module.ts:L42-L46, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.service.ts:L71-L98
6. **Come sopravvive la sessione di pagamento al redirect 3DS se non viene persistita normalmente?** `handle3DSRedirect` imposta `is3DSRedirect: true`; la persistenza include `opfPaymentSessionId` solo in quel caso; al reload `initSync` ripristina lo stato e la pagina di verifica esegue `run3DSRedirectPattern`. — integration-libs/opf/global-functions/core/services/domains/checkout/opf-global-functions-checkout-domain.service.ts:L91-L118, integration-libs/opf/base/root/services/opf-metadata-state-persistence.service.ts:L62-L83, integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L114-L122
7. **Cosa succede se il servizio OPF risponde 401 a un submit hosted fields?** Il `backOff` con `isAuthorizationError` ripete l'intera sequenza, ottenendo un nuovo OTP, fino a `DEFAULT_AUTHORIZATION_ERROR_RETRIES_COUNT` volte. — integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L115-L122, core-libs/core/src/util/occ-http-error-handlers.ts:L26-L31, core-libs/core/src/util/rxjs/back-off.ts:L39-L61
8. **Dove viene sostituito** `CheckoutAuthGuard` **e con quale effetto?** Nel root module eager `OpfCheckoutRootModule`, con `useClass: OpfCheckoutAuthGuard`: l'override vale nell'injector root, quindi per ogni componente CMS che dichiara `CheckoutAuthGuard` (incluso `OpfCheckoutPaymentAndReview`), a differenza dell'override lazy di `CheckoutPaymentTypeFacade` (§9.17). — integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L44-L47, integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.module.ts:L57-L66, integration-libs/opf/b2b-checkout/core/facade/facade-providers.ts:L11-L17
9. **Perché il codice del PSP non viene mai eseguito in SSR?** `loadResources` ed `executeScriptFromHtml` controllano `isPlatformServer` ("SSR mode not supported for security concerns"); anche la registrazione delle global functions è solo browser. — integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L174-L184, integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L397-L406, integration-libs/opf/global-functions/core/facade/opf-global-functions.service.ts:L38-L41

10. **Come si aggiunge Digital Payments al checkout e cosa cambia nel CMS?** L'app importa `DigitalPaymentsModule` nel chunk del checkout; `DpPaymentMethodModule` sostituisce il componente `CheckoutPaymentDetails` con `provideConfig` (via `ConfigModule.withConfig`). — projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L22-L33, integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L32-L38, core-libs/core/src/config/config.module.ts:L28-L34
11. **Come fa OPPS a imporre il login prima di una pagina CMS?** Registra `OppsLoginRequiredGuard` nel multi-token `BEFORE_CMS_PAGE_GUARD`; per utenti anonimi con `?loginRequired=true` salva l'URL di redirect e restituisce l'`UrlTree` della pagina di login. — integration-libs/opps/root/login-required/opps-login-required.module.ts:L11-L20, integration-libs/opps/root/login-required/opps-login-required.guard.ts:L29-L61

TAVOLA 35.1 **4.1 Checkout OPF: dalla lista opzioni al rendering del PSP**

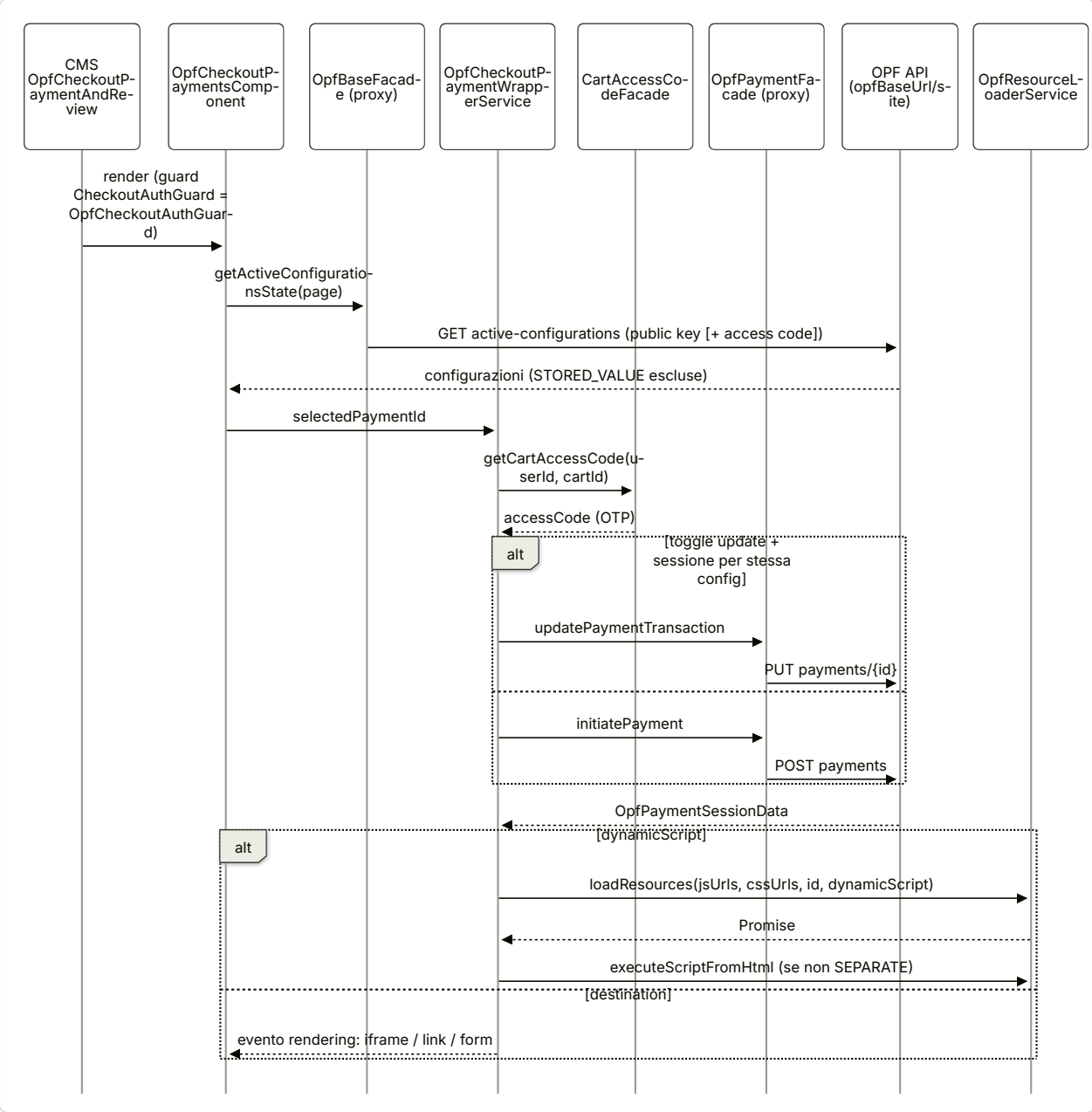


TAVOLA 35.2 **4.2 Ritorno dal PSP: scelta del pattern**

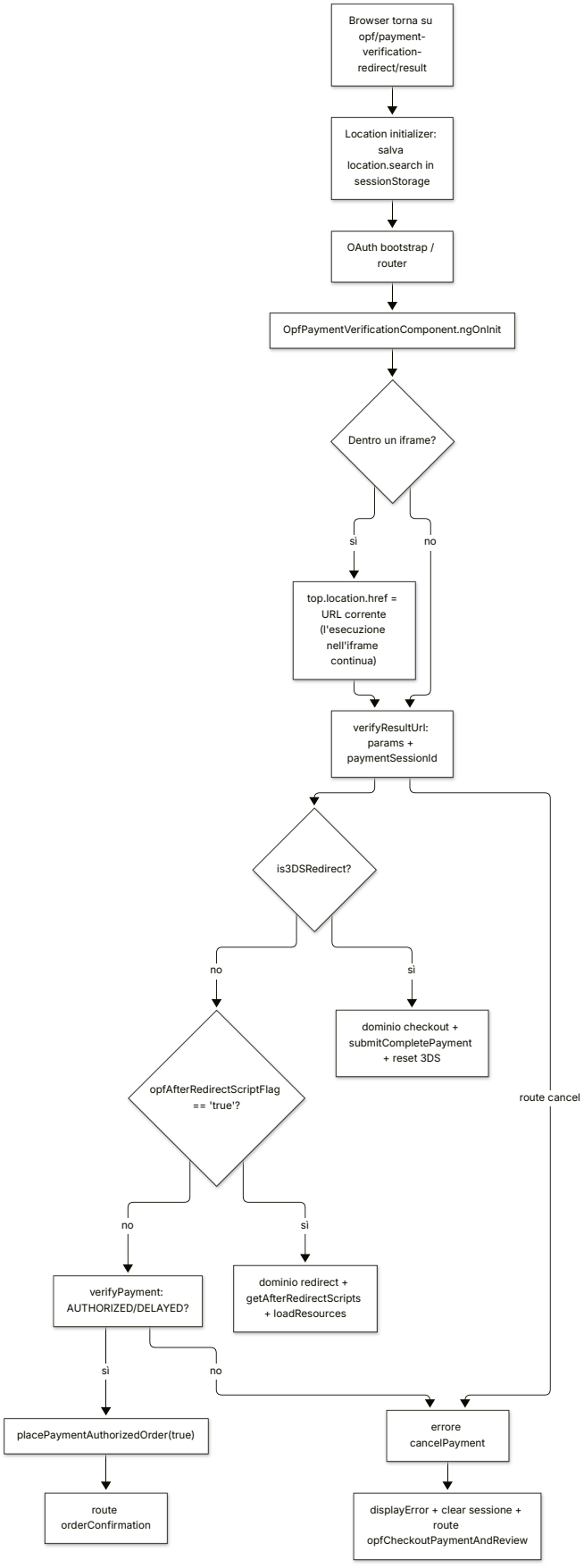
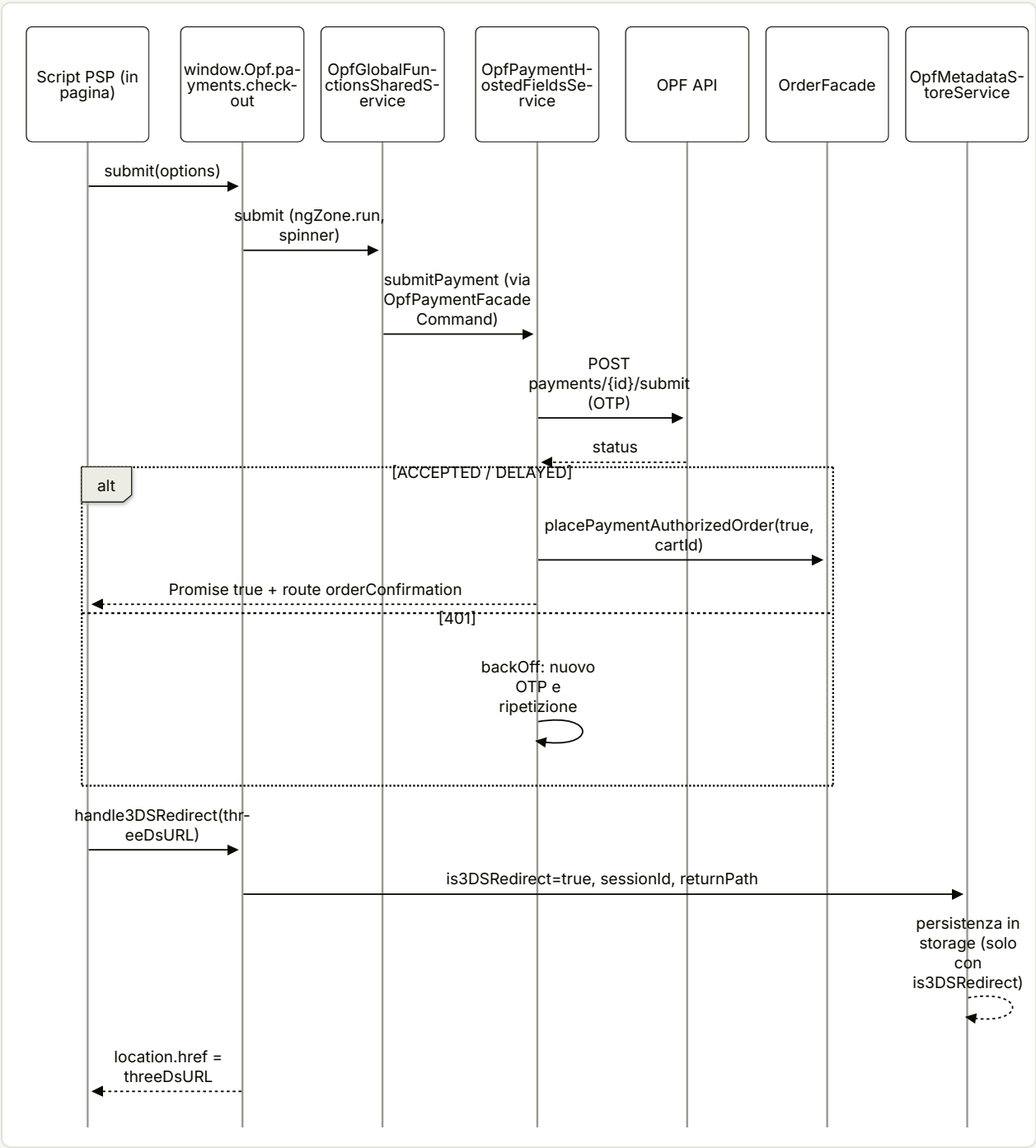


TAVOLA 35.3 **4.3 Hosted fields e 3DS: lo script PSP pilota Spartacus**



36

Integration libs: S/4, OMF, CPQ quote, punchout, EPD

Librerie per backend Commerce arricchiti da addon o sistemi SAP esterni.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Sotto `integration-libs/` vivono librerie che funzionano solo con un backend Commerce arricchito da addon o da sistemi esterni SAP. Questo capitolo ne copre sei:

LIBRERIA	PACCHETTO	COSA AGGIUNGE ALLO STOREFRONT
S/4HANA Service	`@spartacus/s4-service`	prodotti di tipo *servizio*: step di checkout "service details" (data/ora dell'intervento), cancel e reschedule del servizio dal dettaglio ordine
S/4HANA Order Management	`@spartacus/s4om`	*schedule lines* (date/quantità confermate) sotto ogni riga di carrello; prodotti "non in catalogo" restituiti da S/4
Order Management Foundation	`@spartacus/omf`	dettaglio ordine richiesto con un header `guid` aggiuntivo
CPQ quote	`@spartacus/cpq-quote`	colonna sconti CPQ nella tabella delle righe di carrello/quote
Punchout	`@spartacus/punchout`	sessioni cXML punchout avviate da un sistema di procurement esterno, con navigazione limitata e ritorno della *requisition*
EPD Visualization	`@spartacus/epd-visualization`	*visual picking* 3D/2D dei ricambi tramite SAP EPD e UI5 (libreria **deprecata**)

Le integration libs dichiarano la propria versione `"version": "221121.22.0"`, non la 2612.0.0 delle core libs (integration-libs/omf/package.json:L3-L3).

La ragione per studiarle insieme è architetturale: **quasi nessuna di queste librerie porta una feature "propria"**. Invece di definire un nuovo dominio con store, facade lazy e componenti CMS, le librerie si **innestano** su feature esistenti (order, checkout, cart, auth, product) sostituendo token DI già esistenti con `useClass`/`useExisting`, aggiungendo componenti tramite outlet, estendendo con *module augmentation* modelli e enum del core e, in un caso, sostituendo un intero token di configurazione. Sono quindi il miglior catalogo "dal vero" di come si estende Spartacus via DI, e insieme il miglior catalogo dei rischi di farlo:

- quando il token sostituito è fornito da un modulo del chunk lazy, l'override deve stare **nello stesso injector** del binding originale e **dopo** di esso: per questo l'app demo importa `OmfOrderModule`, `S4ServiceOrderModule` e `S4ServiceCheckoutModule` dentro i *wrapper module* lazy delle feature order/checkout, dopo il modulo feature (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27, projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L16-L33). Anche `CpqQuoteModule` sta nel wrapper del carrello dopo `CartBaseModule` (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25), ma lì è una scelta dell'app demo e degli schematics, non un vincolo DI: il servizio core che sostituisce, `CartItemListComponentService`, esiste solo come `providedIn: 'root'` e nessun modulo del core lo fornisce (feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.service.ts:L9-L16);
- due librerie che sovrascrivono lo stesso token (es. B2B e s4-service su `CheckoutStepsSetGuard`) entrano in competizione e vince l'ultima registrata (feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L33-L36, integration-libs/s4-service/checkout/components/s4-service-checkout-component.module.ts:L14-L27);
- il core stesso ha introdotto nuovi extension point (es. `EXPIRED_REFRESH_TOKEN_HANDLERS`) proprio per evitare che due librerie (Punchout e ASM) si contendano `AuthHttpHeaderService` (core-libs/core/src/auth/

user-auth/services/auth-http-header.service.ts:L311-L316).

Prerequisiti: 01 — Dependency Injection (gerarchia degli injector, `useClass`/`useExisting`, multi provider), 12 — Feature module e lazy loading (`featureModules`, `facadeFactory`), 17 — Outlet, 26 — Checkout, 27 — Ordini, 22 — Autenticazione.

---

2. Mappa dei file

Wiring nell'app demo

PATH	RUOLO
`projects/storefrontapp/src/environments/environment.ts`	flag `cpq`, `epdVisualization`, `s4om`, `omf`, `s4Service`, `punchout` da variabili `CX_*` (default `false`)
`projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts`	aggiunge i feature module "root" eager in base ai flag
`projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts`	`OrderModule` + `S4ServiceOrderModule` + `OmfOrderModule` (chunk lazy order)
`projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts`	`CheckoutModule` + B2B + `S4ServiceCheckoutModule`
`projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts`	`CartBaseModule` + `CpqQuoteModule`
`projects/storefrontapp/src/app/spartacus/features/s4-service/s4-service-feature.module.ts`	feature `s4-service` (lazy, modulo vuoto)
`projects/storefrontapp/src/app/spartacus/features/punchout/punchout-feature.module.ts`	loader `import('@spartacus/punchout')`
`projects/storefrontapp/src/app/spartacus/features/epd-visualization/epd-visualization-feature.module.ts`	`apis.baseUrl`, `ui5.bootstrapUrl`
`core-libs/schematics/src/shared/lib-configs/integration-libs/*-schematics-config.ts`	`importAfter` che codifica l'ordine degli import

`@spartacus/omf`

PATH	RUOLO
`integration-libs/omf/root/omf-root.module.ts`	`OmfRootModule` vuoto
`integration-libs/omf/order/omf-order.module.ts`	override DI `OrderHistoryAdapter` + `OrderHistoryFacade`, default config
`integration-libs/omf/order/occ-omf-order-history.adapter.ts`	`OccOmfOrderHistoryAdapter`: header `guid` sul dettaglio ordine
`integration-libs/omf/order/omf-order-history.service.ts`	`OmfOrderHistoryService`: `getQueryParams` → `{ guid }`
`integration-libs/omf/order/config/default-omf-config.ts`	`guidHttpHeaderName: 'guid'`
`integration-libs/omf/order/config/omf-config.ts`	token `OmfConfig`

@spartacus/s4om

PATH	RUOLO
integration-libs/s4om/root/s4om-root.module.ts	outlet ITEM_DETAILS + override ProductCatalogService
integration-libs/s4om/root/services/s4om-product-catalog.service.ts	isProductInCatalog con notOrderable
integration-libs/s4om/root/components/schedule-lines/schedule-lines.component.ts	UI schedule lines
integration-libs/s4om/root/model/augmented-core.model.ts	augmentation OrderEntry.scheduleLines
integration-libs/s4om/s4om.module.ts	entry primario vuoto

@spartacus/cpq-quote

PATH	RUOLO
integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts	3 outlet + override CartItemListComponentService
integration-libs/cpq-quote/cpq-quote-discount/cpq-qute.service.ts	CpqQuoteService: flag in BehaviorSubject (default true)
integration-libs/cpq-quote/cpq-quote-discount/cpq-qute-shared.service.ts	CpqQuoteSharedService sottoclasse del servizio core
integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-heading/cpq-quote-heading.component.ts	header colonna e set del flag
integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-discount-tbody/cpq-quote.component.ts	cella prezzo scontato
integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote/cpq-quote-offer.component.ts	percentuale di sconto
integration-libs/cpq-quote/root/model/augmented-core.model.ts	augmentation OrderEntry.cpqDiscounts

@spartacus/s4-service

PATH	RUOLO
integration-libs/s4-service/root/s4-service-root.module.ts	route figlie, override CheckoutConfig, estensione cmsComponents di checkout/order
integration-libs/s4-service/root/config/default-service-details-checkout-config.ts	6 step di checkout con serviceDetails
integration-libs/s4-service/root/config/default-service-orders-routing-config.ts	path checkout/service-details, cancelservice, rescheduleservice
integration-libs/s4-service/root/config/default-service-delivery-mode-config.ts	delivery mode service-delivery
integration-libs/s4-service/root/model/augmented-types.model.ts	augmentation Order, CheckoutState, BaseStore, CheckoutStepType, OccEndpoints
integration-libs/s4-service/root/facade/checkout-service-details.facade.ts	facade lazy

PATH	RUOLO
integration-libs/s4-service/root/facade/cancel-service-order.facade.ts, integration-libs/s4-service/root/facade/reschedule-service-order.facade.ts	facade astratte senza factory
integration-libs/s4-service/root/facade/checkout-service-schedule-picker.service.ts	utility date/ora
integration-libs/s4-service/root/events/checkout-service-details-event.listener.ts	evento → `CheckoutQueryResetEvent`
integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts	guard degli step (estende quello B2B)
integration-libs/s4-service/checkout/components/checkout-service-details/checkout-service-details.component.ts	nuovo componente CMS `CheckoutServiceDetails`
integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.component.ts	override CMS `CheckoutDeliveryMode`
integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.component.ts	override CMS `CheckoutReviewOrder`
integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts	Command `setServiceScheduleSlot`
integration-libs/s4-service/checkout/occ/adapter/occ-checkout-service-details.adapter.ts	`PATCH .../serviceScheduleSlot`
integration-libs/s4-service/order/core/s4-service-order-core.module.ts	binding facade cancel/reschedule
integration-libs/s4-service/order/occ/adapters/occ-cancel-service-order.adapter.ts, integration-libs/s4-service/order/occ/adapters/occ-reschedule-service-order.adapter.ts	POST cancel / PATCH reschedule
integration-libs/s4-service/order/components/	detail actions, cancel, reschedule, card servizio, guard

@spartacus/punchout

PATH	RUOLO
integration-libs/punchout/root/punchout.root.module.ts	initializer persistenza, interceptor, override `AuthHttpHeaderService`, handler refresh token, bootstrap listener
integration-libs/punchout/root/guards/punchout-navigation.guard.ts	`BEFORE_CMS_PAGE_GUARD` che limita la navigazione
integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts	header `punchoutsid`
integration-libs/punchout/root/services/punchout-store.service.ts	stato in `BehaviorSubject`
integration-libs/punchout/root/services/punchout-state-persistence.service.ts	persistenza del session id
integration-libs/punchout/root/services/punchout-expired-refresh-token-handler.service.ts	handler `EXPIRED_REFRESH_TOKEN_HANDLERS`
integration-libs/punchout/root/services/punchout-auth-http-header.service.ts	override legacy di `AuthHttpHeaderService`

PATH	RUOLO
integration-libs/punchout/root/config/default-punchout-guard-config.ts	rotte ammesse per operazione
integration-libs/punchout/root/facade/punchout.facade.ts	facade lazy
integration-libs/punchout/core/facade/punchout.service.ts	Command di sessione/requisition
integration-libs/punchout/core/services/punchout-auth.service.ts	login con token, logout, fine sessione
integration-libs/punchout/occ/adapters/occ-punchout.adapter.ts	punchout/sessions/...
integration-libs/punchout/components/	session, requisition, buttons, close-session, inspect-cart, error

@spartacus/epd-visualization

PATH	RUOLO
integration-libs/epd-visualization/root/epd-visualization-root.module.ts	feature mapping, default config, validator
integration-libs/epd-visualization/root/config/epd-visualization-config.ts	EpdVisualizationConfig
integration-libs/epd-visualization/root/config/epd-visualization-config-validator.ts	validator
integration-libs/epd-visualization/epd-visualization-api/epd-visualization-api.module.ts	binding adapter → REST v1
integration-libs/epd-visualization/epd-visualization-api/adapters/storage-v1/storage-v1.adapter.ts	nodi della scena
integration-libs/epd-visualization/epd-visualization-api/adapters/visualization-v1/visualization-v1.adapter.ts	lookup visualizzazioni
integration-libs/epd-visualization/core/services/visualization-lookup/visualization-lookup.service.ts	ricerca per usage id prodotto
integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts	mappe nodo ↔ prodotto
integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts	bootstrap UI5, viewport, caricamento scena
integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.service.ts	collega lista prodotti e viewer

File aggiuntivi

PATH	RUOLO
integration-libs/omf/public_api.ts	Entry point principale: riesporta solo `root/public_api` (:L7-L7).
integration-libs/omf/schematics/add-omf/index.ts	Factory `ng-add` `addOmfFeature` (:L25-L43).
integration-libs/omf/schematics/add-omf/index_spec.ts	Test Jest dello schematic OMF (con feature order) (:L24-L145).

PATH	RUOLO
integration-libs/omf/vitest.config.ts	Config Vitest con coverage sempre attiva e soglie 90/90/75/90 (:L24-L46).
integration-libs/s4om/public_api.ts	Entry point principale (dichiarato in ng-package.json): riesporta s4om.module (:L10-L10).
integration-libs/s4om/index.ts	Doppione di public_api.ts con lo stesso export (:L7-L7).
integration-libs/s4om/_index.scss	Entry Sass: core + styles/index (:L1-L3).
integration-libs/s4om/styles/_index.scss	Importa schedule-lines (:L1-L1).
integration-libs/s4om/styles/schedule-lines.scss	Regole globali per cx-schedule-lines e .cx-schedule-line-info (:L1-L27).
integration-libs/s4om/schematics/add-s4om/index.ts	Factory ng-add addS4OMFeature (:L25-L43).
integration-libs/s4om/schematics/add-s4om/index_spec.ts	Test Jest dello schematic S4OM (:L21-L114).
integration-libs/s4om/schematics/constants.ts	Costanti di nomi (cartella, modulo, traduzioni) non importate da nessun file (:L9-L17).
integration-libs/s4om/vitest.config.ts	Config Vitest (soglie 90) (:L42-L47).
integration-libs/cpq-quote/schematics/add-cpq-quote/index.ts	Factory ng-add addCpqQuoteFeature (:L25-L43).
integration-libs/cpq-quote/schematics/add-cpq-quote/index_spec.ts	Test Jest dello schematic CPQ (con feature cart-base) (:L25-L151).
integration-libs/cpq-quote/vitest.config.ts	Config Vitest: coverage solo lcov, senza enabled (:L26-L42).
integration-libs/s4-service/public_api.ts	Entry point principale: riesporta s4-service.module (:L7-L7).
integration-libs/s4-service/styles/_index.scss	Importa i cinque partial dei componenti (:L1-L5).
integration-libs/s4-service/styles/components/_checkout-service-details.scss	Placeholder %cx-service-details (:L1-L29).
integration-libs/s4-service/styles/components/_cancel-service-order.scss	Placeholder %cx-cancel-service-order (form motivo) (:L1-L42).
integration-libs/s4-service/styles/components/_cancel-service-order-headline.scss	Placeholder %cx-cancel-service-order-headline (tabella) (:L1-L38).
integration-libs/s4-service/styles/components/_reschedule-service-order.scss	Classi globali .cx-reschedule-service-*, .cx-section-* (:L1-L34).
integration-libs/s4-service/styles/components/_s4-service-order-detail-actions.scss	Classe globale .cx-order-details-actions (:L1-L7).
integration-libs/s4-service/vitest.config.ts	Config Vitest con alias verso helper di test del core (:L16-L22).
integration-libs/punchout/public_api.ts	Entry point principale: riesporta punchout.module (:L10-L10).
integration-libs/punchout/_index.scss	Entry Sass: allowlist componenti, __body e allowlist page template PunchoutTemplate (:L7-L40).
integration-libs/punchout/schematics/add-punchout/index.ts	Factory ng-add addPunchoutFeatures (:L25-L43).
integration-libs/punchout/schematics/add-punchout/index_spec.ts	Test Jest: solo il caso "senza feature" (:L31-L45).

PATH	RUOLO
integration-libs/punchout/test.ts	Bootstrap Karma con provideZoneChangeDetection() (:L14-L30).
integration-libs/epd-visualization/public_api.ts	Entry point principale: riesporta epd-visualization.module (:L7-L7).
integration-libs/epd-visualization/schematics/add-epd-visualization/index.ts	Factory ng-add addEpdVisualizationFeature, deprecata (:L25-L48).
integration-libs/epd-visualization/schematics/add-epd-visualization/index_spec.ts	Test Jest, anche con tsconfig senza compilerOptions (:L132-L189).
integration-libs/epd-visualization/test.ts	Bootstrap Karma con provideZoneChangeDetection() (:L16-L32).
integration-libs/epd-visualization/styles/_index.scss	Importa components/index (:L1-L1).
integration-libs/epd-visualization/styles/components/_index.scss	Importa visual-picking e visual-viewer (:L1-L2).
integration-libs/epd-visualization/styles/components/visual-picking/_index.scss	Importa i 4 partial del visual picking (:L1-L4).
integration-libs/epd-visualization/styles/components/visual-picking/paged-list.component.scss	Placeholder %cx-epd-visualization-paged-list (carosello a slide) (:L1-L110).
integration-libs/epd-visualization/styles/components/visual-picking/visual-picking-product-filter.component.scss	Placeholder %cx-epd-visualization-product-filter (:L1-L49).
integration-libs/epd-visualization/styles/components/visual-picking/visual-picking-product-list.component.scss	Placeholder %cx-epd-visualization-product-list (thumbnail, highlight) (:L1-L242).
integration-libs/epd-visualization/styles/components/visual-picking/visual-picking-tab.component.scss	Placeholder %cx-epd-visualization-visual-picking-tab (viewer + lista affiancati da md) (:L1-L66).
integration-libs/epd-visualization/styles/components/visual-viewer/_index.scss	Importa i 3 partial del viewer (:L1-L3).
integration-libs/epd-visualization/styles/components/visual-viewer/visual-viewer-animation-slider.component.scss	Placeholder %cx-epd-visualization-animation-slider (:L1-L73).
integration-libs/epd-visualization/styles/components/visual-viewer/visual-viewer-toolbar-button.component.scss	Placeholder %cx-epd-visualization-viewer-toolbar-button (:L1-L79).
integration-libs/epd-visualization/styles/components/visual-viewer/visual-viewer.component.scss	Placeholder %cx-epd-visualization-viewer (griglia, viewport UI5) (:L1-L155).

3. Come funziona passo per passo

3.1 Il pattern comune: "root eager + estensione nel wrapper lazy"

Ogni libreria è attivata nell'app demo da un flag d'ambiente con default `false` (projects/storefrontapp/src/environments/environment.ts:L30-L43). Quando il flag è attivo, il feature module "root" della libreria entra nei `featureModules` eager di `SpartacusFeaturesModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L133-L175).

Le parti che **sovrascrivono** binding di altre feature non sono però caricate come feature lazy proprie:

- `S4ServiceOrderModule` e `OmfOrderModule` sono importati **dopo** `OrderModule` dentro `OrderWrapperModule` (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27), che è proprio il modulo caricato lazy dalla feature order (projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L29-L30);
- `S4ServiceCheckoutModule` è importato in `CheckoutWrapperModule` **dopo** `CheckoutB2BModule` (che entra solo con `environment.b2b`) (projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L16-L33);
- `CpqQuoteModule` è importato dopo `CartBaseModule` in `CartBaseWrapperModule` (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25).

Il motivo è la gerarchia degli injector (vedi 03 — Lazy loading): i provider di `OrderModule` vivono nell'injector del chunk lazy; un override nell'AppModule sarebbe "più in alto" e quindi oscurato dal binding originale. Nello stesso injector, a parità di token, vince il provider registrato per ultimo, quindi l'ordine `[OrderModule, ...extensions]` è essenziale. DA VERIFICARE a runtime che un override in AppModule venga effettivamente oscurato (deduzione dalla gerarchia degli injector lazy, `node_modules` assente).

Gli schematics codificano lo stesso vincolo con `importAfter` (coppie *marker module* → *feature module*): s4-service va dopo `CHECKOUT_B2B_MODULE` e `ORDER_MODULE` (core-libs/schematics/src/shared/lib-configs/integration-libs/s4-service-schematics-config.ts:L74-L83), `OMF_ORDER_MODULE` dopo `ORDER_MODULE` con dipendenza dalla feature order (core-libs/schematics/src/shared/lib-configs/integration-libs/omf-schematics-config.ts:L29-L50), `CPQ_QUOTE_MODULE` dopo `CART_BASE_MODULE` (core-libs/schematics/src/shared/lib-configs/integration-libs/cpq-quote-schematics-config.ts:L45-L51). Gli schematic `add-*` delle librerie sono tutti uguali: analizzano le dipendenze cross-feature, chiamano `addFeatures` e aggiungono le peer dependencies (integration-libs/s4-service/schematics/add-s4-service/index.ts:L25-L42). Per i dettagli vedi 43 — Schematics e migrazioni.

3.2 OMF: un header `guid` sul dettaglio ordine

`OmfRootModule` è un `@NgModule()` vuoto e la costante `OMF_FEATURE = 'OMF'` non è usata da alcun `featureModules` (integration-libs/omf/root/omf-root.module.ts:L9-L10, integration-libs/omf/root/feature-name.ts:L7-L7). Tutto il lavoro è in `OmfOrderModule`, che sostituisce **due** token della feature order (integration-libs/omf/order/omf-order.module.ts:L15-L26):

- `OrderHistoryAdapter` → `OccOmfOrderHistoryAdapter` (`useClass`), al posto di `OccOrderHistoryAdapter` (feature-libs/order/occ/order-occ.module.ts:L42-L42);
- `OrderHistoryFacade` → `OmfOrderHistoryService` (`useClass`), al posto di `useExisting: OrderHistoryService` (feature-libs/order/core/facade/facade-providers.ts:L33-L37).

Il flusso:

1. Lo storico ordini usa `orderHistoryFacade.getQueryParams(order)` come `queryParams` dei link verso il dettaglio (feature-libs/order/components/order-history/order-history.component.ts:L126-L133). Nel core il metodo restituisce `null` ed è `@deprecated` (feature-libs/order/core/facade/order-history.service.ts:L220-L225).
2. `OmfOrderHistoryService` estende `OrderHistoryService` e sovrascrive solo `getQueryParams`, restituendo `{ guid }` se l'ordine lo possiede (integration-libs/omf/order/omf-order-history.service.ts:L12-L24). Il campo `guid?: string` è già nel modello `Order` del core (feature-libs/order/root/model/order.model.ts:L86-L86).
3. Il dettaglio ordine invoca `OrderHistoryAdapter.load`, che ora è `OccOmfOrderHistoryAdapter.load` (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L69-L86).

4. `getOrderGuid` legge `guid` dai query param della rotta attiva; se manca (caso "My-Account-V2") lo cerca nella lista ordini dello store NgRx tramite `OrderSelectors.getOrdersState` (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L31-L53).
5. `getRequestHeader` mette il guid in un header il cui nome è `omf.guidHttpHeaderName` convertito in minuscolo; se la chiave manca e si è in dev mode logga un warning (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L55-L67). Il default è `'guid'` (integration-libs/omf/order/config/default-omf-config.ts:L9-L13).
6. Per l'utente anonimo aggiunge anche l'header `USE_CLIENT_TOKEN`, poi fa la GET verso `getOrderDetailUrl` e normalizza con `ORDER_NORMALIZER` (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L69-L86). `logger` e `getOrderDetailUrl` sono membri `protected` della superclasse (feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L47-L47, feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L212-L212).

`OmfConfig` è il classico config token astratto risolto `useExisting: Config`, con il tipo `OmfConfiguration` preso dall'entry primario `@spartacus/omf` (integration-libs/omf/order/config/omf-config.ts:L7-L21).

3.3 S/4HANA Order Management: schedule lines e prodotti fuori catalogo

`S4omRootModule` (eager) fa due cose (integration-libs/s4om/root/s4om-root.module.ts:L17-L35):

- registra `ScheduleLinesComponent` nell'outlet `CartOutlets.ITEM_DETAILS` in posizione `AFTER`;
- sostituisce `ProductCatalogService` con `useExisting: S4omProductCatalogService`.

Importa inoltre `RequestedDeliveryDateComponentsModule` e `PDFInvoicesComponentsModule` solo per forzare l'installazione di quelle librerie insieme a S4OM, come dice il commento in linea (integration-libs/s4om/root/s4om-root.module.ts:L18-L22). L'entry primario `S4omModule` è vuoto (integration-libs/s4om/s4om.module.ts:L8-L12).

Prodotti fuori catalogo. Il `ProductCatalogService` core considera "in catalogo" qualunque prodotto non nullo (core-libs/core/src/product/services/product-catalog.service.ts:L20-L22). La versione S4OM ritorna `false` se `stock.stockLevelStatus === 'notOrderable'` (prodotto arrivato da S/4 ma assente in Commerce) e `true` se lo status manca (integration-libs/s4om/root/services/s4om-product-catalog.service.ts:L13-L37). Il metodo è consumato dal carrello (link disabilitato nella lista item, add-to-cart "unavailable") e dall'ordine (reorder) (feature-libs/cart/base/components/cart-shared/cart-item-list/cart-item-list.component.ts:L395-L398, feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L158-L160, feature-libs/order/components/order-details/order-detail-reorder/order-detail-reorder.component.ts:L53-L53). Poiché il binding è nel root module eager, vale per tutto l'albero.

Schedule lines. `OrderEntry` è aumentato con `scheduleLines?: ScheduleLine[]`, dove `ScheduleLine = { confirmedAt?: Date; confirmedQuantity?: number }` (integration-libs/s4om/root/model/augmented-core.model.ts:L10-L14, integration-libs/s4om/root/model/schedule-line.model.ts:L7-L10); l'augmentation è importata per side effect dall'index del model (integration-libs/s4om/root/model/index.ts:L7-L9). `ScheduleLinesComponent` legge l'entry da `CartItemContext` (opzionale, fallback `EMPTY`) e fornisce localmente `CxDatePipe` (integration-libs/s4om/root/components/schedule-lines/schedule-lines.component.ts:L13-L27). Il template mostra le righe solo se `hasScheduleLines`, con il testo visivo `aria-hidden` e un testo `cx-visually-hidden` collegato da `aria-describedby` (integration-libs/s4om/root/components/schedule-lines/schedule-lines.component.html:L1-L20).

3.4 CPQ quote: sconti nella tabella delle righe

`CpqQuoteRootModule` è vuoto; tutta la logica è in `CpqDiscountModule`, importato da `CpqQuoteModule` (integration-libs/cpq-quote/root/cpq-quote-root.module.ts:L9-L10, integration-libs/cpq-quote/cpq-

quote.module.ts:L10-L13). CpqDiscountModule (integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L39-L57):

- sostituisce CartItemListComponentService con useClass: CpqQuoteSharedService;
- registra tre componenti negli outlet CPQ_QUOTE_MODULE, CPQ_QUOTE_HEADING, CPQ_QUOTE (tutti AFTER).

Gli id degli outlet sono definiti nel **core cart** (cx-cpq-quote, cx-cpq-quote-heading, cx-cpq-quote-offer) (feature-libs/cart/base/root/models/cart-outlets.model.ts:L25-L27) e piazzati nei template core: CPQ_QUOTE_HEADING nell'intestazione della tabella con context items, CPQ_QUOTE e CPQ_QUOTE_MODULE nella riga con context { items: item } (feature-libs/cart/base/components/cart-shared/cart-item-list/cart-item-list.component.html:L21-L24, feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L44-L49, feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L106-L112). È un caso esemplare di "hook predisposto nel core per un'integrazione".

Il meccanismo del flag:

1. Il CartItemListComponentService core è providedIn: 'root' e ha un solo metodo showBasePriceWithDiscount() che ritorna true (feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.service.ts:L9-L16).
2. CartItemListRowComponent lo legge **una volta**, inizializzando il campo isFlagQuote, e lo usa in *ngIf per il prezzo base standard (feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.ts:L48-L49, feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L37-L37).
3. CpqQuoteSharedService estende il servizio core e delega a CpqQuoteService.getFlag$(), che nonostante il $ ritorna un boolean sincrono dal BehaviorSubject (default true) (integration-libs/cpq-quote/cpq-quote-discount/cpq-qute-shared.service.ts:L16-L23, integration-libs/cpq-quote/cpq-quote-discount/cpq-qute.service.ts:L13-L22).
4. CpqQuoteHeadingComponent riceve dal context outlet l'array di entry e imposta il flag a !hasDiscounts: se almeno un item ha cpqDiscounts, il prezzo base standard va nascosto e compare la colonna sconto (integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-heading/cpq-quote-heading.component.ts:L55-L72).
5. CpqQuoteDiscountComponent.getDiscountedPrice calcola il prezzo unitario scontato come (basePrice*quantity - appliedDiscount)/quantity (integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-discount-tbody/cpq-quote.component.ts:L55-L71), formattato con number: '1.2-2' : 'en-US' e con il prezzo base barrato se ci sono sconti (integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-discount-tbody/cpq-quote.component.html:L12-L31).
6. CpqQuoteOfferComponent mostra la percentuale appliedDiscount/(basePrice*quantity)*100, senza decimali se intera, altrimenti con 2 (integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote/cpq-quote-offer.component.ts:L52-L74).

Il dato arriva dal backend: OrderEntry è aumentato con cpqDiscounts?: CpqDiscounts[] (campi appliedValue, isoCode, value tipati any) (integration-libs/cpq-quote/root/model/augmented-core.model.ts:L10-L14, integration-libs/cpq-quote/root/model/cpqDiscounts.model.ts:L7-L11).

3.5 S/4HANA Service — parte root (eager)

S4ServiceRootModule (integration-libs/s4-service/root/s4-service-root.module.ts:L56-L83):

- importa CheckoutServiceDetailsEventModule;

- registra due route figlie `path: ''` identificate da `data.cxRoute` (`cancelServiceDetails`, `rescheduleServiceDetails`) con `CmsPageGuard` + `PageLayoutComponent` (integration-libs/s4-service/root/s4-service-root.module.ts:L59-L72). I path reali arrivano dalla routing config: `my-account/order/cancelservice/:orderCode` e `my-account/order/rescheduleservice/:orderCode` (con `paramsMapping orderCode → code`), più `checkout/service-details` (integration-libs/s4-service/root/config/default-service-orders-routing-config.ts:L9-L23). È il meccanismo generale per cui Spartacus identifica il nome di una route da `data.cxRoute` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L146-L150; vedi 14 — Routing);
- **sostituisce il token** `CheckoutConfig` con `useValue: defaultServiceDetailsCheckoutConfig` (integration-libs/s4-service/root/s4-service-root.module.ts:L75-L75). Normalmente `CheckoutConfig` è un token astratto `useExisting: Config` che espone la configurazione globale fusa (feature-libs/checkout/base/root/config/checkout-config.ts:L18-L22): qui invece chi inietta `CheckoutConfig` riceve un oggetto fisso;
- fornisce default config per routing, delivery mode di servizio e `cmsComponents`, più `CxDatePipe` e `CheckoutServiceSchedulePickerService` (integration-libs/s4-service/root/s4-service-root.module.ts:L76-L80).

`defaultServiceDetailsCheckoutConfig` definisce 6 step (paymentType, deliveryAddress, deliveryMode, paymentDetails, **serviceDetails**, reviewOrder), `express: false`, `guest: false`, `defaultDeliveryMode: [FREE]` (integration-libs/s4-service/root/config/default-service-details-checkout-config.ts:L13-L57).

Estendere un enum del core. `CheckoutStepType.SERVICE_DETAILS` viene aggiunto sia per il type checker (augmentation di `enum`) sia a runtime con un'assegnazione diretta all'oggetto enum (integration-libs/s4-service/root/model/augmented-types.model.ts:L38-L47). Altre augmentation: `Order` riceve `servicedAt`, `serviceCancellable`, `serviceReschedulable`; `CheckoutState` riceve `servicedAt`; `BaseStore` riceve `serviceOrderConfiguration { leadDays, serviceScheduleTimes }`; `OccEndpoints` riceve `setServiceScheduleSlot` (integration-libs/s4-service/root/model/augmented-types.model.ts:L16-L52). `S4ServiceDeliveryModeConfig` è un config token con `s4ServiceDeliveryMode?: DeliveryMode` (integration-libs/s4-service/root/model/augmented-types.model.ts:L54-L64), di default `code: 'service-delivery'`, costo `USD0.00` (integration-libs/s4-service/root/config/default-service-delivery-mode-config.ts:L10-L22).

Nessuna feature lazy propria (di fatto). s4-service non mappa i suoi componenti CMS su una feature sua: estende `featureModules[CHECKOUT_FEATURE].cmsComponents` (B2B + `CheckoutServiceDetails`) e `featureModules[ORDER_FEATURE].cmsComponents` (order + 3 componenti service) (integration-libs/s4-service/root/s4-service-root.module.ts:L31-L55). Siccome `deepMerge` **sostituisce** gli array invece di concatenarli (`isObject` esclude gli array, quindi l'ultimo valore vince) (core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/config/utils/deep-merge.ts:L36-L45), la libreria deve ripetere gli elenchi B2B/order completi. L'app demo registra comunque una feature `S4_SERVICE_FEATURE` (`'s4-service'`) che carica `S4ServiceModule`, il quale però è vuoto (projects/storefrontapp/src/app/spartacus/features/s4-service/s4-service-feature.module.ts:L26-L29, integration-libs/s4-service/s4-service.module.ts:L9-L12).

Facade. `CheckoutServiceDetailsFacade` è una facade lazy (`facadeFactory`) legata a `CHECKOUT_CORE_FEATURE` con 5 metodi (integration-libs/s4-service/root/facade/checkout-service-details.facade.ts:L13-L27); `CHECKOUT_CORE_FEATURE` è alias di `CHECKOUT_FEATURE` (feature-libs/checkout/base/root/checkout-root.module.ts:L44-L45), quindi il proxy carica il chunk checkout. Al contrario `CancelServiceOrderFacade` e `RescheduleServiceOrderFacade` sono classi astratte `providedIn: 'root'` **senza** factory: risolvibili solo dove `S4ServiceOrderCoreModule` le ha fornite (integration-libs/s4-service/root/facade/cancel-service-order.facade.ts:L12-L30, integration-libs/s4-service/root/facade/reschedule-service-order.facade.ts:L11-L24).

Utility date. `CheckoutServiceSchedulePickerService` legge `leadDays` e `serviceScheduleTimes` dal `baseSite.baseStore.serviceOrderConfiguration` (default 0 / `[]`) (integration-libs/s4-service/root/facade/checkout-

service-schedule-picker.service.ts:L48-L61); la data minima è oggi + `leadDays` + 1, formattata `yyyy-MM-dd` (integration-libs/s4-service/root/facade/checkout-service-schedule-picker.service.ts:L24-L32); `convertToDateTime` produce `YYYY-MM-DDTHH:MM:00` + offset del fuso locale (integration-libs/s4-service/root/facade/checkout-service-schedule-picker.service.ts:L70-L74); `convertDateTimeToReadableString` usa `Date.toLocaleString()` (integration-libs/s4-service/root/facade/checkout-service-schedule-picker.service.ts:L82-L85).

Evento. `CheckoutServiceDetailsEventListener` traduce ogni `CheckoutServiceDetailsSetEvent` in un `CheckoutQueryResetEvent` (integration-libs/s4-service/root/events/checkout-service-details-event.listener.ts:L23-L29), che è tra i notifier di reset della query dei dettagli checkout (feature-libs/checkout/base/core/facade/checkout-query.service.ts:L38-L39). Il listener è istanziato eagerly iniettandolo nel costruttore di `CheckoutServiceDetailsEventModule` (integration-libs/s4-service/root/events/checkout-service-details-event.module.ts:L10-L17). Per il pattern vedi 20 — Facade, command & query, eventi.

3.6 S/4HANA Service — checkout

`S4ServiceCheckoutModule` = components + core + occ (integration-libs/s4-service/checkout/s4-service-checkout.module.ts:L12-L19).

- **Core**: `CheckoutServiceDetailsFacade → useExisting CheckoutServiceDetailsService`, più il connector (integration-libs/s4-service/checkout/core/s4-service-checkout-core.module.ts:L13-L24). Il Command `setServiceScheduleSlotCommand` (strategia `CancelPrevious`) verifica le precondizioni, chiama il connector con `{ scheduledAt }` e a successo emette `CheckoutServiceDetailsSetEvent` (integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L39-L65). Precondizioni: `userId` e `cartId` presenti, utente anonimo ammesso solo con guest cart, altrimenti `Error('Checkout conditions not met')` (integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L67-L85). Lo stato della data servizio è derivato dalla query dei dettagli checkout (`state.data?.servicedAt`), senza store dedicato (integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L91-L97). `getServiceProducts` filtra le entry con `productTypes === ProductTypes.SERVICE`; `hasNonServiceItems` confronta il numero di entry in consegna (`getDeliveryEntries()`, il pickup non si applica ai servizi) con quello dei servizi (integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L99-L131).
- **OCC**: `CheckoutServiceDetailsAdapter → OccCheckoutServiceDetailsAdapter` e default config degli endpoint (integration-libs/s4-service/checkout/occ/s4-service-checkout-occ.module.ts:L13-L22). La config ridefinisce `getCheckoutDetails` aggiungendo `servicedAt` ai fields e introduce `setServiceScheduleSlot` (integration-libs/s4-service/checkout/occ/config/default-occ-checkout-s4-service-config.ts:L10-L25). L'adapter fa una `PATCH` del body e normalizza gli errori con `tryNormalizeHttpError` (integration-libs/s4-service/checkout/occ/adapter/occ-checkout-service-details.adapter.ts:L27-L44).
- **Components**:
  - `CheckoutStepsSetGuard` è sostituito con `useExisting: CheckoutServiceOrderStepsSetGuard` (integration-libs/s4-service/checkout/components/s4-service-checkout-component.module.ts:L14-L27). È il guard dichiarato dai componenti CMS di checkout progress (feature-libs/checkout/base/components/checkout-progress/checkout-progress.module.ts:L37-L37); anche il checkout B2B sostituisce lo stesso token (feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L33-L36), quindi conta l'ordine di import nel wrapper (§3.1).
  - `CheckoutServiceOrderStepsSetGuard` estende `CheckoutB2BStepsSetGuard`; in `canActivate` disabilita SERVICE_DETAILS se il carrello non ha servizi e DELIVERY_MODE se ha solo servizi, poi delega a `super.canActivate` (integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts:L21-L43). Sovrascrive `isB2BStepSet` aggiungendo il caso `SERVICE_DETAILS → isServiceDetailsSet` (integration-libs/s4-service/checkout/components/guards/checkout-service-

order-steps-set.guard.ts:L78-L99). isServiceDetailsSet prima chiama setServiceDeliveryMode() — che, se il carrello ha SOLO servizi, imposta il delivery mode s4ServiceDeliveryMode.code — poi ritorna true o l'URL dello step (integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts:L45-L76).

- Override CMS CheckoutDeliveryMode → ServiceCheckoutDeliveryModeComponent (estende CheckoutDeliveryModeComponent) (integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.module.ts:L34-L42, integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.component.ts:L40-L47); il template mostra prima, se ci sono servizi, l'opzione "service delivery" presa dalla config (integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.component.html:L1-L5, integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.component.html:L133-L149).
- Override CMS CheckoutReviewOrder → ServiceCheckoutReviewSubmitComponent (estende B2BCheckoutReviewSubmitComponent) con guard CheckoutAuthGuard, CartNotEmptyGuard (integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.module.ts:L40-L49); aggiunge SERVICE_DETAILS ai "delivery steps" e nasconde la card delivery mode quando il mode è quello di servizio (integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.component.ts:L116-L126).
- Nuovo componente CMS CheckoutServiceDetails (cx-service-details) (integration-libs/s4-service/checkout/components/checkout-service-details/checkout-service-details.module.ts:L32-L41). Il form è precompilato con la data salvata o con data minima e primo orario disponibile (integration-libs/s4-service/checkout/components/checkout-service-details/checkout-service-details.component.ts:L85-L109); next() converte data+ora, chiama setServiceScheduleSlot e a successo checkoutStepService.next, a errore mostra serviceOrderCheckout.unknownError (integration-libs/s4-service/checkout/components/checkout-service-details/checkout-service-details.component.ts:L129-L164).

3.7 S/4HANA Service — order

- **Core**: CancelServiceOrderFacade → CancelServiceOrderService e RescheduleServiceOrderFacade → RescheduleServiceOrderService (useExisting), più i due connector (integration-libs/s4-service/order/core/s4-service-order-core.module.ts:L22-L39). cancelService prende lo userId e chiama il connector; loadOrderDetails delega a OrderHistoryFacade.getOrderDetails() (integration-libs/s4-service/order/core/facade/cancel-service-order.service.ts:L20-L38).
- **OCC**: cancelServiceOrder → users/${userId}/orders/${code}/serviceOrder/cancellation, rescheduleService → .../serviceOrder/serviceScheduleSlot (integration-libs/s4-service/order/occ/config/default-occ-s4-service-config.ts:L10-L15). Cancel è una POST senza normalizzazione errori; reschedule è una PATCH con tryNormalizeHttpError (integration-libs/s4-service/order/occ/adapters/occ-cancel-service-order.adapter.ts:L20-L32, integration-libs/s4-service/order/occ/adapters/occ-reschedule-service-order.adapter.ts:L26-L42).
- **Components**:
  - ServiceDetailsCardModule sostituisce OrderOverviewComponentService con ServiceOrderOverviewComponentService e mette ServiceDetailsCardComponent nell'outlet OrderOutlets.SERVICE_DETAILS (integration-libs/s4-service/order/components/order-summary/service-details-card.module.ts:L16-L31). Il servizio nasconde il delivery mode di servizio, altrimenti delega a super (integration-libs/s4-service/order/components/order-summary/service-order-overview-component.service.ts:L12-L20, feature-libs/order/components/order-details/order-overview/order-overview-component.service.ts:L12-L14). L'outlet cx-service-details-card è già nel template core dell'order overview con context { item: order, readonly: true } (feature-libs/

order/root/model/order-outlets.model.ts:L13-L13, feature-libs/order/components/order-details/order-overview/order-overview.component.html:L112-L115).

- Override CMS `AccountOrderDetailsActionsComponent → S4ServiceOrderDetailActionsComponent` (integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.module.ts:L28-L37); il template mostra i bottoni cancel/reschedule in base ai flag API `order.serviceCancellable/serviceReschedulable` e linka le cxRoute `cancelServiceDetails/rescheduleServiceDetails` (integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.component.html:L42-L57).
- Nuovi componenti CMS `CancelServiceOrder` (guard `AuthGuard, CancelServiceOrderGuard`), `CancelServiceOrderHeadline` (`AuthGuard`), `RescheduleServiceOrder` (`AuthGuard, ServiceOrderGuard`) (integration-libs/s4-service/order/components/cancel-service-order/cancel-service-order.module.ts:L34-L41, integration-libs/s4-service/order/components/cancel-service-order-headline/cancel-service-order-headline.module.ts:L38-L45, integration-libs/s4-service/order/components/reschedule-service-order/reschedule-service-order.module.ts:L45-L52).
- `CancelServiceOrderGuard` passa solo se l'ordine ha `serviceCancellable`, altrimenti `UrlTree` verso la route `orders` (integration-libs/s4-service/order/components/guards/cancel-service-order.guard.ts:L21-L37); `ServiceOrderGuard` passa solo se `serviceReschedulable`, altrimenti messaggio `rescheduleService.serviceNotReschedulable` e `false` (integration-libs/s4-service/order/components/guards/service-order.guard.ts:L20-L34).
- `CancelServiceOrderComponent` invia **tutte** le entry dell'ordine (`orderEntryNumber`, `quantity`) più il motivo, poi rinaviga ai dettagli con un messaggio (integration-libs/s4-service/order/components/cancel-service-order/cancel-service-order.component.ts:L59-L102); `RescheduleServiceOrderComponent` filtra le entry ai soli prodotti SERVICE e chiama `rescheduleService(orderCode, dateTime)` (integration-libs/s4-service/order/components/reschedule-service-order/reschedule-service-order.component.ts:L77-L85, integration-libs/s4-service/order/components/reschedule-service-order/reschedule-service-order.component.ts:L136-L163).

L'entry point `@spartacus/s4-service/order` esporta solo module e components (non core/occ), mentre `@spartacus/s4-service/checkout` esporta anche core e occ (integration-libs/s4-service/order/public_api.ts:L7-L8, integration-libs/s4-service/checkout/public_api.ts:L7-L10).

3.8 Punchout — parte root (eager)

Punchout è l'unica libreria del gruppo con una feature lazy "vera" (`punchout`) e con una parte eager corposa. `PunchoutRootModule` (integration-libs/punchout/root/punchout.root.module.ts:L36-L84):

1. importa `PunchoutNavigationModule`, che contribuisce `PunchoutNavigationGuard` al multi-token `BEFORE_CMS_PAGE_GUARD` (integration-libs/punchout/root/guards/punchout-navigation.module.ts:L12-L21);
2. fornisce tre default config (componenti CMS, routing, guard) e un `provideAppInitializer` che chiama `PunchoutStatePersistenceService.initSync()` (integration-libs/punchout/root/punchout.root.module.ts:L39-L47);
3. registra `PunchoutCartInterceptor` con `useExisting` nel multi-token `HTTP_INTERCEPTORS` (integration-libs/punchout/root/interceptors/index.ts:L11-L17);
4. sostituisce `AuthHttpHeaderService` con una **factory** dipendente dal feature toggle (§3.10) (integration-libs/punchout/root/punchout.root.module.ts:L49-L66);
5. contribuisce `PunchoutExpiredRefreshTokenHandler` a `EXPIRED_REFRESH_TOKEN_HANDLERS` (integration-libs/punchout/root/punchout.root.module.ts:L67-L71);

6. fornisce `PunchoutUiRestrictionService` e lo aggancia al componente root via `APP_BOOTSTRAP_LISTENER` (integration-libs/punchout/root/punchout.root.module.ts:L72-L81).

Stato. `PunchoutStoreService` è uno store minimale su `BehaviorSubject<PunchoutState>`: `setPunchoutState` sostituisce, `updatePunchoutState` fonde, `clearPunchoutState` ripristina lo stato iniziale (integration-libs/punchout/root/services/punchout-store.service.ts:L11-L49). Niente NgRx.

Persistenza. `initSync` sincronizza con `localStorage` (default di `syncWithStorage`), sotto la chiave `spartacus⚿⚿punchout` (key `punchout`, contesto vuoto), **solo** il `punchoutSessionId` (integration-libs/punchout/root/services/punchout-state-persistence.service.ts:L23-L36, core-libs/core/src/state/services/state-persistence.service.ts:L44-L48, core-libs/core/src/state/services/state-persistence.service.ts:L124-L130). Dopo l'avvio, un id assente è scritto come `''` e non `undefined`, perché `StatePersistenceService` non permette di cancellare una chiave già creata (integration-libs/punchout/root/services/punchout-state-persistence.service.ts:L42-L53). Al refresh, `onRead` ripristina l'id e richiama `initPunchoutSession({ isPageRefresh: true })`, salvo che si sia già sulla pagina di sessione (integration-libs/punchout/root/services/punchout-state-persistence.service.ts:L62-L76).

Rilevamento. `isPunchoutSessionPage` è vero se il path corrente (prima di `?`) contiene il path semantico `punchoutSession` **e** c'è una query string (integration-libs/punchout/root/services/punchout-detection.service.ts:L23-L34). Le route `punchout/cxml/session` e `punchout/cxml/error` sono `protected: false` e `authFlow: true` (integration-libs/punchout/root/config/default-punchout-routing-config.ts:L9-L30).

Interceptor. `PunchoutCartInterceptor` aggiunge l'header `punchoutsid` a ogni richiesta il cui URL contiene `/carts/{punchoutCartId}` (integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts:L17-L49).

Guard di navigazione. `PunchoutNavigationGuard.canActivate` calcola `cxRoute` (da `route.data.cxRoute` o dal path con `RoutingConfigService.getRouteName`), ricava l'operazione punchout e, se la route non è ammessa né per cxRoute né per URL, mostra un warning e naviga a `redirectPage` (integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L36-L61). L'operazione è determinata solo per utente loggato con `punchoutSessionId`; se la sessione non è in stato la richiede via facade; ogni errore → `undefined`, cioè nessuna restrizione (integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L96-L131). In `isAllowedUrls`, `'/'` significa "solo home" (route con `url.length === 0`), gli altri URL sono confrontati con `includes` sul path relativo (integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L63-L82). Default: INSPECT ammette solo session/requisition/inspect e rimanda a `punchoutInspect`; EDIT/CREATE ammettono home + category, brand, quickOrder, product, cart, search, punchoutError e rimandano a `home` (integration-libs/punchout/root/config/default-punchout-guard-config.ts:L10-L46).

UI restriction. `PunchoutUiRestrictionService.isPunchoutSessionActive` aggiunge/rimuove la classe CSS `cxPunchoutSessionActive` sull'elemento root (integration-libs/punchout/root/services/punchout-ui-restriction.service.ts:L24-L58); gli stili con quella classe nascondono il bottone secondario del dialog "added to cart" e il bottone recensioni (integration-libs/punchout/styles/_punchout-template.scss:L49-L59).

3.9 Punchout — sessione e requisition (lazy)

`PunchoutFacade` è una facade lazy (`facadeFactory`, feature `punchout`) con 7 metodi (integration-libs/punchout/root/facade/punchout.facade.ts:L17-L37); la feature è mappata ai 6 componenti CMS punchout (integration-libs/punchout/root/config/default-punchout-cms-component-config.ts:L10-L23) e l'app demo aggiunge il loader (projects/storefrontapp/src/app/spartacus/features/punchout/punchout-feature.module.ts:L33-L40). `PunchoutModule` = components + core + occ; core lega `PunchoutFacade → useExisting PunchoutService`; occ lega `PunchoutAdapter → OccPunchoutAdapter` (integration-libs/punchout/punchout.module.ts:L12-L15, integration-libs/punchout/core/facade/facade-providers.ts:L11-L17, integration-libs/punchout/occ/punchout-occ.module.ts:L15-L25). Endpoint: `punchout/sessions/${sessionId}` e `.../`

`requisition`, quest'ultimo con `discardCartEntries=true` se richiesto (`integration-libs/punchout/occ/config/default-occ-punchout-config.ts:L10-L13`, `integration-libs/punchout/occ/adapters/occ-punchout.adapter.ts:L49-L69`).

Avvio sessione:

1. `PunchoutSessionComponent` (template vuoto) legge il query param `sid` e chiama `initPunchoutSession` (`integration-libs/punchout/components/punchout-session/punchout-session.component.ts:L17-L51`).
2. `getPunchoutSessionCommand`: senza id termina la sessione; altrimenti logout silenzioso → richiesta sessione OCC → login con il token ricevuto → load del carrello → stato → (solo EDIT, non refresh) snapshot della requisition iniziale → (non refresh) routing alla pagina target (`integration-libs/punchout/core/facade/punchout.service.ts:L69-L109`).
3. `requestPunchoutSessionCommand` mette in cache in memoria la prima risposta valida e la riusa; senza `accessToken`, `customerId` o `cartId` lancia `Error('Punchout login info missing')` (`integration-libs/punchout/core/facade/punchout.service.ts:L241-L261`).
4. `PunchoutAuthService.loginWithToken` scrive direttamente `access_token`, `access_token_stored_at`, `token_type` in `AuthStorageService`, imposta `OCC_USER_ID_CURRENT` e dispatcha `AuthActions.Login` (`integration-libs/punchout/core/services/punchout-auth.service.ts:L64-L75`). Non passa quindi per il flusso OAuth descritto in 22 — Autenticazione.
5. `routeToTargetPage`: CREATE/EDIT a livello PRODUCT con `selectedItem` → PDP; EDIT → cart; INSPECT → `punchoutInspect`; altrimenti `/` (`integration-libs/punchout/core/facade/punchout.service.ts:L298-L320`).

Ritorno al procurement:

1. I bottoni chiamano `submitRequisition(cancel)` o `closePunchoutSession()` (`integration-libs/punchout/components/punchout-buttons/punchout-buttons.component.ts:L36-L38`, `integration-libs/punchout/components/punchout-close-session/punchout-close-session.component.ts:L29-L31`).
2. `closePunchoutSessionCommand`: CREATE → `cancelRequisition: true` (si invia un carrello vuoto); EDIT → `closePunchoutSession: true` (si invia lo snapshot iniziale); poi apre la pagina requisition (`integration-libs/punchout/core/facade/punchout.service.ts:L194-L232`).
3. `getPunchoutRequisitionCommand` blocca la navigazione manuale alla pagina requisition (nessun flag in stato → warning + home), restituisce lo snapshot iniziale se si chiude una sessione EDIT, altrimenti chiama OCC (`integration-libs/punchout/core/facade/punchout.service.ts:L120-L171`).
4. `PunchoutRequisitionComponent` riempie un form nascosto `cxml-base64` con `orderAsCXML`, attende un tick (`timer(0)`), fa logout del punchout user e invia il form in POST a `browseFormPostUrl`, cioè al sistema di procurement (`integration-libs/punchout/components/punchout-requisition/punchout-requisition.component.ts:L42-L95`, `integration-libs/punchout/components/punchout-requisition/punchout-requisition.component.html:L1-L14`).

`endPunchoutSession` (auth service) pulisce lo stato, mostra `httpHandlers.unauthorized.common`, fa `coreLogout` e naviga a `punchoutError` (`integration-libs/punchout/core/services/punchout-auth.service.ts:L81-L97`).

3.10 Punchout e il refresh token scaduto: da override a handler

Questo è l'esempio più istruttivo del capitolo su come il core sia evoluto per **ridurre** gli override DI concorrenti.

- **Percorso legacy (toggle spento).** `PunchoutAuthHttpHeaderService` estende `AuthHttpHeaderService` e sovrascrive `handleExpiredRefreshToken`: sulla pagina di sessione fa logout silenzioso, con sessione attiva la termina, altrimenti `super` (`integration-libs/punchout/root/services/punchout-auth-http-header.`

service.ts:L24-L66). Il problema: anche ASM sostituisce `AuthHttpHeaderService` (feature-libs/asm/root/asm-root.module.ts:L73-L83), e un token può avere un solo vincitore.

- **Percorso nuovo (toggle acceso).** `AuthHttpHeaderService.handleExpiredRefreshToken` interroga in sequenza gli handler del multi-token `EXPIRED_REFRESH_TOKEN_HANDLERS` e usa il fallback (logout + login + messaggio) solo se nessuno ha gestito il caso (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L285). Il commento core lo motiva esplicitamente: evitare di sovrascrivere l'intero servizio, "which would create a conflict with other service overrides (e.g., ASM)" (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L311-L316). `PunchoutExpiredRefreshTokenHandler` riproduce la logica punchout e ritorna `of(false)` se non applicabile (integration-libs/punchout/root/services/punchout-expired-refresh-token-handler.service.ts:L31-L45).
- **Ponte di compatibilità.** Il provider factory di Punchout per `AuthHttpHeaderService` sceglie: con toggle attivo `DELEGATED_AUTH_HTTP_HEADER_SERVICE` (fornito da ASM) o, in sua assenza, `DEFAULT_AUTH_HTTP_HEADER_SERVICE`; con toggle spento `PunchoutAuthHttpHeaderService` (integration-libs/punchout/root/punchout.root.module.ts:L49-L66). `DEFAULT_AUTH_HTTP_HEADER_SERVICE` crea un `AuthHttpHeaderService` "pulito" con `new`; entrambi i token sono dichiarati temporanei (fix CXPSA-12514) (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L50-L70); `DELEGATED_AUTH_HTTP_HEADER_SERVICE` è un hook senza factory perché Punchout possa delegare all'override ASM senza importarlo (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L72-L82). ASM fornisce entrambi i token (feature-libs/asm/root/asm-root.module.ts:L73-L83).
- Il toggle `enableExpiredRefreshTokenHandlers` vale `true` nei default ed è attivo anche nell'app demo (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1072-L1072, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L360-L360). Vedi 42 — Feature toggle.

3.11 EPD Visualization (deprecata)

Tutta la libreria è `@deprecated since v221121.5.0` (integration-libs/epd-visualization/epd-visualization.module.ts:L12-L22). `EpdVisualizationRootModule` mappa la feature `epd-visualization` al componente CMS `VisualPickingTabComponent`, fornisce la default config e un config validator (integration-libs/epd-visualization/root/epd-visualization-root.module.ts:L17-L38).

Config. `EpdVisualizationConfig.epdVisualization` contiene `ui5.bootstrapUrl`, `apis.baseUrl`, `usageIds` (folder/product) e `visualPicking.productReferenceType` (integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L19-L104). I default coprono solo usage id (`CommerceCloud-Folder`, `CommerceCloud-SparePart/ProductCode`) e `productReferenceType: 'SPAREPART'`, **non** `apis` né `ui5` (integration-libs/epd-visualization/root/config/epd-visualization-default-config.ts:L12-L35); l'app demo li fornisce (projects/storefrontapp/src/app/spartacus/features/epd-visualization/epd-visualization-feature.module.ts:L24-L31). Il validator controlla presenza e protocollo http/https delle URL, gli usage id e `productReferenceType` (integration-libs/epd-visualization/root/config/epd-visualization-config-validator.ts:L16-L39), ma i validator girano solo in dev mode e producono soltanto `logger.warn` (core-libs/core/src/config/config-validator/config-validator.module.ts:L28-L35, core-libs/core/src/config/config-validator/config-validator.ts:L42-L47).

API. `EpdVisualizationApiModule` lega gli adapter astratti core a implementazioni REST v1 (`SceneAdapter → StorageV1Adapter`, `VisualizationAdapter → VisualizationV1Adapter`), che non sono nella public API (integration-libs/epd-visualization/epd-visualization-api/epd-visualization-api.module.ts:L18-L24, integration-libs/epd-visualization/epd-visualization-api/public_api.ts:L7-L10). `StorageV1Adapter.getNodes` chiama `${baseUrl}/vis/public/storage/v1/scenes/{sceneId}/nodes` (integration-libs/epd-visualization/epd-visualization-api/adapters/storage-v1/storage-v1.adapter.ts:L49-L108); `VisualizationV1Adapter.lookupVisualization` chiama `.../visualization/v1/lookup/visualization?usage=…`

`&folderUsageId=…` con usage id serializzati in JSON e URI-encoded (integration-libs/epd-visualization/epd-visualization-api/adapters/visualization-v1/visualization-v1.adapter.ts:L50-L68).

Core. `VisualizationLookupService.findMatchingVisualizations` costruisce lo usage id del prodotto e tiene solo visualizzazioni `3DModel` o `2DDrawing` (integration-libs/epd-visualization/core/services/visualization-lookup/visualization-lookup.service.ts:L39-L69). `SceneNodeToProductLookupService.populateMapsForScene` scarica i nodi della scena e popola due mappe bidirezionali nodeId ↔ productCodes in `BehaviorSubject` (integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L52-L102), con varianti sincrone che ritornano `[]` finché le mappe non sono popolate (integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L184-L234).

Componenti e UI5.

1. `VisualPickingTabComponent` ottiene `VisualViewerService` e `VisualPickingProductListService` dai figli con `@ViewChild(..., { read: Service })` e li passa a `VisualPickingTabService.initialize` in `ngAfterViewInit` (integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.component.ts:L36-L59).
2. `initialize` (solo browser) collega la lista filtrata a `includedProductCodes` del viewer e, quando arrivano le product references, chiama `loadVisualization(currentProduct.code)` (integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.service.ts:L46-L89).
3. Le references sono caricate per il tipo `visualPicking.productReferenceType` via `ProductReferenceService` (integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/visual-picking-product-list.service.ts:L109-L128).
4. `VisualViewerComponent` fornisce un'istanza propria di `VisualViewerService` e tutti i suoi `@Input/@Output` sono proxy verso il servizio (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.component.ts:L27-L50).
5. Il costruttore di `VisualViewerService` in SSR non fa nulla; in browser concatena bootstrap UI5 → init librerie → creazione viewport (`shareReplay`) (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L78-L99). `bootstrapUi5` inietta uno `<script>` in `<head>` con `src = ui5.bootstrapUrl` e callback globale `document.onUi5Bootstrapped`, saltando se UI5 è già presente (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1105-L1140); `initializeUi5` carica `sap.m`, `sap.ui.layout`, `sap.ui.vk`, `sap.ui.richtexteditor` (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1142-L1158); `addViewport` crea Viewport e collaboratori via `sap.ui.require` e lo piazza sull'elemento host (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1304-L1389).
6. `loadVisualization` risolve la visualizzazione (0 / 1 / >1 match → `NoMatchFound`, `UniqueMatchFound`, `MultipleMatchesFound`), popola le mappe nodi-prodotti e carica la scena; in SSR ritorna subito un `UnexpectedError` (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1032-L1092, integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1402-L1449). `loadScene` crea un `ContentResource` con `source: ${baseUrl}/vis/public/storage/v1` e `useSecureConnection: false` (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1549-L1565).

`CompactAddToCartComponent` estende `AddToCartComponent` del secondary entry `@spartacus/cart/base/components/add-to-cart` cambiando solo il template (integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/compact-add-to-cart/compact-add-to-cart.component.ts:L10-L29).

3.12 Packaging: schematics, stili, test

Schematics `ng-add`**.** Le sei factory hanno lo stesso corpo: `validateSpartacusInstallation`, `analyzeCrossFeatureDependencies`, poi la catena `analyzeApplication` → `addFeatures` → `addPackageJsonDependenciesForLibrary(peerDependencies)` → `finalizeInstallation` (integration-libs/omf/schematics/add-omf/index.ts:L25-L43, integration-libs/s4om/schematics/add-s4om/index.ts:L25-L43, integration-libs/cpq-quote/schematics/add-cpq-quote/index.ts:L25-L43, integration-libs/punchout/schematics/add-punchout/index.ts:L25-L43, integration-libs/epd-visualization/schematics/add-epd-visualization/index.ts:L25-L48). La conoscenza specifica (dipendenze, root module, `importAfter`) sta in `core-libs/schematics`, ad es. `S4OM_SCHEMATICS_CONFIG` (core-libs/schematics/src/shared/lib-configs/integration-libs/s4om-schematics-config.ts:L28-L59); per questo `integration-libs/s4om/schematics/constants.ts` è un doppione inerte, con nomi che divergono (`S4OM_MODULE_NAME` vs `S4OM_FEATURE_MODULE_NAME`) (integration-libs/s4om/schematics/constants.ts:L9-L17, core-libs/schematics/src/shared/lib-configs/integration-libs/s4om-schematics-config.ts:L19-L26). Lo spec punchout copre solo il caso senza feature (integration-libs/punchout/schematics/add-punchout/index_spec.ts:L31-L45).

Stili. OMF e CPQ non hanno Sass. Punchout, s4-service ed EPD seguono lo schema placeholder + allowlist + `$skipComponentStyles`; punchout aggiunge un'allowlist di page template che estende `%PunchoutTemplate` su `cx-page-layout.PunchoutTemplate` (integration-libs/punchout/_index.scss:L7-L40, integration-libs/epd-visualization/_index.scss:L5-L29). In s4-service i partial di reschedule e detail actions definiscono classi globali anziché placeholder, quindi sono sempre applicati e non disattivabili via allowlist (integration-libs/s4-service/styles/components/_reschedule-service-order.scss:L1-L34, integration-libs/s4-service/styles/components/_s4-service-order-detail-actions.scss:L1-L7); lo stesso vale per integration-libs/s4om/styles/schedule-lines.scss:L1-L27.

Test. OMF, S4OM, CPQ e s4-service usano Vitest; punchout ed EPD restano su Karma (integration-libs/punchout/project.json:L20-L26) con un `test.ts` custom che aggiunge `provideZoneChangeDetection()` per evitare gli NG0100 introdotti da Angular 21 (integration-libs/punchout/test.ts:L14-L30, integration-libs/epd-visualization/test.ts:L16-L32). Le soglie di coverage differiscono: 90/90/75/90 per OMF e S4OM, 85/85/65/85 per CPQ e s4-service; in CPQ la coverage non ha `enabled: true` e la costante `root` è dichiarata ma non usata (integration-libs/omf/vitest.config.ts:L24-L46, integration-libs/s4om/vitest.config.ts:L42-L47, integration-libs/cpq-quote/vitest.config.ts:L11-L11, integration-libs/cpq-quote/vitest.config.ts:L26-L42, integration-libs/s4-service/vitest.config.ts:L43-L48).

---

4. Diagrammi

4.1 Dove vive ogni override (injector root vs chunk lazy)

■ **TAVOLA 36.1** diagramma a tutta pagina, in fondo al capitolo

Le frecce dentro i chunk rappresentano l'ordine di import nel wrapper (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27, projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L16-L33, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25): a parità di token vince il modulo più a destra/in basso.

4.2 OMF: dal link dello storico al dettaglio con header guid

■ TAVOLA 36.2 diagramma a tutta pagina, in fondo al capitolo

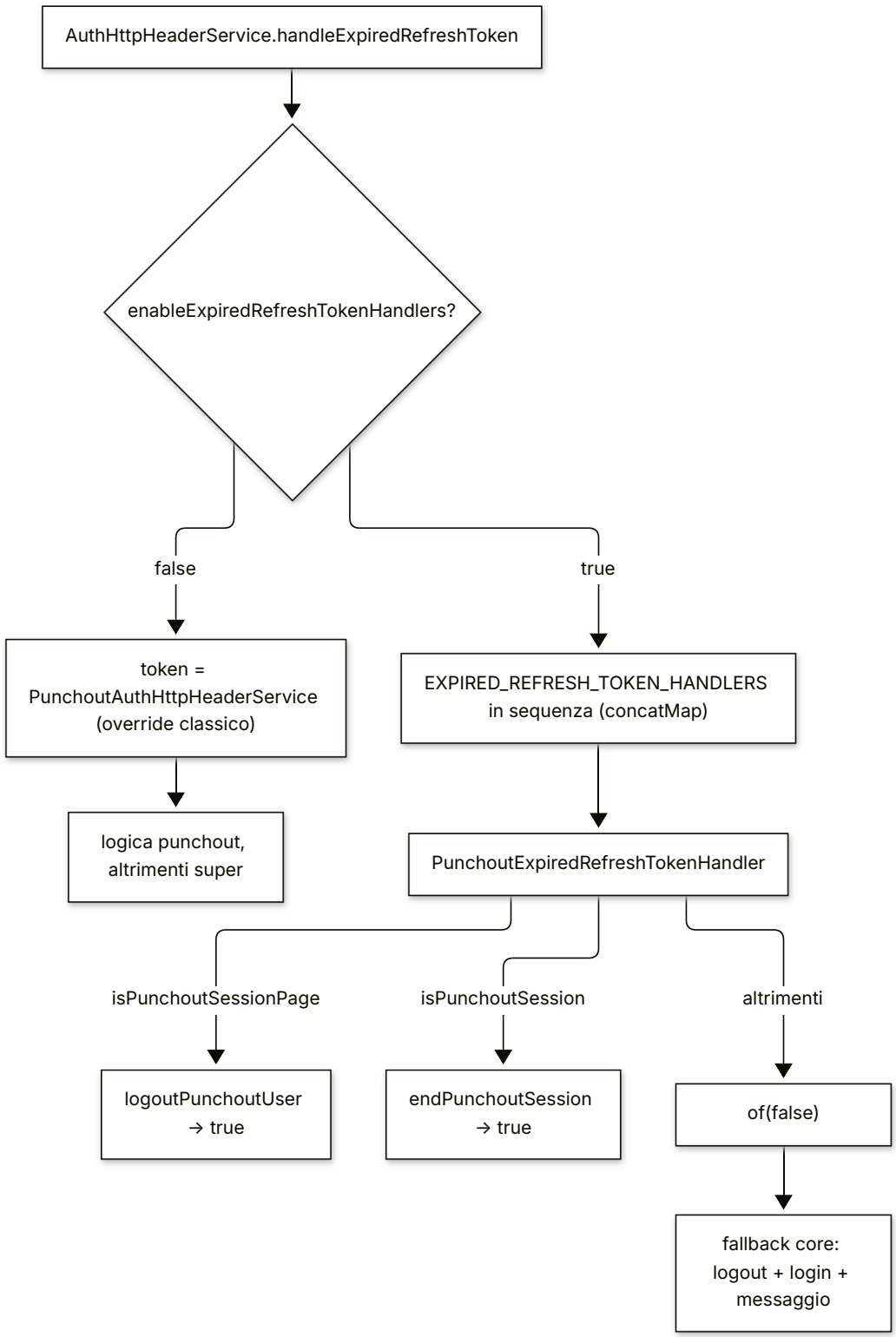
4.3 S/4 Service: step "service details" del checkout

■ TAVOLA 36.3 diagramma a tutta pagina, in fondo al capitolo

4.4 Punchout: ciclo di vita della sessione

■ TAVOLA 36.4 diagramma a tutta pagina, in fondo al capitolo

4.5 Refresh token scaduto: handler vs override



4.6 EPD: visual picking

■ **TAVOLA 36.5** diagramma a tutta pagina, in fondo al capitolo

5. Estratti di codice reale

5.1 Ordine degli import nel wrapper lazy order

projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27

```ts
const extensions: Type<any>[] = [];
if (environment.s4Service) {
  extensions.push(S4ServiceOrderModule);
}
if (environment.omf) {
  extensions.push(OmfOrderModule);
}
if (environment.opf) {
  extensions.push(OpfOrderModule);
}
@NgModule({
  imports: [OrderModule, ...extensions],
})
export class OrderWrapperModule {}
```

5.2 OMF: doppio override DI

integration-libs/omf/order/omf-order.module.ts:L15-L26

```ts
@NgModule({
  providers: [
    OmfOrderHistoryService,
    provideDefaultConfig(defaultOmfConfig),
    { provide: OrderHistoryAdapter, useClass: OccOmfOrderHistoryAdapter },
    {
      provide: OrderHistoryFacade,
      useClass: OmfOrderHistoryService,
    },
  ],
})
export class OmfOrderModule {}
```

5.3 OMF: guid da query param o dallo store, e header

integration-libs/omf/order/occ-omf-order-history.adapter.ts:L39-L67

```ts
  getOrderGuid(orderCode: string): Observable<string | undefined> {
    return this.route.queryParams.pipe(
      switchMap((queryParams) => {
        if (queryParams.guid) {
          // when navigating from Order History to Order Details page
```

```
          return of(queryParams.guid);
        } else {
          // when loading Order History page in case of My-Account-V2
          return this.getOrderDetailsValue(orderCode).pipe(
            map((order) => (order ? order.guid : undefined))
          );
        }
      })
    );
  }

  getRequestHeader(guid: string | undefined): HttpHeaders {
    let headers = new HttpHeaders();
    if (!this.config.omf?.guidHttpHeaderName && isDevMode()) {
      this.logger.warn(
        `There is no guidHttpHeaderName configured in OMF configuration`
      );
    }
    const guidHeader = this.config.omf?.guidHttpHeaderName?.toLowerCase?.();
    if (guid && guidHeader) {
      headers = headers.set(guidHeader, guid);
    }
    return headers;
  }
```

5.4 S4OM: outlet + override di servizio core

integration-libs/s4om/root/s4om-root.module.ts:L17-L35

TS

```
@NgModule({
  imports: [
    ScheduleLinesModule,
    RequestedDeliveryDateComponentsModule, //Adding dependency with Requested Delivery Date so that the library gets installed along with S4OM
    PDFInvoicesComponentsModule, //Adding dependency with PDF Invoices so that the library gets installed along with S4OM
  ],
  providers: [
    provideOutlet({
      id: CartOutlets.ITEM_DETAILS,
      position: OutletPosition.AFTER,
      component: ScheduleLinesComponent,
    }),
    {
      provide: ProductCatalogService,
      useExisting: S4omProductCatalogService,
    },
```

```
  ],
})
export class S4omRootModule {}
```

integration-libs/s4om/root/services/s4om-product-catalog.service.ts:L23-L37

```ts
  isProductInCatalog(product?: Product): boolean {
    if (!product) {
      return false;
    }

    const stockLevelStatus = product?.stock?.stockLevelStatus;

    if (!stockLevelStatus) {
      return true;
    }

    return (
      stockLevelStatus !== this.PRODUCT_NOT_IN_CATALOGUE_STOCK_LEVEL_STATUS
    );
  }
```

5.5 CPQ: il flag condiviso

integration-libs/cpq-quote/cpq-quote-discount/cpq-qute.service.ts:L13-L23

```ts
export class CpqQuoteService {
  private isFlagSubject = new BehaviorSubject<boolean>(true); // Default value is true
  isFlag$ = this.isFlagSubject.asObservable();

  setFlag(value: boolean) {
    this.isFlagSubject.next(value);
  }
  getFlag$() {
    return this.isFlagSubject.getValue();
  }
}
```

integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-heading/cpq-quote-heading.component.ts:L55-L63

```ts
    if (this.outlet?.context$) {
      this.subscription.add(
        this.outlet.context$.subscribe((context) => {
          this.datacheck = context;
          const hasDiscounts = this.datacheck.some(
            (item: any) => item.cpqDiscounts && item.cpqDiscounts.length > 0
```

```
        );
        // Set flag based on the existence of cpqDiscounts in any item
        this.cpqQuoteService.setFlag(!hasDiscounts);
```

integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L39-L46

```ts
  providers: [
    CpqQuoteService,
    { provide: CartItemListComponentService, useClass: CpqQuoteSharedService },
    provideOutlet({
      id: CartOutlets.CPQ_QUOTE_MODULE,
      position: OutletPosition.AFTER,
      component: CpqQuoteDiscountComponent,
    }),
```

5.6 s4-service: estensione dei `cmsComponents` e sostituzione di `CheckoutConfig`

integration-libs/s4-service/root/s4-service-root.module.ts:L31-L55

```ts
export const S4_SERVICE_CMS_COMPONENTS: string[] = [
  ...CHECKOUT_B2B_CMS_COMPONENTS,
  'CheckoutServiceDetails',
];
export const S4_SERVICE_ORDER_CMS_COMPONENTS: string[] = [
  'RescheduleServiceOrder',
  'CancelServiceOrderHeadline',
  'CancelServiceOrder',
];
export function defaultS4ServiceComponentsConfig() {
  const config: CmsConfig = {
    featureModules: {
      [CHECKOUT_FEATURE]: {
        cmsComponents: S4_SERVICE_CMS_COMPONENTS,
      },
      [ORDER_FEATURE]: {
        cmsComponents: [
          ...ORDER_CMS_COMPONENTS,
          ...S4_SERVICE_ORDER_CMS_COMPONENTS,
        ],
      },
    },
  };
  return config;
}
```

integration-libs/s4-service/root/s4-service-root.module.ts:L74-L81

```ts
  providers: [
    { provide: CheckoutConfig, useValue: defaultServiceDetailsCheckoutConfig },
    provideDefaultConfig(defaultServiceOrdersRoutingConfig),
    provideDefaultConfig(defaultServiceDeliveryModeConfig),
    provideDefaultConfigFactory(defaultS4ServiceComponentsConfig),
    CxDatePipe,
    CheckoutServiceSchedulePickerService,
  ],
```

5.7 s4-service: estendere un enum del core anche a runtime

integration-libs/s4-service/root/model/augmented-types.model.ts:L38-L47

```ts
declare module '@spartacus/checkout/base/root' {
  enum CheckoutStepType {
    SERVICE_DETAILS = 'serviceDetails',
  }
  interface CheckoutState {
    servicedAt?: ServiceDateTime; //response property name
  }
}

(CheckoutStepType as any)['SERVICE_DETAILS'] = 'serviceDetails';
```

5.8 s4-service: guard degli step

integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts:L26-L43

```ts
  canActivate(route: ActivatedRouteSnapshot): Observable<GuardResult> {
    return combineLatest(
      this.checkoutServiceDetailsFacade.hasServiceItems(),
      this.checkoutServiceDetailsFacade.hasNonServiceItems()
    ).pipe(
      switchMap(([hasServiceItems, hasNonServiceItems]) => {
        this.checkoutStepService.disableEnableStep(
          CheckoutStepType.SERVICE_DETAILS,
          !hasServiceItems
        );
        this.checkoutStepService.disableEnableStep(
          CheckoutStepType.DELIVERY_MODE,
          !hasNonServiceItems
        );
        return super.canActivate(route);
      })
    );
  }
```

integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts:L62-L76

```ts
setServiceDeliveryMode(): Observable<unknown> {
  return combineLatest([
    this.checkoutServiceDetailsFacade.hasServiceItems(),
    this.checkoutServiceDetailsFacade.hasNonServiceItems(),
  ]).pipe(
    switchMap(([hasServiceItems, hasNonServiceItems]) => {
      if (!hasNonServiceItems && hasServiceItems) {
        return this.checkoutDeliveryModesFacade.setDeliveryMode(
          this.config.s4ServiceDeliveryMode?.code ?? ''
        );
      }
      return of(undefined);
    })
  );
}
```

5.9 s4-service: Command e precondizioni

integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L39-L85

```ts
protected setServiceScheduleSlotCommand: Command<ServiceDateTime> =
  this.commandService.create<ServiceDateTime>(
    (scheduledAt) =>
      this.checkoutPreconditions().pipe(
        switchMap(([userId, cartId]) =>
          this.serviceDetailsConnector
            .setServiceScheduleSlot(userId, cartId, {
              scheduledAt,
            })
            .pipe(
              tap(() =>
                this.eventService.dispatch(
                  {
                    userId,
                    cartId,
                    scheduledAt,
                  },
                  CheckoutServiceDetailsSetEvent
                )
              )
            )
        )
      ),
    {
      strategy: CommandStrategy.CancelPrevious,
    }
```

```
  );

protected checkoutPreconditions(): Observable<[string, string]> {
  return combineLatest([
    this.userIdService.takeUserId(),
    this.activeCartFacade.takeActiveCartId(),
    this.activeCartFacade.isGuestCart(),
  ]).pipe(
    take(1),
    map(([userId, cartId, isGuestCart]) => {
      if (
        !userId ||
        !cartId ||
        (userId === OCC_USER_ID_ANONYMOUS && !isGuestCart)
      ) {
        throw new Error('Checkout conditions not met');
      }
      return [userId, cartId];
    })
  );
}
```

5.10 Punchout: provider del root module

integration-libs/punchout/root/punchout.root.module.ts:L42-L81

TS

```
  provideAppInitializer(() => {
    const punchoutPersistenceService = inject(
      PunchoutStatePersistenceService
    );
    punchoutPersistenceService.initSync();
  }),
  ...interceptors,
  // below provider should be removed after enableExpiredRefreshTokenHandlers is fully rolled out
  {
    provide: AuthHttpHeaderService,
    useFactory: () => {
      const featureToggles = inject(FeatureToggles);
      const delegatedAuthHttpHeaderService = inject(
        DELEGATED_AUTH_HTTP_HEADER_SERVICE,
        {
          optional: true,
        }
      );
      // delegatedService is expected to be used by ASM to address breaking changes of CXSPA-12514
      return featureToggles.enableExpiredRefreshTokenHandlers
        ? (delegatedAuthHttpHeaderService ??
            inject(DEFAULT_AUTH_HTTP_HEADER_SERVICE))
```

```
        : inject(PunchoutAuthHttpHeaderService);
    },
  },
  {
    provide: EXPIRED_REFRESH_TOKEN_HANDLERS,
    useExisting: PunchoutExpiredRefreshTokenHandler,
    multi: true,
  },
  PunchoutUiRestrictionService,
  {
    provide: APP_BOOTSTRAP_LISTENER,
    multi: true,
    useFactory: (): ((compRef: ComponentRef<any>) => void) => {
      const punchoutComponentsService = inject(PunchoutUiRestrictionService);
      return (compRef: ComponentRef<any>) =>
        punchoutComponentsService.init(compRef);
    },
  },
```

5.11 Punchout: handler del refresh token

integration-libs/punchout/root/services/punchout-expired-refresh-token-handler.service.ts:L31-L45

```ts
public handleExpiredRefreshTokenIfApplicable(): Observable<boolean> {
  if (this.punchoutDetectionService.isPunchoutSessionPage()) {
    return this.punchoutFacade.logoutPunchoutUser().pipe(
      map(() => true),
      defaultIfEmpty(true)
    );
  }
  if (this.punchoutDetectionService.isPunchoutSession()) {
    return this.punchoutFacade.endPunchoutSession().pipe(
      map(() => true),
      defaultIfEmpty(true)
    );
  }
  return of(false);
}
```

5.12 Punchout: interceptor del carrello

integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts:L27-L49

```ts
intercept(
  request: HttpRequest<any>,
  next: HttpHandler
): Observable<HttpEvent<any>> {
```

```
  return this.punchoutStoreService.getPunchoutState().pipe(
    take(1),
    switchMap((punchoutState) => {
      if (
        punchoutState?.punchoutSessionId &&
        punchoutState?.punchoutSession?.cartId &&
        request.url.includes(`/carts/${punchoutState.punchoutSession.cartId}`)
      ) {
        request = request.clone({
          headers: request.headers.append(
            this.PUNCHOUT_SESSION_ID_HEADER_KEY,
            punchoutState.punchoutSessionId
          ),
        });
      }
      return next.handle(request);
    })
  );
}
```

5.13 Punchout: guard di navigazione

integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L36-L61

TS

```
canActivate(
  route: CmsActivatedRouteSnapshot,
  _state: RouterStateSnapshot
): Observable<GuardResult> {
  const relativeUrl = `${route.url.map((u) => u.path).join('/')}`;
  const cxRoute =
    route?.data?.cxRoute ??
    this.routingConfigService.getRouteName(relativeUrl);
  return this.getPunchoutOperation(cxRoute).pipe(
    map((punchoutOperation: PunchOutOperation | undefined) => {
      if (
        punchoutOperation &&
        !this.isAllowedCxRoute(cxRoute, punchoutOperation) &&
        !this.isAllowedUrls(route, punchoutOperation, relativeUrl)
      ) {
        this.handleWarning();
        this.routingService.go(
          this.config.punchoutNavigation?.[punchoutOperation]?.redirectPage
        );
        return false;
      }

      return true;
```

```
    })
  );
}
```

5.14 Punchout: cache della sessione

integration-libs/punchout/core/facade/punchout.service.ts:L241-L261

```ts
protected requestPunchoutSessionCommand: Command<string, PunchoutSession> =
  this.commandService.create((punchoutSessionId: string) => {
    return this.cachedPunchoutSessionResponse
      ? of(this.cachedPunchoutSessionResponse)
      : this.punchoutConnector.getPunchoutSession(punchoutSessionId).pipe(
          map((punchoutSession) => {
            if (
              !punchoutSession?.token?.accessToken ||
              !punchoutSession?.customerId ||
              !punchoutSession?.cartId
            ) {
              throw new Error('Punchout login info missing');
            }
            this.cachedPunchoutSessionResponse = {
              ...punchoutSession,
              token: { ...punchoutSession.token },
            };
            return punchoutSession;
          })
        );
  });
```

5.15 EPD: bootstrap di UI5 solo in browser

integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L86-L98

```ts
  if (!this.windowRef.isBrowser()) {
    return;
  }
  const ui5BootStrapped$: Observable<void> =
    this.bootstrapUi5('ui5bootstrap');
  const ui5Initialized$: Observable<void> = ui5BootStrapped$.pipe(
    mergeMap(this.initializeUi5.bind(this))
  );
  this.viewportAdded$ = ui5Initialized$.pipe(
    mergeMap(this.addViewport.bind(this)),
    shareReplay()
  );
  this.executeWhenSceneLoaded(this.setInitialPropertyValues.bind(this));
```

integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1117-L1138

```ts
const script = this.windowRef.document.createElement('script');
script.setAttribute('id', scriptElementId);
this.windowRef.document
  .getElementsByTagName('head')[0]
  .appendChild(script);

(this.windowRef.document as any).onUi5Bootstrapped = () => {
  subscriber.next();
  subscriber.complete();
};

script.onerror = (error: any) => {
  subscriber.error(error);
  subscriber.complete();
};

script.id = 'sap-ui-bootstrap';
script.type = 'text/javascript';
script.setAttribute('data-sap-ui-compatVersion', 'edge');
script.setAttribute('data-sap-ui-async', 'true');
script.setAttribute('data-sap-ui-onInit', 'document.onUi5Bootstrapped()');
script.src = ui5Config.bootstrapUrl;
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`omf.guidHttpHeaderName`	`string`	`'guid'`	integration-libs/omf/order/config/default-omf-config.ts:L9-L13	integration-libs/omf/order/occ-omf-order-history.adapter.ts:L57-L62
`checkout` (intero, via token `CheckoutConfig`)	`CheckoutConfig`	6 step con `serviceDetails`, `express: false`, `guest: false` — **sostituisce** il token	integration-libs/s4-service/root/config/default-service-details-checkout-config.ts:L13-L57	provider integration-libs/s4-service/root/s4-service-root.module.ts:L75-L75
`s4ServiceDeliveryMode`	`DeliveryMode`	code `service-delivery`, `USD0.00`	integration-libs/s4-service/root/config/default-service-delivery-mode-config.ts:L10-L22	integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts:L69-L71, integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.component.ts:L124-L126, integration-libs/s4-service/order/components/order-summary/service-order-overview-component.service.ts:L15-L19

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
routing.routes.checkoutServiceDetails / cancelServiceDetails / rescheduleServiceDetails	RoutingConfig	checkout/service-details, my-account/order/{cancel, reschedule}service/:orderCode	integration-libs/s4-service/root/config/default-service-orders-routing-config.ts:L9-L23	integration-libs/s4-service/root/s4-service-root.module.ts:L59-L72
backend.occ.endpoints.setServiceScheduleSlot / getCheckoutDetails	string	users/${userId}/carts/${cartId}/serviceOrder/serviceScheduleSlot; fields con servicedAt	integration-libs/s4-service/checkout/occ/config/default-occ-checkout-s4-service-config.ts:L10-L25	integration-libs/s4-service/checkout/occ/adapter/occ-checkout-service-details.adapter.ts:L32-L34
backend.occ.endpoints.cancelServiceOrder / rescheduleService	string	users/${userId}/orders/${code}/serviceOrder/{cancellation, serviceScheduleSlot}	integration-libs/s4-service/order/occ/config/default-occ-s4-service-config.ts:L10-L15	integration-libs/s4-service/order/occ/adapters/occ-cancel-service-order.adapter.ts:L25-L27, integration-libs/s4-service/order/occ/adapters/occ-reschedule-service-order.adapter.ts:L31-L33
BaseStore.serviceOrderConfiguration.{leadDays,serviceScheduleTimes}	dato backend (base site)	0 / []	augmentation integration-libs/s4-service/root/model/augmented-types.model.ts:L16-L21	integration-libs/s4-service/root/facade/checkout-service-schedule-picker.service.ts:L48-L61
featureModules[CHECKOUT_FEATURE/ORDER_FEATURE].cmsComponents	string[]	B2B + CheckoutServiceDetails; order + 3	integration-libs/s4-service/root/s4-service-root.module.ts:L40-L55	lazy loading CMS (12)
punchoutNavigation[CREATE/EDIT/INSPECT].{allowedUrls,allowedCxRoutes,redirectPage}	PunchoutNavigationGuardConfig	vedi §3.8	integration-libs/punchout/root/config/default-punchout-guard-config.ts:L22-L46	integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L53-L93
routing.routes.punchout{Session, Requisition,Inspect, Error}	RoutingConfig	punchout/cxml/*	integration-libs/punchout/root/config/default-punchout-routing-config.ts:L9-L30	integration-libs/punchout/root/services/punchout-detection.service.ts:L28-L28
backend.occ.endpoints.punchoutSession / punchoutSessionRequisition	string	punchout/sessions/${sessionId}[/requisition]	integration-libs/punchout/occ/config/default-occ-punchout-config.ts:L10-L13	integration-libs/punchout/occ/adapters/occ-punchout.adapter.ts:L37-L59
featureModules.punchout.cmsComponents	string[]	6 componenti	integration-libs/punchout/root/config/default-punchout-cms-component-config.ts:L10-L23	lazy loading
feature toggle enableExpiredRefreshTokenHandlers	boolean	true	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1072-L1072	integration-libs/punchout/root/punchout.root.module.ts:L61-L64, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L268
epdVisualization.ui5.bootstrapUrl	string	nessuno (obbligatorio)	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L51-L60	integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1105-L1138

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
epdVisualization.apis.baseUrl	string	nessuno	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L65-L72	integration-libs/epd-visualization/epd-visualization-api/adapters/storage-v1/storage-v1.adapter.ts:L49-L56, integration-libs/epd-visualization/epd-visualization-api/adapters/visualization-v1/visualization-v1.adapter.ts:L50-L57, integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1557-L1563
epdVisualization.usageIds.{folderUsageId,productUsageId}	UsageIdConfig	CommerceCloud-Folder; CommerceCloud-SparePart/ProductCode	integration-libs/epd-visualization/root/config/epd-visualization-default-config.ts:L15-L29	integration-libs/epd-visualization/core/services/visualization-lookup/visualization-lookup.service.ts:L42-L46, integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L39-L43
epdVisualization.visualPicking.productReferenceType	string	'SPAREPART'	integration-libs/epd-visualization/root/config/epd-visualization-default-config.ts:L30-L32	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/visual-picking-product-list.service.ts:L83-L89
chunk i18n	TranslationChunksConfig	s4Service, s4omScheduleLines, cpqQuote, punchout	integration-libs/s4-service/assets/translations/translations.ts:L9-L11, integration-libs/s4om/assets/translations/translations.ts:L9-L11, integration-libs/cpq-quote/assets/translations/translations.ts:L9-L11, integration-libs/punchout/assets/translations/translations.ts:L9-L11	feature module dell'app (37)
flag app demo CX_CPQ, CX_EPD_VISUALIZATION, CX_S4OM, CX_OMF, CX_S4_SERVICE, CX_PUNCHOUT	boolean (env)	false	projects/storefrontapp/src/environments/environment.ts:L30-L43	projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L133-L175 e wrapper module

Contenuto dei chunk: s4-service `['serviceOrderCheckout', 'cancelService', 'rescheduleService']` (integration-libs/s4-service/assets/translations/translations.ts:L9-L11), s4om `['s4omScheduleLines']` (integration-libs/s4om/assets/translations/translations.ts:L9-L11), cpq-quote `['cpqQuoteHeading', 'discountCaption']` (integration-libs/cpq-quote/assets/translations/translations.ts:L9-L11), punchout `['punchout']` (integration-libs/punchout/assets/translations/translations.ts:L9-L11).

7. Commenti e note notevoli dal sorgente

- *"below provider should be removed after enableExpiredRefreshTokenHandlers is fully rolled out"* — il provider factory di `AuthHttpHeaderService` di Punchout è dichiaratamente transitorio (integration-libs/punchout/root/punchout.root.module.ts:L49-L49).

- *"delegatedService is expected to be used by ASM to address breaking changes of CXSPA-12514"* (integration-libs/punchout/root/punchout.root.module.ts:L60-L60).
- *"This approach allows features like Punchout to handle expired tokens without needing to override the entire AuthHttpHeaderService via DI, which would create a conflict with other service overrides (e.g., ASM)."* — la motivazione degli handler nel core (core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L311-L316).
- *"It is a workaround to address CXSPA-9608 - Public pages not displayed when token is invalid. To be removed once CXSPA-9608 is closed."* — sull'handler punchout (integration-libs/punchout/root/services/punchout-expired-refresh-token-handler.service.ts:L28-L29) e identico sul servizio legacy (integration-libs/punchout/root/services/punchout-auth-http-header.service.ts:L53-L54).
- *"Note that StatePersistenceService does not allow to delete key/value once it has been created."* (integration-libs/punchout/root/services/punchout-state-persistence.service.ts:L50-L50).
- *"handle race condition with logout guard, it needs to be assessed before punchout session is requested"* (integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L110-L110).
- *"punchoutSession ajax request will always return same response during a punchout session."* — giustificazione della cache in memoria (integration-libs/punchout/core/facade/punchout.service.ts:L53-L57).
- *"timer(0) wait for next Javascript event loop, this way DOM native elements becomes accessible."* (integration-libs/punchout/components/punchout-requisition/punchout-requisition.component.ts:L85-L85).
- *"@deprecated since 221121.1 - Displaying the individual action buttons will depend on their boolean flags from API response."* — su `displayActions$` e `checkServiceStatus` (integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.component.ts:L49-L52, integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.component.ts:L60-L63).
- *"//check if we need to add serializer and normalizer."* — l'adapter checkout s4-service non converte il payload (integration-libs/s4-service/checkout/occ/adapter/occ-checkout-service-details.adapter.ts:L38-L38).
- *"//guards: inherited from standard config,"* — l'override CMS delle detail actions non ridichiara i guard (integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.module.ts:L33-L33).
- *"TODO:#9574 - we should be able to remove the spread of `CHECKOUT_BASE_CMS_COMPONENTS`."* — ereditato da `CHECKOUT_B2B_CMS_COMPONENTS`, usato da s4-service (feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L24-L27).
- *"Adding dependency with Requested Delivery Date so that the library gets installed along with S4OM"* (integration-libs/s4om/root/s4om-root.module.ts:L20-L21).
- *"@deprecated: Method doesn't pass facade method's requirements (returns void or Observable)"* — su `getQueryParams`, proprio il metodo su cui si appoggia OMF (feature-libs/order/root/facade/order-history.facade.ts:L145-L148).
- *"@deprecated since v221121.5.0 - The epd-visualization integration library will be removed in the future."* (integration-libs/epd-visualization/root/epd-visualization-root.module.ts:L28-L30).
- *"Not in Spartacus public API since the /storage/v1 prefix will be deprecated and relocated"* (integration-libs/epd-visualization/epd-visualization-api/public_api.ts:L9-L10) e analogo nell'adapter (integration-libs/epd-visualization/epd-visualization-api/adapters/storage-v1/storage-v1.adapter.ts:L28-L31).
- *"Important: Please check the End of Cloud Provisioning column..."* su `ui5.bootstrapUrl`: la versione UI5 va scelta tra quelle ancora servite (integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L52-L58).

- *"add $filter when usageId filtering available in backend"* (integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L80-L80).
- Commento di file errato *"Public API Surface of estimated-delivery-date"* nella public API di cpq-quote (integration-libs/cpq-quote/public_api.ts:L7-L9).

8. Extension point

EXTENSION POINT	COME SI USA	EXPORT	RICETTA
`OmfConfig` / `OmfConfiguration` (`guidHttpHeaderName`)	`provideConfig({ omf: { guidHttpHeaderName: '...' } })`	`OmfConfiguration` da `@spartacus/omf/root` (integration-libs/omf/root/public_api.ts:L7-L8); `OmfConfig` **non** è esportato da integration-libs/omf/order/public_api.ts:L7-L9	50 — Ricette config e DI
`OccOmfOrderHistoryAdapter`, `OmfOrderHistoryService`	sottoclasse + provider nel wrapper order, dopo `OmfOrderModule`	`@spartacus/omf/order` (integration-libs/omf/order/public_api.ts:L7-L9)	52 — Ricette OCC
`S4omProductCatalogService` (override di `ProductCatalogService`)	sottoclasse e `useExisting`	`@spartacus/s4om/root` (integration-libs/s4om/root/public_api.ts:L7-L12)	50 — Ricette config e DI
`ScheduleLinesComponent` / outlet `CartOutlets.ITEM_DETAILS`	`provideOutlet` con altro componente	`@spartacus/s4om/root`	51 — Ricette CMS, outlet, layout
`CpqQuoteService`, `CpqQuoteSharedService`, componenti CPQ; outlet `CPQ_QUOTE_MODULE/HEADING/CPQ_QUOTE`	override servizi, `provideOutlet`	`@spartacus/cpq-quote` (integration-libs/cpq-quote/cpq-quote-discount/public_api.ts:L7-L12); outlet in `@spartacus/cart/base/root` (feature-libs/cart/base/root/models/cart-outlets.model.ts:L25-L27)	51 — Ricette CMS, outlet, layout
`CheckoutServiceDetailsFacade` (lazy)	iniettare per leggere/scrivere lo slot	`@spartacus/s4-service/root` (integration-libs/s4-service/root/public_api.ts:L7-L12)	53 — Feature, cart/checkout
`CancelServiceOrderFacade`, `RescheduleServiceOrderFacade`	iniettabili solo nel chunk order	`@spartacus/s4-service/root`	50 — Ricette config e DI
`CheckoutServiceSchedulePickerService`	sottoclasse per formati data	`@spartacus/s4-service/root`	50 — Ricette config e DI
`S4ServiceDeliveryModeConfig`	`provideConfig({ s4ServiceDeliveryMode: {...} })`	`@spartacus/s4-service/root`	50 — Ricette config e DI
`CheckoutServiceDetailsSetEvent`	`EventService.get(...)`	`@spartacus/s4-service/root` (integration-libs/s4-service/root/events/checkout-service-details.events.ts:L18-L28)	50 — Ricette eventi

EXTENSION POINT	COME SI USA	EXPORT	RICETTA
`CheckoutServiceDetailsAdapter` / `OccCheckoutServiceDetailsAdapter`	adapter custom	`@spartacus/s4-service/checkout` (integration-libs/s4-service/checkout/public_api.ts:L7-L10)	52 — Ricette OCC
`CheckoutServiceOrderStepsSetGuard`	sottoclasse + `useExisting` su `CheckoutStepsSetGuard` nel wrapper checkout, dopo s4-service	`@spartacus/s4-service/checkout`	53 — Checkout step e guard
`Cancel/RescheduleServiceOrderAdapter`	**non esportati** (integration-libs/s4-service/order/public_api.ts:L7-L8)	—	54 — API pubblica vs interna
`ServiceOrderOverviewComponentService`, guard e componenti order	override servizio / mapping CMS	`@spartacus/s4-service/order`	51 — Ricette CMS, outlet, layout
`S4_SERVICE_CMS_COMPONENTS`, `S4_SERVICE_ORDER_CMS_COMPONENTS`	riuso quando si estendono di nuovo i `cmsComponents` di checkout/order	`@spartacus/s4-service/root`	53 — Feature lazy
`PunchoutNavigationGuardConfig`	`provideConfig({ punchoutNavigation: {...} })`	`@spartacus/punchout/root` (integration-libs/punchout/root/config/index.ts:L7-L10)	50 — Ricette config e DI
`PunchoutFacade`	facade lazy	`@spartacus/punchout/root`	50 — Ricette config e DI
Servizi punchout root (`PunchoutStoreService`, `PunchoutDetectionService`, `PunchoutUiRestrictionService`, `PunchoutStatePersistenceService`, `PunchoutExpiredRefreshTokenHandler`, `PunchoutAuthHttpHeaderService`)	sottoclasse + provider	`@spartacus/punchout/root` (integration-libs/punchout/root/public_api.ts:L7-L13)	50 — Ricette config e DI
`PunchoutNavigationGuard` (`BEFORE_CMS_PAGE_GUARD`)	multi provider aggiuntivo	`@spartacus/punchout/root`; `PunchoutCartInterceptor` **non** esportato	50 — Ricette interceptor
`PunchoutAdapter`, `PunchoutConnector`, `PUNCHOUT_SESSION_NORMALIZER`, `PUNCHOUT_REQUISITION_NORMALIZER`, `PunchoutService`, `PunchoutAuthService`	adapter/converter custom	`@spartacus/punchout/core`	52 — Ricette OCC
`EXPIRED_REFRESH_TOKEN_HANDLERS`	multi provider di un handler	`@spartacus/core` (core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts:L21-L23)	50 — Ricette config e DI
`EpdVisualizationConfig`	`provideConfig({ epdVisualization: {...} })`	`@spartacus/epd-visualization/root`	50 — Ricette config e DI
`SceneAdapter`, `VisualizationAdapter`, `NODES_RESPONSE_NORMALIZER`, `LOOKUP_VISUALIZATIONS_RESPONSE_NORMALIZER`	adapter/converter custom (libreria deprecata)	`@spartacus/epd-visualization/core`	52 — Ricette OCC

8.1 Esempio: cambiare il nome dell'header OMF

Solo config, nessun codice: `OmfConfig` legge da `Config` (integration-libs/omf/order/config/omf-config.ts:L11-L17), quindi basta un `provideConfig` in qualunque injector che contribuisca alla config globale (vedi 11 — Configurazione).

```ts
// esempio illustrativo (simboli: provideConfig da @spartacus/core)
provideConfig({
  omf: { guidHttpHeaderName: 'X-Order-Guid' },
});
```

L'adapter convertirà comunque il nome in minuscolo (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L62-L62).

8.2 Esempio: aggiungere una route ammessa durante il punchout

`punchoutNavigation` è letto tramite `PunchoutNavigationGuardConfig`, `useExisting: Config` (integration-libs/punchout/root/config/punchout-guard-config.ts:L12-L34). Poiché `deepMerge` sostituisce gli array (core-libs/core/src/config/utils/deep-merge.ts:L36-L45), una `allowedCxRoutes` custom deve **ripetere** tutte le route di default, non solo aggiungere la nuova.

```ts
// esempio illustrativo: tutte le route di default di EDIT + 'myCustomRoute'
provideConfig({
  punchoutNavigation: {
    [PunchOutOperation.EDIT]: {
      allowedUrls: ['/'],
      allowedCxRoutes: [
        'punchoutSession', 'punchoutRequisition',
        'category', 'brand', 'quickOrder', 'product', 'cart', 'search', 'punchoutError',
        'myCustomRoute',
      ],
      redirectPage: { cxRoute: 'home' },
    },
  },
});
```

DA VERIFICARE: il tipo `PunchoutNavigationGuardConfig.punchoutNavigation` dichiara obbligatorie tutte e tre le chiavi di operazione (integration-libs/punchout/root/config/punchout-guard-config.ts:L17-L33); se `provideConfig` richieda un cast per una config parziale non è verificabile senza compilare (`node_modules` assente).

8.3 Esempio: personalizzare il guard degli step s4-service

Il guard va registrato **nello stesso injector** e **dopo** `S4ServiceCheckoutModule` (cioè in coda alle `extensions` di `CheckoutWrapperModule`), altrimenti verrà sovrascritto (projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L16-L33).

```ts
// esempio illustrativo
@Injectable({ providedIn: 'root' })
export class MyStepsSetGuard extends CheckoutServiceOrderStepsSetGuard {
  // override di canActivate / isB2BStepSet ...
}

@NgModule({
  providers: [{ provide: CheckoutStepsSetGuard, useExisting: MyStepsSetGuard }],
})
export class MyCheckoutExtensionModule {}
// CheckoutWrapperModule: imports: [CheckoutModule, ...extensions, MyCheckoutExtensionModule]
```

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `CheckoutConfig` **sostituito, non fuso.** `{ provide: CheckoutConfig, useValue: defaultServiceDetailsCheckoutConfig }` nel root module eager: chi inietta `CheckoutConfig` ottiene SOLO quell'oggetto, quindi un `provideConfig({ checkout: ... })` dell'app non è visibile tramite quel token (lo è via `Config`) (integration-libs/s4-service/root/s4-service-root.module.ts:L75-L75 vs feature-libs/checkout/base/root/config/checkout-config.ts:L18-L22). Per personalizzare gli step con s4-service attivo bisogna ri-sostituire il token.
2. **Competizione su** `CheckoutStepsSetGuard`**.** B2B e s4-service lo sostituiscono nello stesso injector lazy: vince l'ultimo modulo importato. Se s4-service fosse importato prima di `CheckoutB2BModule`, il guard s4 verrebbe perso e lo step SERVICE_DETAILS non sarebbe mai verificato (feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L33-L36, projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L16-L31).
3. **Gli array** `cmsComponents` **non si fondono.** Se un'altra libreria registra `featureModules[CHECKOUT_FEATURE].cmsComponents` dopo s4-service senza includere `CheckoutServiceDetails`, quel componente non viene più caricato lazy (core-libs/core/src/config/utils/deep-merge.ts:L36-L45).
4. **Override fuori dal wrapper.** `OmfOrderModule` (e gli override s4-service di token forniti dal chunk, come `OrderOverviewComponentService` e `CheckoutStepsSetGuard`) funzionano solo se importati nello stesso injector (wrapper lazy) e dopo il modulo feature; importarli nell'AppModule non sovrascriverebbe i provider del chunk lazy (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L24-L26, feature-libs/order/components/order-details/order-details.module.ts:L127-L127). DA VERIFICARE a runtime (deduzione dalla gerarchia degli injector lazy). `CpqQuoteModule` non rientra in questa trappola: `CartItemListComponentService` ha solo `providedIn: 'root'` e il suo unico consumatore lo inietta con `inject(CartItemListComponentService)`, quindi non c'è un binding nel chunk da oscurare e un provider registrato nell'injector root avrebbe comunque la precedenza sul fallback `providedIn: 'root'` (feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.service.ts:L9-L16, feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.ts:L48-L48).
5. **Più istanze di facade order con OMF.** `OmfOrderModule` fornisce `OmfOrderHistoryService` e, separatamente, `OrderHistoryFacade` con `useClass: OmfOrderHistoryService`: i due token producono istanze diverse, quindi chi inietta `OmfOrderHistoryService` per estenderlo non riceve l'oggetto usato come facade. Resta anche l'istanza core di `OrderHistoryService` (lo stato è comunque nello store NgRx). Con `useExisting` ce ne sarebbe una sola (integration-libs/omf/order/omf-order.module.ts:L17-L23, feature-libs/order/core/facade/facade-providers.ts:L33-L37).

6. `getQueryParams` **non è un metodo proxy.** Non è nella lista `methods` della `facadeFactory` di `OrderHistoryFacade`: chiamato sul proxy prima del caricamento della feature non esisterebbe; OMF funziona perché lo storico ordini vive nel chunk order (feature-libs/order/root/facade/order-history.facade.ts:L22-L40, feature-libs/order/root/facade/order-history.facade.ts:L148-L148).
7. **Deep link al dettaglio OMF senza guid.** `getOrderGuid` cerca il guid nello stato della LISTA ordini: aprendo direttamente l'URL del dettaglio senza `?guid` e senza lista caricata, l'header non viene inviato (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L31-L53).
8. **CPQ: lettura one-shot del flag.** `isFlagQuote` è letto una volta alla costruzione della riga, mentre il flag è impostato dall'heading in una subscription; l'esito dipende dall'ordine di creazione dei componenti. `CpqQuoteService` è inoltre uno stato globale condiviso fra tutte le tabelle (feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.ts:L49-L49, integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-heading/cpq-quote-heading.component.ts:L63-L63). DA VERIFICARE a runtime.
9. **CPQ: subscription persa.** `CpqQuoteDiscountComponent` sovrascrive `this.subscription` in `ngOnInit`, perdendo la subscription a `isFlag$` creata nel costruttore (leak) (integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-discount-tbody/cpq-quote.component.ts:L35-L44).
10. **CPQ: locale fisso.** I prezzi scontati sono formattati con `'en-US'` ignorando la lingua attiva (integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-discount-tbody/cpq-quote.component.html:L18-L18).
11. **s4-service: pipe mai sottoscritta.** `S4ServiceOrderDetailActionsComponent.ngOnInit` crea `this.order$.pipe(tap(...))` senza `subscribe`: `displayServiceMessage` non viene mai eseguito (integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.component.ts:L56-L58).
12. **s4-service: dipendenza dal formato di** `toLocaleString()`**.** `getServiceDetailsCard` divide la stringa su `','` e chiama `.trim()` sul secondo pezzo: in locali senza virgola `split(',')[1]` è `undefined` → TypeError (integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.component.ts:L141-L145, integration-libs/s4-service/order/components/order-summary/service-details-card.component.ts:L70-L74).
13. `@Optional()` **inefficace.** Il decoratore su una proprietà inizializzata con `inject(OutletContextData)` non ha effetto: `inject` senza `{ optional: true }` lancia se il token manca (integration-libs/s4-service/order/components/order-summary/service-details-card.component.ts:L29-L29).
14. **Subscription annidate non gestite** in `CheckoutServiceDetailsComponent` e `RescheduleServiceOrderComponent` (integration-libs/s4-service/checkout/components/checkout-service-details/checkout-service-details.component.ts:L98-L105, integration-libs/s4-service/order/components/reschedule-service-order/reschedule-service-order.component.ts:L117-L124).
15. **Errori cancel non normalizzati.** `OccCancelServiceOrderAdapter` non usa `tryNormalizeHttpError`, a differenza degli altri adapter (integration-libs/s4-service/order/occ/adapters/occ-cancel-service-order.adapter.ts:L31-L31).
16. **Facade s4-service order senza factory.** Iniettare `CancelServiceOrderFacade`/`RescheduleServiceOrderFacade` fuori dal chunk order fallisce (integration-libs/s4-service/root/facade/cancel-service-order.facade.ts:L12-L15).
17. **Feature** `s4-service` **vuota.** La feature lazy registrata nell'app non porta codice; il lavoro vero è nei wrapper checkout/order (integration-libs/s4-service/s4-service.module.ts:L9-L12).
18. **Stili s4-service.** L'allowlist SCSS non include `cx-reschedule-service-order` né le detail actions (integration-libs/s4-service/_index.scss:L5-L6). Vedi 38 — Stili.
19. **Command eager in punchout.** I Command sono eseguiti subito (`execute` fa `commands$.next`): chiamate senza `subscribe`, come `endPunchoutSession()` nell'error handler della requisition, vengono comunque

eseguite (core-libs/core/src/util/command-query/command.service.ts:L126-L131, integration-libs/punchout/components/punchout-requisition/punchout-requisition.component.ts:L50-L52).

20. **Cache di sessione mai azzerata.** `cachedPunchoutSessionResponse` vive per tutta la vita del `PunchoutService` e non viene mai resettata, nemmeno a fine sessione (integration-libs/punchout/core/facade/punchout.service.ts:L57-L57, integration-libs/punchout/core/facade/punchout.service.ts:L254-L257). DA VERIFICARE se il servizio viene ricreato tra due sessioni nella stessa tab.
21. **Il guard punchout fallisce "aperto".** `catchError(() => of(undefined))` inghiotte ogni errore e in quel caso non applica restrizioni (integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L129-L129).
22. `allowedUrls` **per sottostringa.** `isAllowedUrls` usa `includes` sul path relativo: un URL ammesso parziale autorizza qualsiasi path che lo contenga (integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L77-L81).
23. **Interceptor su ogni richiesta.** `PunchoutCartInterceptor` legge lo stato su OGNI richiesta HTTP (anche non-OCC) e confronta con `url.includes('/carts/{id}')` (integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts:L31-L46).
24. **Refuso nel selettore** `cx-punchout-requsition` (integration-libs/punchout/components/punchout-requisition/punchout-requisition.component.ts:L26-L26).
25. **EPD: servizi** `providedIn: 'root'` **che iniettano** `ElementRef`**/**`ChangeDetectorRef`**.** `VisualPickingTabService`, `VisualViewerService` e altri funzionano solo perché i componenti li ridichiarano in `providers` (integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.service.ts:L30-L39, integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.component.ts:L27-L27, integration-libs/epd-visualization/components/visual-viewer/visual-viewer.component.ts:L30-L30).
26. **EPD: id dello script e callback globale.** Lo script UI5 riceve prima l'id `ui5bootstrap`, poi viene sovrascritto con `sap-ui-bootstrap`; la callback è una proprietà globale su `document` (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1118-L1133).
27. **EPD: accesso diretto a** `document`**.** `getCSSPropertyValue` usa il global `document` e `cx-storefront` invece di `WindowRef` (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1391-L1396).
28. **EPD:** `useSecureConnection: false` nel `ContentResource` (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1561-L1561).
29. **EPD: config mancante non bloccante.** Senza `apis`/`ui5` il validator dà solo un warning in dev. Adapter e viewer fanno cast non nulli: se manca l'oggetto si ottiene un `TypeError` su `.baseUrl`/`.bootstrapUrl`; se manca solo la proprietà, un URL `undefined/vis/public/...` (integration-libs/epd-visualization/epd-visualization-api/adapters/storage-v1/storage-v1.adapter.ts:L49-L56, integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1108-L1108, integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1138-L1138).
30. **EPD: documentazione invertita** in `SelectionDisplayMode` (Highlight descritto come outline e viceversa) (integration-libs/epd-visualization/components/visual-viewer/models/selection-display-mode.ts:L12-L22).
31. **EPD: config letta nel costruttore.** `SceneNodeToProductLookupService` legge `usageIds` nel costruttore con cast: se la config non è completa al momento dell'istanza fallisce (integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L35-L43).
32. **Collection punchout con factory sbagliata.** Lo schematic `add` punta a `./add-punchout/index#addCustomerTicketingFeatures`, funzione che `add-punchout/index.ts` non esporta (copia da customer-ticketing): solo `ng-add` funziona (integration-libs/punchout/schematics/collection.json:L12-L16, integration-libs/punchout/schematics/add-punchout/index.ts:L25-L25).

33. **Collection s4om con factory mancante.** `add-spartacus-library` punta a `./add-library/index#addSpartacusLibrary`, ma `integration-libs/s4om/schematics/` contiene solo `add-s4om`, `collection.json` e `constants.ts` (integration-libs/s4om/schematics/collection.json:L12-L17).
34. **Due entry file in s4om.** `index.ts` e `public_api.ts` esportano lo stesso modulo, ma sia ng-packagr sia il path mapping `@spartacus/s4om` usano solo `public_api.ts`: `index.ts` non è referenziato (integration-libs/s4om/index.ts:L7-L7, integration-libs/s4om/public_api.ts:L10-L10, integration-libs/s4om/ng-package.json:L4-L6, tsconfig.json:L612-L612).

---

10. Come fare debug

- **Quale implementazione ha vinto?** Nel chunk lazy, da un componente della feature (es. order history) ispezionare l'injector con Angular DevTools e verificare che `OrderHistoryFacade` sia un `OmfOrderHistoryService` e `OrderHistoryAdapter` un `OccOmfOrderHistoryAdapter` (integration-libs/omf/order/omf-order.module.ts:L19-L23). Lo stesso per `CheckoutStepsSetGuard` → `CheckoutServiceOrderStepsSetGuard` (integration-libs/s4-service/checkout/components/s4-service-checkout-component.module.ts:L20-L25) e per `CartItemListComponentService` → `CpqQuoteSharedService` (integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L41-L41). Se vince l'originale, controllare l'ordine nel wrapper (§3.1).
- **OMF: header assente.** In DevTools Network, sulla GET del dettaglio ordine, cercare l'header con il nome configurato (minuscolo). Se manca: verificare `?guid=` nell'URL (§3.2 passo 4) o che la lista ordini sia nello store; in dev mode cercare in console il warning *"There is no guidHttpHeaderName configured in OMF configuration"* (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L57-L61).
- **S4OM: link prodotto disabilitato in carrello.** Controllare `product.stock.stockLevelStatus` nella risposta OCC: `notOrderable` → fuori catalogo (integration-libs/s4om/root/services/s4om-product-catalog.service.ts:L28-L36).
- **CPQ: colonna sconto non visibile.** Verificare che le entry abbiano `cpqDiscounts` non vuoto e inserire un breakpoint su `setFlag` in `CpqQuoteHeadingComponent` e su `showBasePriceWithDiscount` in `CpqQuoteSharedService` per osservare l'ordine (§9.8) (integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-heading/cpq-quote-heading.component.ts:L63-L63, integration-libs/cpq-quote/cpq-quote-discount/cpq-qute-shared.service.ts:L21-L23).
- **s4-service: step mancante.** Lo step `serviceDetails` si vede solo se il carrello contiene entry con `productTypes === SERVICE` (integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L99-L131) e se nessun altro ha ri-sostituito `CheckoutConfig` (§9.1). Dopo il submit verificare la `PATCH .../serviceScheduleSlot` e la successiva GET dei dettagli checkout con `servicedAt` nei fields (integration-libs/s4-service/checkout/occ/config/default-occ-checkout-s4-service-config.ts:L10-L25).
- **s4-service: componente CMS non renderizzato.** Controllare in config finale `featureModules.checkout.cmsComponents` / `featureModules.order.cmsComponents` (§9.3).
- **Punchout.** Ispezionare `localStorage` (chiave `spartacus⁉️⁉️punchout`, solo il session id; vedi §3.8) (integration-libs/punchout/root/services/punchout-state-persistence.service.ts:L23-L36); la classe `cxPunchoutSessionActive` sull'elemento root; l'header `punchoutsid` sulle chiamate `/carts/{id}` (integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts:L34-L44). Per redirect inattesi, breakpoint in `PunchoutNavigationGuard.canActivate` su `cxRoute` e `punchoutOperation` (integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L40-L50). Per il refresh token, verificare il valore del toggle `enableExpiredRefreshTokenHandlers` (§3.10).

- **EPD.** Verificare in `<head>` la presenza dello script `sap-ui-bootstrap` e `window.sap`; in dev mode cercare i warning del config validator (core-libs/core/src/config/config-validator/config-validator.ts:L42-L47); controllare le chiamate `.../vis/public/visualization/v1/lookup/visualization` e `.../storage/v1/scenes/{id}/nodes` (integration-libs/epd-visualization/epd-visualization-api/adapters/visualization-v1/visualization-v1.adapter.ts:L50-L68, integration-libs/epd-visualization/epd-visualization-api/adapters/storage-v1/storage-v1.adapter.ts:L49-L108). In SSR il viewer non fa nulla per costruzione (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L86-L88).
- **Backend.** Quali addon Commerce espongano `serviceOrder/*`, `punchout/sessions/*`, `cpqDiscounts`, `scheduleLines`, `guid` non è deducibile dal frontend. `DA VERIFICARE` nella documentazione backend.

---

11. Domande di autoverifica

1. **Perché `OmfOrderModule` è importato in `OrderWrapperModule` e non nell'AppModule?** Perché sostituisce `OrderHistoryAdapter` e `OrderHistoryFacade`, i cui binding originali sono forniti dal chunk lazy order; l'override deve stare nello stesso injector e dopo `OrderModule` (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27, integration-libs/omf/order/omf-order.module.ts:L15-L26).
2. **Da dove prende OMF il guid quando l'URL del dettaglio non ha `?guid`?** Dalla lista ordini nello store NgRx (`OrderSelectors.getOrdersState`), cercando l'ordine per `code` (integration-libs/omf/order/occ-omf-order-history.adapter.ts:L31-L53).
3. **Cosa cambia S4OM nel concetto di "prodotto in catalogo"?** Un prodotto con `stock.stockLevelStatus === 'notOrderable'` non è considerato in catalogo; senza status lo è (integration-libs/s4om/root/services/s4om-product-catalog.service.ts:L23-L37), contro il `!!product` del core (core-libs/core/src/product/services/product-catalog.service.ts:L20-L22).
4. **Come fa CPQ a nascondere il prezzo base standard nelle righe del carrello?** Sostituisce `CartItemListComponentService` con `CpqQuoteSharedService`, il cui `showBasePriceWithDiscount()` ritorna il flag impostato dall'heading a `!hasDiscounts` (integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L41-L41, integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-heading/cpq-quote-heading.component.ts:L55-L63).
5. **Perché con s4-service attivo un `provideConfig({ checkout: { steps: [...] } })` può non avere effetto?** Perché `S4ServiceRootModule` fornisce `CheckoutConfig` con `useValue`, scavalcando il normale `useExisting: Config` (integration-libs/s4-service/root/s4-service-root.module.ts:L75-L75, feature-libs/checkout/base/root/config/checkout-config.ts:L18-L22).
6. **Come viene aggiunto `SERVICE_DETAILS` a un enum definito in un'altra libreria?** Con una module augmentation per i tipi e con un'assegnazione `(CheckoutStepType as any)['SERVICE_DETAILS'] = 'serviceDetails'` per il runtime (integration-libs/s4-service/root/model/augmented-types.model.ts:L38-L47).
7. **Perché s4-service ripete l'intera lista `CHECKOUT_B2B_CMS_COMPONENTS` invece di aggiungere solo `CheckoutServiceDetails`?** Perché `deepMerge` sostituisce gli array: l'ultima `cmsComponents` fusa vince (integration-libs/s4-service/root/s4-service-root.module.ts:L31-L34, core-libs/core/src/config/utils/deep-merge.ts:L36-L45).
8. **Cosa succede dopo che lo slot di servizio è stato salvato con successo?** Il Command emette `CheckoutServiceDetailsSetEvent`; il listener lo trasforma in `CheckoutQueryResetEvent`, che ricarica i dettagli checkout (con `servicedAt` nei fields) (integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L48-L58, integration-libs/s4-service/root/events/checkout-service-

details-event.listener.ts:L23-L29, feature-libs/checkout/base/core/facade/checkout-query.service.ts:L38-L39).

9. **Come gestisce Punchout un refresh token scaduto quando** `enableExpiredRefreshTokenHandlers` **è attivo, e perché non sovrascrive più** `AuthHttpHeaderService`**?** Contribuisce `PunchoutExpiredRefreshTokenHandler` a `EXPIRED_REFRESH_TOKEN_HANDLERS`; il core interroga gli handler in sequenza e usa il fallback solo se nessuno gestisce il caso. Si evita così il conflitto con l'override ASM (integration-libs/punchout/root/punchout.root.module.ts:L67-L71, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L265-L285, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L311-L316).
10. **Che cosa persiste Punchout nello storage del browser e perché un id assente diventa** `''`**?** Solo il `punchoutSessionId`, in `localStorage` sotto la chiave `spartacus⚿⚿punchout` (key `punchout` con prefisso e contesto vuoto aggiunti da `StatePersistenceService`, core-libs/core/src/state/services/state-persistence.service.ts:L124-L130); dopo l'avvio un id assente è scritto come `''` perché `StatePersistenceService` non permette di cancellare la chiave (integration-libs/punchout/root/services/punchout-state-persistence.service.ts:L23-L53).
11. **Perché il visual viewer EPD non fa nulla durante l'SSR?** Il costruttore di `VisualViewerService` esce subito se `!windowRef.isBrowser()`, e `loadVisualization` lato server ritorna un `UnexpectedError`; UI5 è caricato iniettando uno script nel DOM del browser (integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L86-L88, integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1032-L1092, integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L1105-L1140).

TAVOLA 36.1 **4.1 Dove vive ogni override (injector root vs chunk lazy)**

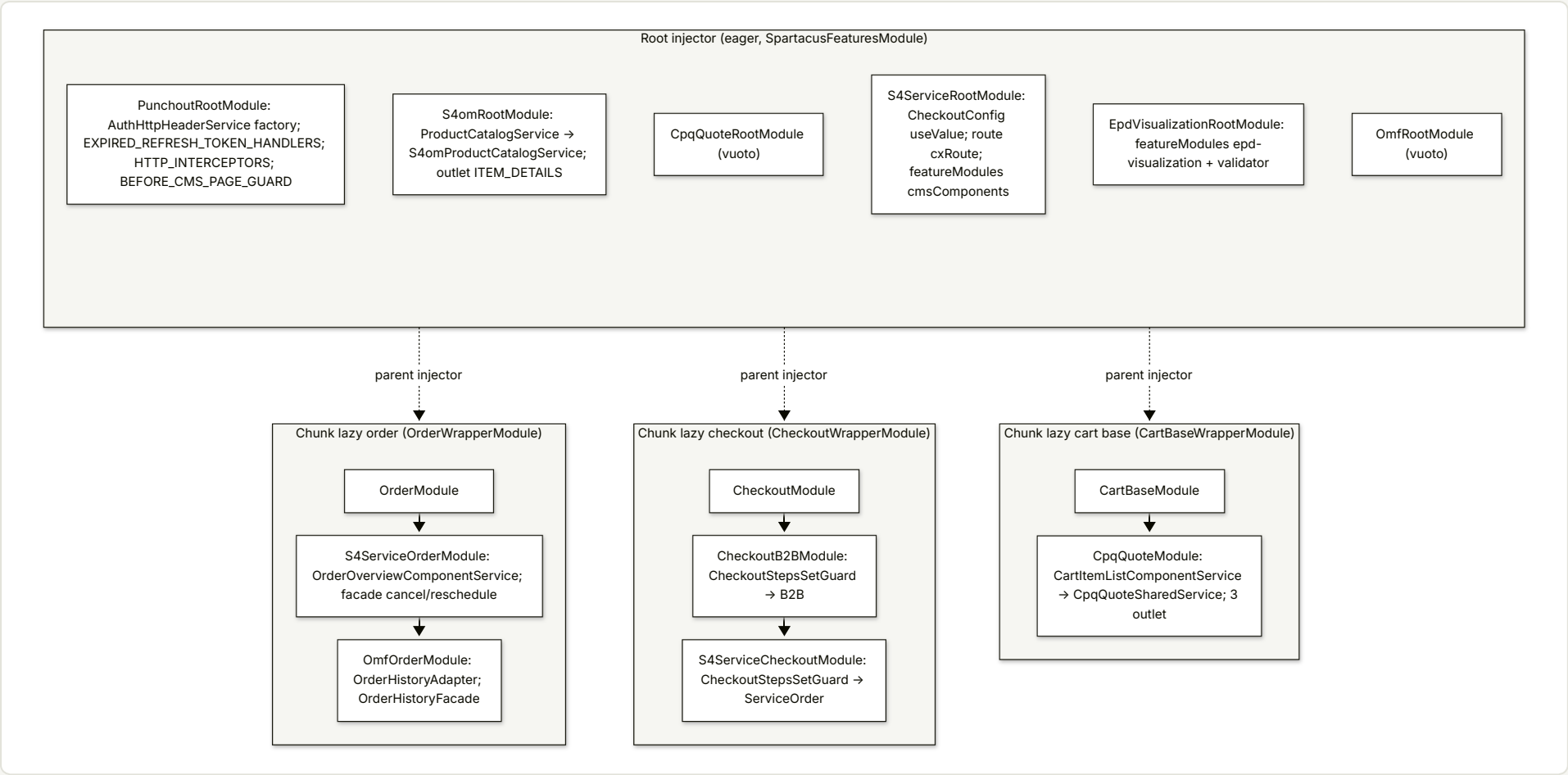


TAVOLA 36.2 **4.2 OMF: dal link dello storico al dettaglio con header guid**

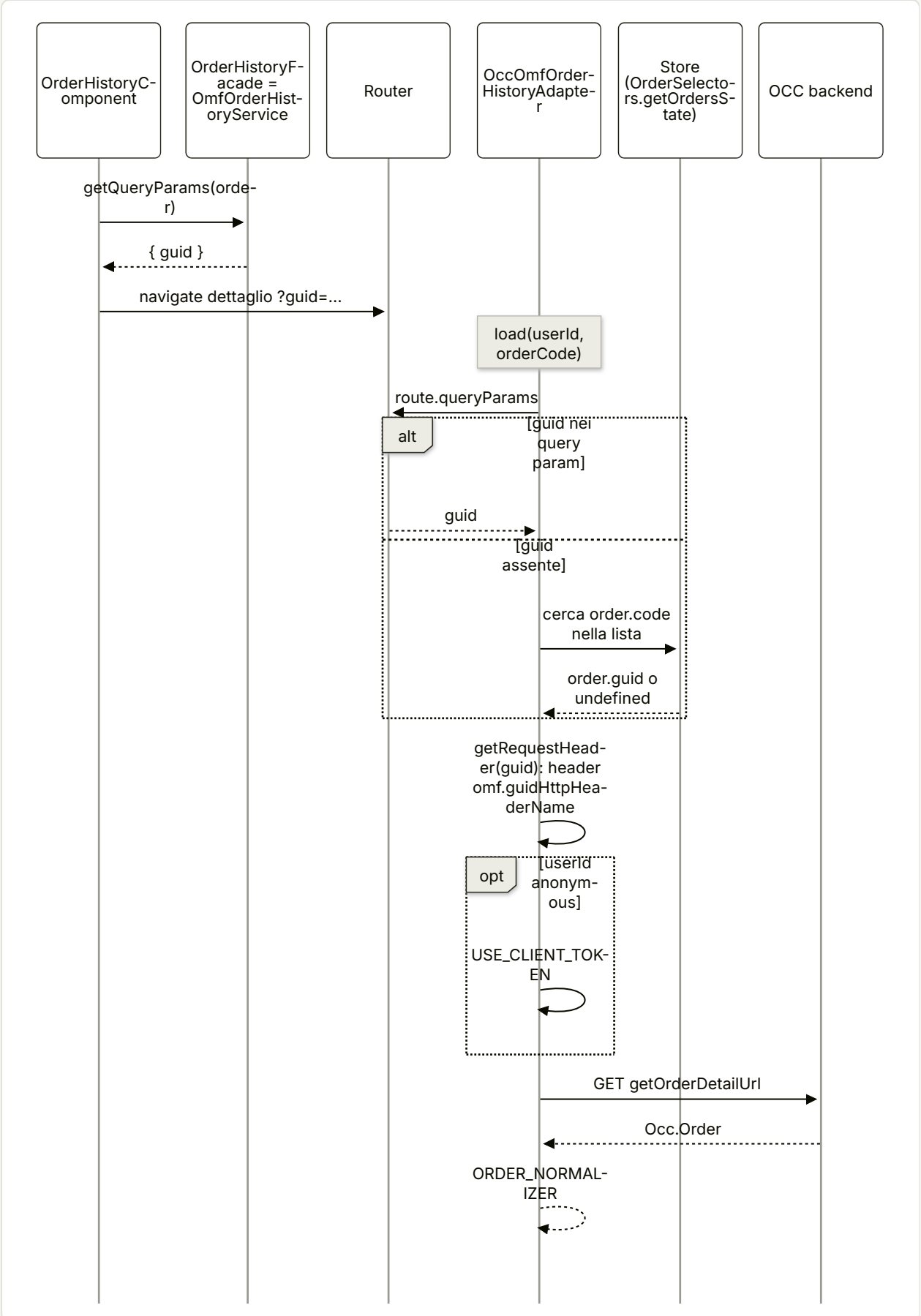


TAVOLA 36.3 **4.3 S/4 Service: step "service details" del checkout**

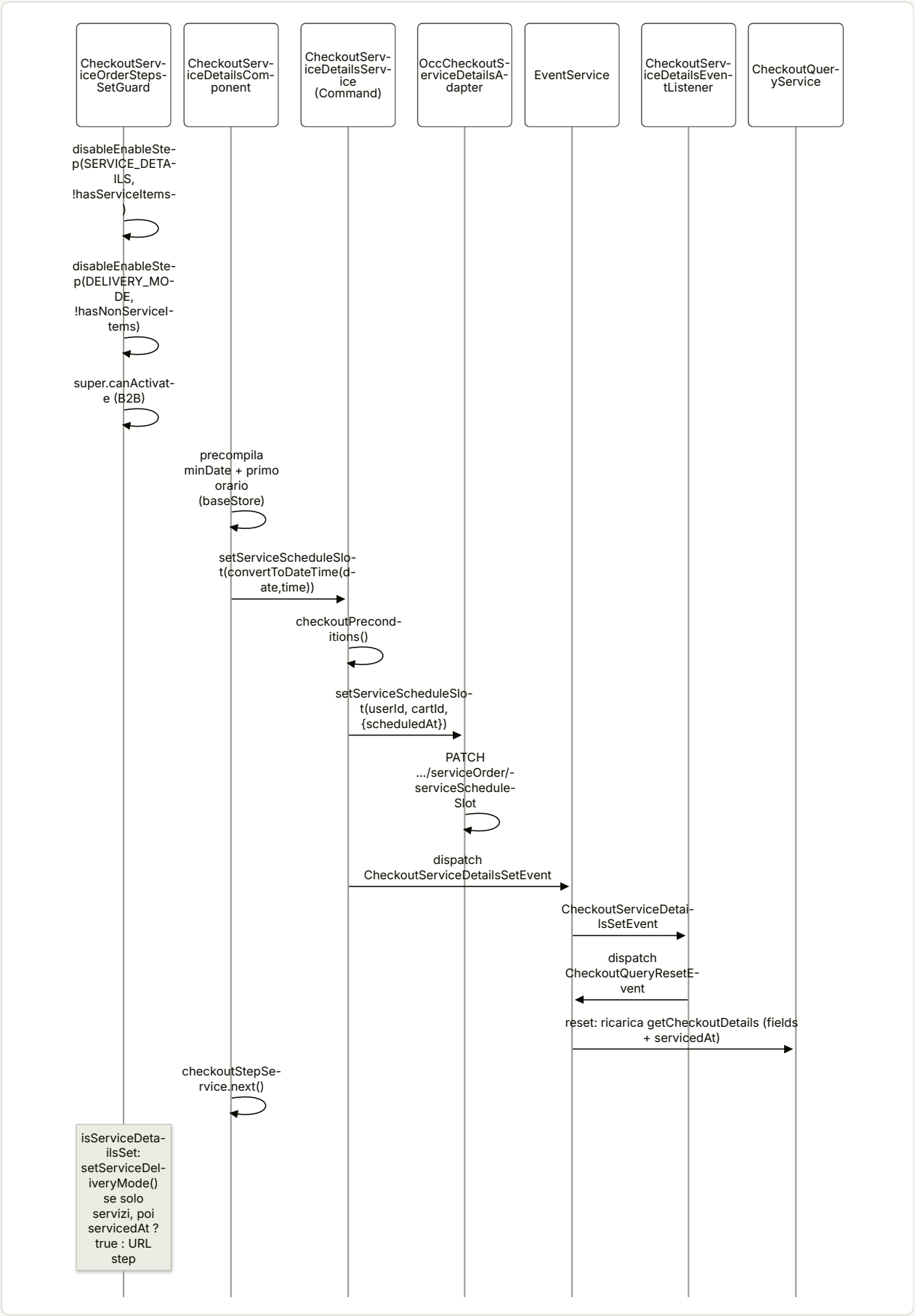


TAVOLA 36.4 **4.4 Punchout: ciclo di vita della sessione**

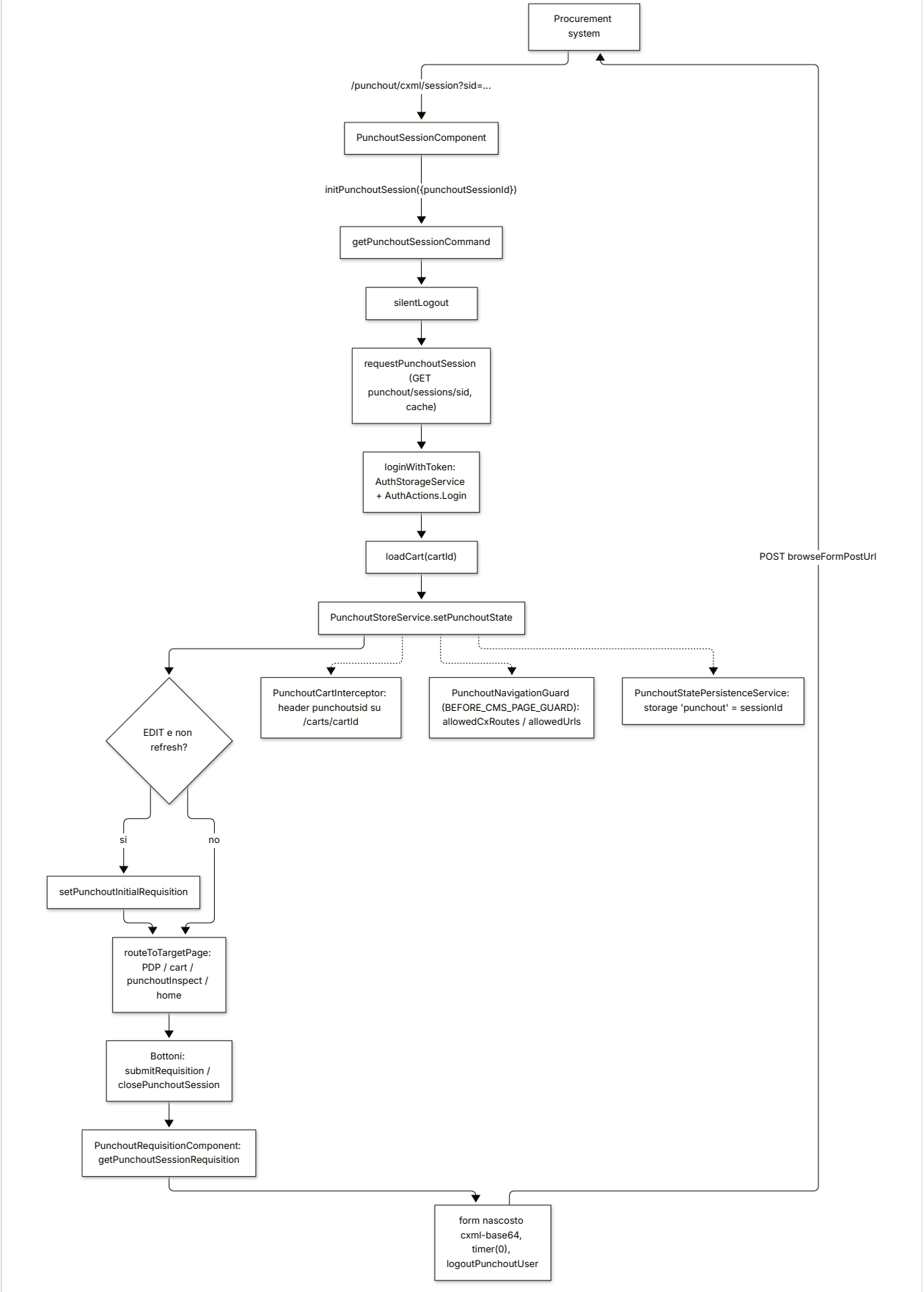
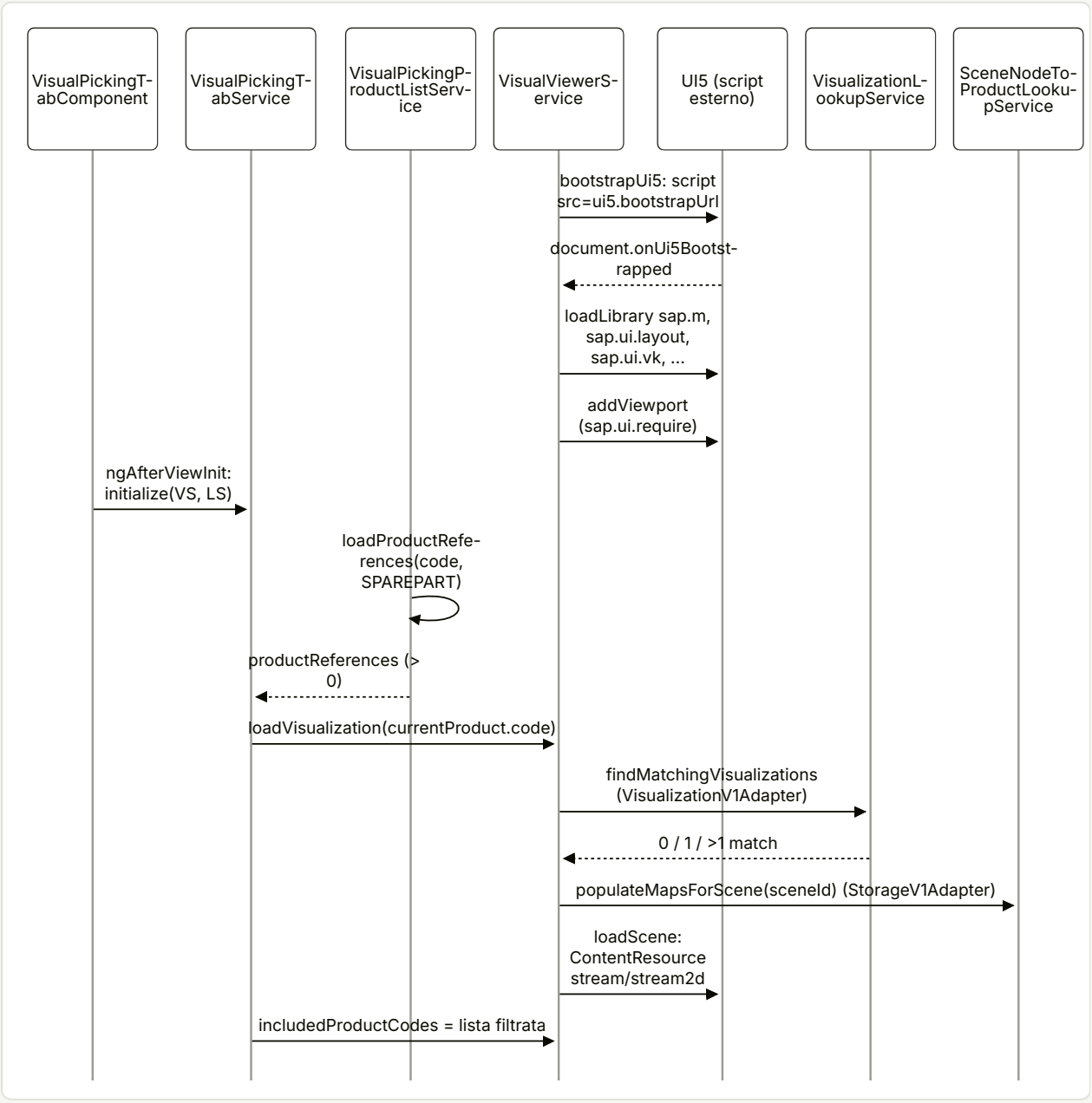


TAVOLA 36.5 **4.6 EPD: visual picking**



37

Internazionalizzazione (i18n)

Chiavi di traduzione a runtime su i18next, chunk, pipe di formattazione, istanza per richiesta in SSR.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Spartacus non usa il sistema i18n di Angular (`@angular/localize`, `$localize`, build per lingua). Tutti i testi dello storefront passano a runtime da **chiavi di traduzione** come `'common.back'` o `'orderDetails.orderNumber'`, risolte da un servizio astratto, `TranslationService`. L'implementazione di default è un adattatore su **i18next**. Questa scelta ha tre conseguenze, che ritornano in tutto il capitolo:

1. **Un solo bundle per tutte le lingue.** La lingua attiva è quella del site context (`LanguageService.getActive()`, core-libs/core/src/site-context/facade/language.service.ts:L49-L54), e il cambio lingua avviene senza ricaricare la pagina: `I18nextInitializer` si sottoscrive alla lingua attiva e chiama `i18next.changeLanguage(lang)` (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L100-L106).
2. **Le traduzioni sono dati di configurazione.** Si forniscono con `provideConfig({ i18n: { resources, chunks, ... } })` e si fondono con il deep merge della configurazione globale (vedi 11 — Configurazione). `I18nConfig` è un alias del `Config` globale (core-libs/core/src/i18n/config/i18n-config.ts:L14-L18), esteso tramite module augmentation (core-libs/core/src/i18n/config/i18n-config.ts:L94-L96).
3. **Le traduzioni sono divise in *chunk*.** Un chunk è un namespace i18next che raggruppa più "namespace logici" (la prima parte della chiave, prima del punto). La mappa `i18n.chunks` dice quali namespace stanno in quale chunk (core-libs/core/src/i18n/config/i18n-config.ts:L85-L90). Un chunk si può caricare in modo eager (`i18n.resources`) oppure lazy tramite un *backend* (`i18n.backend.loadPath` per HTTP, `i18n.backend.loader` per un `import()` dinamico) (core-libs/core/src/i18n/config/i18n-config.ts:L30-L78).

Accanto alle traduzioni testuali ci sono due pipe di formattazione legate alla lingua attiva: `cxDate` (`CxDatePipe`, estende il `DatePipe` di Angular) e `cxNumeric` (`CxNumericPipe`, estende `DecimalPipe`). Entrambe passano ad Angular come locale la lingua del site context (core-libs/core/src/i18n/date.pipe.ts:L25-L27, core-libs/core/src/i18n/numeric.pipe.ts:L22-L24).

Le traduzioni "di prodotto" stanno nel pacchetto `@spartacus/assets` (`core-libs/assets/`) per il core, e nel secondary entry point `assets` di ogni feature lib (es. feature-libs/order/assets/public_api.ts:L7-L7). Il pacchetto assets ha anche alcuni script di generazione (JSON per il lazy loading, `.properties` per i traduttori), di cui vedremo che due sono rimasti indietro rispetto al codice.

Infine c'è un aspetto meno visibile ma fondamentale: in SSR ogni richiesta deve avere **la propria istanza di i18next**, altrimenti richieste concorrenti in lingue diverse si pesterebbero i piedi. Per questo l'istanza non è l'import globale di `i18next` ma un `InjectionToken` con factory `i18next.createInstance()` (core-libs/core/src/i18n/i18next/i18next-instance.ts:L10-L22).

Capitoli collegati: il site context e `LanguageService` sono in 13 — Site context e theme; `CONFIG_INITIALIZER` e `ConfigInitializerService` in 10 — Bootstrap e inizializzazione e 11 — Configurazione; il precaricamento dei chunk dal CMS in 16 — Rendering CMS; il `LoggerService` in 41 — Errori e global message.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/i18n/config/i18n-config.ts	classe astratta `I18nConfig` (`useExisting: Config`) e augmentation di `Config`
core-libs/core/src/i18n/config/default-i18n-config.ts	`defaultI18nConfig`: `fallbackLang: undefined`, `debug: false` (non pubblico)
core-libs/core/src/i18n/config/i18n-config-initializer.ts	`I18nConfigInitializer`: risolve `i18n.fallbackLang` da `context.language[0]`
core-libs/core/src/i18n/i18n.module.ts	`I18nModule` (pipe) + `forRoot()` (provider) + `initI18nConfig`
core-libs/core/src/i18n/translation.service.ts	classe astratta `TranslationService` (`translate`, `loadChunks`)
core-libs/core/src/i18n/translation-chunk.service.ts	`TranslationChunkService`: namespace → chunk, avviso sui duplicati
core-libs/core/src/i18n/translation-resources.ts	tipi `TranslationResourceKey`, `TranslationResources`, `TranslationChunksConfig`
core-libs/core/src/i18n/translatable.ts	interfaccia `Translatable` e guard `isTranslatable`
core-libs/core/src/i18n/translate.pipe.ts	`TranslatePipe` (`cxTranslate`, impura)
core-libs/core/src/i18n/date.pipe.ts	`CxDatePipe` (`cxDate`)
core-libs/core/src/i18n/numeric.pipe.ts	`CxNumericPipe` (`cxNumeric`)
core-libs/core/src/i18n/index.ts	barrel pubblico di i18n
core-libs/core/src/i18n/i18next/i18next-instance.ts	token `I18NEXT_INSTANCE` (un'istanza per root injector)
core-libs/core/src/i18n/i18next/i18next-providers.ts	`i18nextProviders`: `APP_INITIALIZER` + provider dei backend (non pubblico)
core-libs/core/src/i18n/i18next/i18next-initializer.ts	`I18nextInitializer`: opzioni di init, risorse statiche, sincronizzazione lingua
core-libs/core/src/i18n/i18next/i18next-translation.service.ts	`I18nextTranslationService`: implementazione di default di `TranslationService`
core-libs/core/src/i18n/i18next/i18next-plugins/i18next-logger-plugin.ts	token `I18NEXT_LOGGER_PLUGIN`: logger i18next → `LoggerService` (non pubblico)
core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.initializer.ts	classe astratta `I18nextBackendInitializer` (`Applicable`)
core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.providers.ts	`i18nextBackendProviders`: multi provider HTTP + resources-to-backend
core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.service.ts	`I18nextBackendService`: sceglie il backend con `resolveApplicable`
core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts	`I18nextHttpBackendInitializer` (`loadPath`, URL assoluto in SSR)
core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts	token `I18NEXT_HTTP_BACKEND_CLIENT` (richieste via `HttpClient`)
core-libs/core/src/i18n/i18next/i18next-backend/i18next-resources-to-backend.initializer.ts	`I18nextResourcesToBackendInitializer` (`loader`)

PATH	RUOLO
core-libs/core/src/i18n/testing/*.ts	I18nTestingModule, MockTranslatePipe, MockDatePipe, MockTranslationService, mockTranslate
core-libs/storefront/cms-structure/services/cms-i18n.service.ts	CmsI18nService: precarica i chunk dichiarati dai componenti CMS
core-libs/storefront/shared/components/form/date-picker/date-picker.service.ts	DatePickerService: placeholder fisso yyyy-mm-dd
core-libs/assets/src/public_api.ts	entry di @spartacus/assets
core-libs/assets/src/translations/translations.ts	translationChunksConfig + alias translationsXx per 17 lingue
core-libs/assets/src/translations/translation-chunks-config.ts	copia dell'interfaccia TranslationChunksConfig
core-libs/assets/src/translations/<lang>/index.ts	per ogni lingua: 12 JSON (uno per chunk) esposti come oggetto
core-libs/assets/generate-translations-ts-2-json.ts	genera dist/assets/i18n-assets/en/<chunk>.json
core-libs/assets/generate-translations-ts-2-properties.ts	esporta verso lang/properties/*.properties (non compila più, vedi §9)
core-libs/assets/generate-translations-properties-2-ts.ts	reimporta i .properties (non compila più, vedi §9)
feature-libs/*/assets/translations/translations.ts	chunk config e traduzioni di ogni feature lib (44 file tra feature e integration)
scripts/i18n/validate-translations-json-files.ts	validatore dei JSON eseguito da build:libs

3. Come funziona passo per passo

3.1 Registrazione dei provider

BaseCoreModule importa I18nModule.forRoot() (core-libs/core/src/base-core.module.ts:L33-L33), quindi ogni app Spartacus ha i provider i18n senza doverli dichiarare. forRoot() registra quattro cose (core-libs/core/src/i18n/i18n.module.ts:L41-L56):

- la config di default defaultI18nConfig, con fallbackLang: undefined e debug: false (core-libs/core/src/i18n/config/default-i18n-config.ts:L9-L14);
- l'alias TranslationService → I18nextTranslationService con useExisting (core-libs/core/src/i18n/i18n.module.ts:L46-L46);
- i18nextProviders, cioè un APP_INITIALIZER più i provider dei backend (core-libs/core/src/i18n/i18next/i18next-providers.ts:L13-L28);
- un CONFIG_INITIALIZER multi la cui factory è initI18nConfig (core-libs/core/src/i18n/i18n.module.ts:L48-L53).

I18nModule senza forRoot() importa ed esporta solo le tre pipe standalone TranslatePipe, CxDatePipe, CxNumericPipe (core-libs/core/src/i18n/i18n.module.ts:L36-L39): è quello che i moduli feature importano per usare cxTranslate nei template.

3.2 Il fallback dinamico: `I18nConfigInitializer`

`initI18nConfig` decide se serve calcolare la lingua di fallback a runtime: se `config.i18n.fallbackLang` è diverso da `undefined` (anche `false`) restituisce `null`, altrimenti restituisce l'initializer (core-libs/core/src/i18n/i18n.module.ts:L23-L34). `ConfigInitializerService.initialize` salta semplicemente gli initializer `null` (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L140-L143).

Se l'initializer è attivo, dichiara lo scope `i18n.fallbackLang` (core-libs/core/src/i18n/config/i18n-config-initializer.ts:L14-L17) e il suo `resolveConfig` aspetta che lo scope `context.language` sia stabile, poi usa il primo elemento dell'array come fallback (core-libs/core/src/i18n/config/i18n-config-initializer.ts:L24-L33). Lo scope `context` è prodotto da `SiteContextConfigInitializer` (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L26-L26). `getStable(...scopes)` emette subito se il config è già stabile, altrimenti aspetta che nessuno scope in corso si sovrapponga a quelli richiesti (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70). Il risultato viene unito con deep merge sia nel root config sia nel config globale, poi lo scope viene chiuso (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L158-L166).

In pratica: se non configuri `fallbackLang`, la lingua di fallback diventa la lingua di default del sito (la prima di `context.language`), che può arrivare anche dal backend tramite il caricamento dinamico del site context.

3.3 Inizializzazione di i18next (`APP_INITIALIZER`)

L'`APP_INITIALIZER` di `i18nextProviders` aspetta `getStable('i18n')` (cioè anche il fallback dinamico appena descritto) e poi chiama `I18nextInitializer.initialize()` (core-libs/core/src/i18n/i18next/i18next-providers.ts:L14-L25).

`initialize()` costruisce le opzioni con `getI18nextConfig()`, registra il plugin logger e chiama `init(config, callback)` (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L36-L44). Le opzioni base sono (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L49-L61):

- `ns: []`: nessun namespace precaricato (il default di i18next sarebbe `['translation']`, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1619-L1622);
- `fallbackLng` e `debug` dalla config;
- `interpolation.escapeValue: false` e `interpolation.skipOnVariables: false`, entrambi `true` per default in i18next (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1665-L1676);
- `showSupportNotice: false`, con un `@ts-ignore` perché l'opzione non esiste nei tipi della versione installata (in i18next 25.7.4 la stringa `showSupportNotice` non compare nel sorgente ESM).

Se esiste `config.i18n.backend`, alle opzioni si aggiunge quanto restituito da `I18nextBackendService.initialize()` (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L63-L68), vedi §3.6.

Nel **callback** di `init`, e non come opzione `resources` di i18next, Spartacus aggiunge le risorse statiche e avvia la sincronizzazione della lingua (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L38-L43). Il commento spiega perché: passare `resources` a i18next disattiverebbe il caricamento dei chunk dal backend. Infatti in i18next `loadResources` usa il backend solo se `!this.options.resources || this.options.partialBundledLanguages` (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1853-L1857).

`addTranslationResources` itera su lingua e chunk di `config.i18n.resources` e chiama `addResourceBundle(lang, chunk, data, true, true)`, cioè con `deep` e `overwrite` (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L79-L92). Con `deep` i18next fa un `deepExtend` sul bundle esistente, dopo aver copiato i dati con `JSON.parse(JSON.stringify(...))` (deps:i18next-25.7.4/package/dist/esm/i18next.js:L369-L385).

`synchronizeLanguage` si sottoscrive una sola volta (grazie a `??`) a `LanguageService.getActive()` e chiama `changeLanguage(lang)` a ogni emissione; la subscription si chiude in `ngOnDestroy` (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L100-L110). `getActive()` filtra i valori nulli, quindi non emette finché lo store non ha una lingua attiva (core-libs/core/src/site-context/facade/language.service.ts:L49-L54).

Quando si risolve la Promise dell'`APP_INITIALIZER`? In i18next 25.7.4 il default è `initAsync: true` (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1619-L1622) e, poiché Spartacus non passa `resources`, il caricamento viene rimandato con `setTimeout(load, 0)` (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1846-L1850). Il deferred restituito da `init` si risolve in `finish`, dopo `changeLanguage(this.options.lng, finish)`: lì `isInitialized` diventa `true` e viene chiamato il callback (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1834-L1844). Spartacus non passa `lng`, quindi la lingua "vera" arriva solo con la prima emissione di `getActive()` dentro `synchronizeLanguage`. **DA VERIFICARE**: il comportamento interno di `changeLanguage(undefined)` (rilevamento lingua, uso di `fallbackLng`) non è stato letto riga per riga.

`changeLanguage` emette l'evento `languageChanged` solo dopo `loadResources`, e solo se nel frattempo non è stata richiesta un'altra lingua (`isLanguageChangingTo === lng`) (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1955-L1962).

3.4 Istanza e logger

`I18NEXT_INSTANCE` è un `InjectionToken` `providedIn: 'root'` con factory `i18next.createInstance()` (core-libs/core/src/i18n/i18next/i18next-instance.ts:L19-L22). In i18next `createInstance` costruisce un nuovo oggetto i18n (deps:i18next-25.7.4/package/dist/esm/i18next.js:L2122-L2126). Poiché in SSR ogni richiesta ha il proprio platform/root injector, ognuna riceve la propria istanza (vedi 07 — SSR, hydration e TransferState). Tutte le classi che toccano i18next la iniettano con `@Inject(I18NEXT_INSTANCE)` (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L24-L29, core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L23-L28).

`I18NEXT_LOGGER_PLUGIN` è un altro token root: la factory restituisce un modulo i18next `type: 'logger'` che inoltra `log`/`warn`/`error` al `LoggerService` (core-libs/core/src/i18n/i18next/i18next-plugins/i18next-logger-plugin.ts:L17-L40). Il logger interno di i18next scarta `log` e `warn` quando `debug` è false; `error` passa sempre (deps:i18next-25.7.4/package/dist/esm/i18next.js:L219-L231). Quindi `i18n.debug: true` serve a vedere nei log Spartacus gli eventi di caricamento.

3.5 Dalla chiave al chunk: `TranslationChunkService`

Nel costruttore, `TranslationChunkService` inverte `config.i18n.chunks` in una mappa namespace → chunk. Se un namespace compare in più chunk vince il primo, e i doppioni finiscono in `duplicates` (core-libs/core/src/i18n/translation-chunk.service.ts:L20-L33); in dev mode viene emesso un `warn` che indica quale chunk è stato usato (core-libs/core/src/i18n/translation-chunk.service.ts:L34-L36, core-libs/core/src/i18n/translation-chunk.service.ts:L51-L65).

`getChunkNameForKey(key)` prende il segmento prima del primo `.` e lo cerca nella mappa; se non lo trova usa il namespace stesso come nome del chunk (core-libs/core/src/i18n/translation-chunk.service.ts:L39-L49). Con la config di `@spartacus/assets`, per esempio, `'searchBox.placeholder'` → namespace `searchBox` → chunk `common`, perché `searchBox` è tra i namespace del chunk `common` (core-libs/assets/src/translations/translations.ts:L7-L27).

3.6 Scelta del backend

`i18nextBackendProviders` registra in multi provider, nell'ordine, `I18nextHttpBackendInitializer` e `I18nextResourcesToBackendInitializer`, entrambi con `useExisting` (core-libs/core/src/i18n/i18next/

i18next-backend/i18next-backend.providers.ts:L12-L23). `I18nextBackendService.initialize()` sceglie il vincitore con `resolveApplicable` e ne restituisce `initialize()`, oppure `{}` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.service.ts:L20-L38).

- `I18nextHttpBackendInitializer.hasMatch()` è true se c'è `i18n.backend.loadPath` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L38-L40). `initialize()` registra `i18next-http-backend` e restituisce `{ backend: getBackendConfig() }` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L47-L50). La config usa come `request` la funzione del token `I18NEXT_HTTP_BACKEND_CLIENT` e imposta `reloadInterval: false` per non lasciare un `setInterval` che impedirebbe all'SSR di terminare (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L55-L71). In SSR un `loadPath` relativo diventa assoluto premettendo `windowRef.location.origin` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L79-L91).
- `I18NEXT_HTTP_BACKEND_CLIENT` restituisce una funzione `(options, url, payload, callback)` che usa l'`HttpClient` di Angular con `responseType: 'text'`; in caso di errore passa `data: null` e lo status (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L16-L44). Il commento della factory lo motiva così: "It uses Angular HttpClient under the hood, so it works in SSR" (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L23-L26). Per come `HttpClient` passa la richiesta alla catena di interceptor vedi 06 — HttpClient e interceptor.
- `I18nextResourcesToBackendInitializer.hasMatch()` è true se c'è `i18n.backend.loader` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-resources-to-backend.initializer.ts:L23-L25); `initialize()` registra `I18nextResourcesToBackend(loader)` e restituisce `{ backend: {} }` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-resources-to-backend.initializer.ts:L27-L37). `DA VERIFICARE`: il modo in cui il plugin `i18next-resources-to-backend` chiama `loader(lng, ns)` non è verificabile, il pacchetto non è in `/tmp/deps`.

Le dipendenze dichiarate da `@spartacus/core` sono `i18next ^25.7.4`, `i18next-http-backend ^3.0.6` e `i18next-resources-to-backend ^1.2.2` (core-libs/core/package.json:L26-L28).

3.7 Tradurre una chiave: `I18nextTranslationService.translate`

1. Per ogni chiave (stringa o array) calcola il chunk e la chiave con namespace `chunk:key`, e le salva in un array e in una `Map` chiave → chunk (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L42-L53, core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L121-L123). Il separatore `:` coincide con il default `nsSeparator` di i18next (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1631-L1632).
2. Restituisce un Observable "freddo": a ogni subscribe, se `i18next.isInitialized` è false non fa nulla (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L55-L59).
3. Chiama `loadNamespaces(chunks, cb)`; nel callback, se `exists(namespacedKeys)` emette `t(namespacedKeys, options)`, altrimenti segnala la chiave mancante ed emette il valore di fallback (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L61-L79). Passare un array di chiavi a `t` attiva la catena di fallback di i18next: vince la prima chiave risolvibile, come dice il JSDoc di `TranslationService` (core-libs/core/src/i18n/translation.service.ts:L11-L21).
4. Se, al ritorno da `loadNamespaces`, il callback non è ancora stato chiamato e `whitespaceUntilLoaded` è true, emette subito uno spazio non separabile (carattere 160) (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L81-L84).
5. Si registra su `languageChanged` per ritradurre e si deregistra all'unsubscribe; l'Observable non completa mai (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L87-L89).

Il punto 3 regge sulla sincronia del callback: i commenti in testa al metodo chiedono di emettere **in modo sincrono** quando il chunk è già caricato, perché un'emissione differita innescherebbe change detection aggiuntive che, insieme a `switchMap`, possono creare loop infiniti (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L35-L40). In i18next, `loadNamespaces` aggiunge i namespace mancanti a `options.ns` e chiama `loadResources` (deps:i18next-25.7.4/package/dist/esm/i18next.js:L2075-L2089). Il callback è sincrono in due casi: senza backend registrato (il connettore emette un warn e chiama subito il callback, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1548-L1552) e quando non c'è nulla da caricare né in sospeso (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1555-L1559). È questa la "cache under the hood" citata nel commento.

Il valore di fallback è `[chunk:key, ...]` in dev mode e uno spazio non separabile in produzione (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L100-L106); il warn sulle chiavi mancanti compare solo in dev mode ed elenca i chunk che si è tentato di caricare (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L108-L119).

`loadChunks(chunkNames)` delega direttamente a `i18next.loadNamespaces(chunkNames)` (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L93-L95).

3.8 La pipe `cxTranslate`

`TranslatePipe` si chiama `cxTranslate` ed è `pure: false` (core-libs/core/src/i18n/translate.pipe.ts:L25-L29): Angular la chiama a ogni change detection, e la pipe deve quindi decidere da sola quando risottoscriversi.

- Input falsy → `''`, con un `error` in dev mode (core-libs/core/src/i18n/translate.pipe.ts:L46-L53).
- Un `Translatable` con `raw` restituisce il testo così com'è; con `params` i parametri si uniscono alle options, e a parità di nome prevalgono i `params`; la chiave è `input.key` oppure l'input stesso (core-libs/core/src/i18n/translate.pipe.ts:L55-L63). `Translatable` ha `key?: string | string[]`, `params?`, `raw?`, e `isTranslatable` controlla che sia un oggetto non null con `key` o `raw` (core-libs/core/src/i18n/translatable.ts:L7-L23).
- La pipe si risottoscrive solo se cambia la chiave (confronto per riferimento) o se le options non sono shallow-equal; chiama `service.translate(key, options, true)`, quindi sempre con `whitespaceUntilLoaded`, e a ogni valore salva il risultato e chiama `cd.markForCheck()` (core-libs/core/src/i18n/translate.pipe.ts:L69-L89).
- `ngOnDestroy` chiude la subscription (core-libs/core/src/i18n/translate.pipe.ts:L91-L95).

Siccome la subscribe avviene dentro `transform` e il valore sincrono viene salvato prima del `return this.translatedValue` (core-libs/core/src/i18n/translate.pipe.ts:L65-L66), con un chunk già caricato il testo è disponibile già al primo passaggio di change detection.

3.9 Precaricamento dei chunk per i componenti CMS

Un mapping di componente CMS può dichiarare `i18nKeys?: string[]` (core-libs/core/src/cms/config/cms-config.ts:L76-L76). Quando le guard CMS di una pagina restituiscono `true`, `CmsPageGuardService` chiama `cmsI18n.loadForComponents(componentTypes)` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L81-L85). `CmsComponentsService.getI18nKeys` raccoglie le chiavi dei soli componenti che devono essere renderizzati (core-libs/storefront/cms-structure/services/cms-components.service.ts:L266-L276), e `CmsI18nService` le converte in un `Set` di chunk e chiama `translation.loadChunks` (core-libs/storefront/cms-structure/services/cms-i18n.service.ts:L21-L28). Così i chunk lazy partono durante la navigazione, prima che i componenti vengano creati. A questo SHA, però, nessun mapping di componente del repo imposta `i18nKeys`: il meccanismo esiste ma è usato solo dalle app che lo configurano (grep su `*.ts`).

3.10 `cxDate` e `cxNumeric`

`CxDatePipe` (`cxDate`) estende `DatePipe` e chiama `super('')`, cioè con un `LOCALE_ID` vuoto (core-libs/core/src/i18n/date.pipe.ts:L12-L19). `transform` passa a `super.transform` come quarto argomento la lingua attiva (core-libs/core/src/i18n/date.pipe.ts:L25-L27); nel `DatePipe` di Angular prevale il locale passato, `locale || this.locale` (angular:packages/common/src/pipes/date_pipe.ts:L263-L275).

`getLang()` legge la lingua attiva in modo sincrono (subscribe seguita subito da unsubscribe, `''` se lo store non ha ancora emesso) e la valida con `getLocaleId(lang)`. Se Angular non ha i locale data per quella lingua, `getLocaleId` lancia, la pipe emette un warn (solo in dev mode) e ripiega su `'en'` (core-libs/core/src/i18n/date.pipe.ts:L29-L55). In Angular `findLocaleData` normalizza il codice (minuscolo, _ → -), prova il locale padre, ricade su `localeEn` se il padre è `en` e altrimenti lancia `MISSING_LOCALE_DATA` (angular:packages/core/src/i18n/locale_data_api.ts:L49-L72, angular:packages/core/src/i18n/locale_data_api.ts:L181-L183).

`CxNumericPipe` (`cxNumeric`) fa la stessa cosa sopra `DecimalPipe` (core-libs/core/src/i18n/numeric.pipe.ts:L12-L53), che in Angular usa `locale ||= this._locale` (angular:packages/common/src/pipes/number_pipe.ts:L109-L120). Una differenza: in `CxNumericPipe` i metodi di supporto sono `protected` (quindi estendibili), in `CxDatePipe` sono `private`.

Nessuna delle due dichiara `pure`, quindi sono pipe pure: il default di `@Pipe` è `{pure: true, ...p}` (angular:packages/core/src/metadata/directives.ts:L752-L752).

L'app demo registra i locale `de`, `ja`, `zh` con `registerLocaleData` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L29-L31); `en` è già incluso in Angular.

Date picker. Nelle librerie a questo SHA non esiste alcun componente DateTimePicker. Il selettore `cx-date-time-picker` sopravvive solo in un helper E2E Cypress (`fillDateTimePicker`), un residuo che non trova corrispondenza nel codice delle librerie (projects/storefrontapp-e2e-cypress/cypress/helpers/b2b/my-company/features/utils/form.ts:L85-L90). Il componente data dello storefront è `DatePickerComponent` (`cx-date-picker`), che importa `TranslatePipe` e `CxDatePipe` (core-libs/storefront/shared/components/form/date-picker/date-picker.component.ts:L27-L58). Il suo template usa un `<input type="date">` con placeholder e pattern presi da `DatePickerService`, e passa a `cx-form-errors` i limiti `min`/`max` formattati con `cxDate` e la label tradotta con `cxTranslate` (core-libs/storefront/shared/components/form/date-picker/date-picker.component.html:L1-L25). Il placeholder è fisso, `yyyy-mm-dd`, e non dipende dalla lingua attiva (core-libs/storefront/shared/components/form/date-picker/date-picker.service.ts:L24-L37). Il componente è trattato in 45 — Componenti storefront.

3.11 Da dove arrivano le traduzioni: `@spartacus/assets` e le feature lib

`@spartacus/assets` esporta `translationChunksConfig` e le traduzioni (core-libs/assets/src/public_api.ts:L11-L12). `translationChunksConfig` definisce 12 chunk core: `common` (18 namespace), `deliveryMode`, `myAccountV2Consent`, `myAccountV2NotificationPreference`, `myAccount`, `payment`, `pdf`, `product`, `pwa`, `siteThemeSwitcher`, `user`, `video` (core-libs/assets/src/translations/translations.ts:L7-L66). Lo stesso file riesporta 17 lingue con alias `translationsXx`: cs, de, en, es, es_CO, fr, hi, hu, id, it, ja, ko, pl, pt, ru, zh, zh_TW (core-libs/assets/src/translations/translations.ts:L68-L84). Ogni `index.ts` di lingua importa 12 JSON, uno per chunk, e li espone come oggetto `{ chunk: json }` (core-libs/assets/src/translations/en/index.ts:L7-L32).

L'app demo carica in modo eager en, ja, de e zh, passa `chunks: translationChunksConfig` e fissa `fallbackLang: 'en'` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L55-L67). Ogni modulo feature dell'app aggiunge le proprie risorse e i propri chunk con un altro `provideConfig(<I18nConfig>{ i18n: ... })`, che si fonde con deep merge (projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L38-L48). Le traduzioni della feature sono fornite

eager, nel modulo root della feature, anche se il codice della feature è lazy (`featureModules` + `import()`, projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L26-L33).

Ogni feature lib ha un secondary entry point `assets` (feature-libs/order/assets/public_api.ts:L7-L7) con una chunk config tipizzata `TranslationChunksConfig` di `@spartacus/core` e le 17 lingue riesportate con prefisso, es. `orderTranslationsEn` (feature-libs/order/assets/translations/translations.ts:L7-L42). Tra `feature-libs/` e `integration-libs/` ci sono 44 file `assets/translations/translations.ts`.

3.12 Generazione e validazione

- Lo script `build` di assets esegue `nx build assets` e poi `generate:translations:ts-2-json`; ci sono anche `properties-2-ts` (seguito da prettier) e `ts-2-properties` (core-libs/assets/package.json:L6-L11).
- `generate-translations-ts-2-json.ts` svuota e riscrive `dist/assets/i18n-assets/<lang>/<chunk>.json`, ma solo per `en`; se `dist/assets/` non esiste stampa un messaggio ed esce (core-libs/assets/generate-translations-ts-2-json.ts:L10-L43). Sono i file pensati per `loadPath: 'assets/i18n-assets/{{lng}}/{{ns}}.json'`, l'esempio del JSDoc (core-libs/core/src/i18n/config/i18n-config.ts:L37-L40). DA VERIFICARE: come un'app copi `dist/assets/i18n-assets` tra i propri asset; non è stata letta nessuna config `assets` di build che lo faccia.
- `build:libs` del repo esegue prima `scripts/i18n/validate-translations-json-files.ts` (package.json:L35-L35). Il validatore impone un limite di 50 KB per file e rifiuta valori, e chiavi delle sole foglie (stringhe), che contengono `<script`, `onerror`, `onload`, `onclick`, `javascript:`, `__proto__`, `constructor`, `eval` (scripts/i18n/validate-translations-json-files.ts:L9-L19). Le chiavi che puntano a oggetti annidati vengono attraversate senza controllo (scripts/i18n/validate-translations-json-files.ts:L39-L64). Ogni foglia deve essere una stringa e non deve avere la forma di un JSON (oggetto o array) (scripts/i18n/validate-translations-json-files.ts:L20-L73). Scansiona solo `feature-libs`, `integration-libs` e `projects` (scripts/i18n/validate-translations-json-files.ts:L97-L105).

---

4. Diagrammi

4.1 Bootstrap: dalla config all'istanza i18next pronta

■ **TAVOLA 37.1** diagramma a tutta pagina, in fondo al capitolo

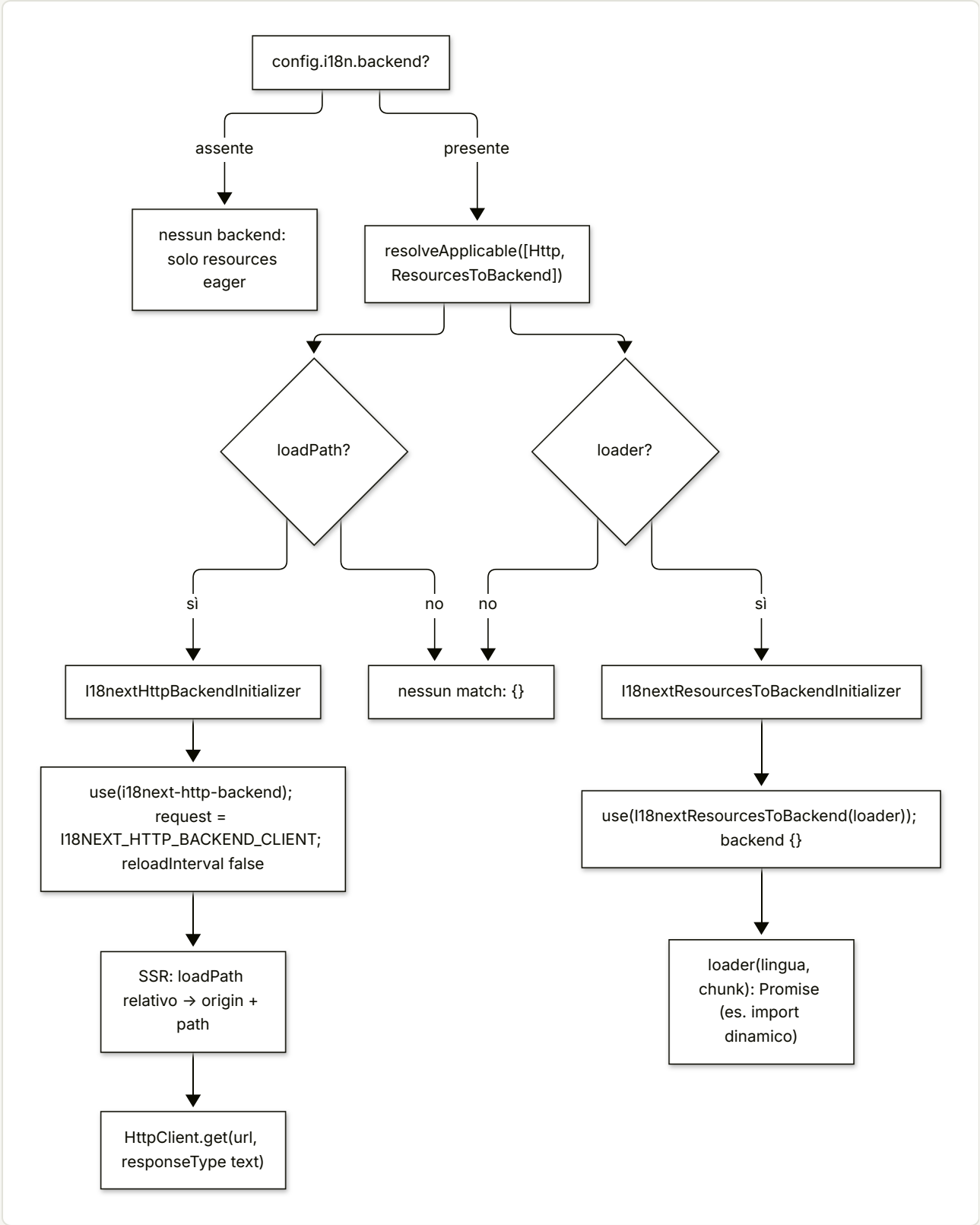
Riferimenti: core-libs/core/src/i18n/i18n.module.ts:L23-L56, core-libs/core/src/i18n/config/i18n-config-initializer.ts:L24-L33, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L140-L166, core-libs/core/src/i18n/i18next/i18next-providers.ts:L14-L25, core-libs/core/src/i18n/i18next/i18next-initializer.ts:L36-L106, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1834-L1850, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1955-L1962.

4.2 `{{ 'key' | cxTranslate }}`: dal template al testo

■ **TAVOLA 37.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/i18n/translate.pipe.ts:L42-L89, core-libs/core/src/i18n/translation-chunk.service.ts:L39-L49, core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L42-L119.

4.3 Scelta del backend e lazy loading di un chunk



Se `loadPath` e `loader` sono entrambi presenti, entrambi corrispondono con pari priorità e vince il provider registrato per ultimo, cioè resources-to-backend (§9). Riferimenti: `core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.providers.ts:L12-L23`, `core-libs/core/src/util/applicable.ts:L63-L86`, `core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L38-L91`, `core-libs/`

core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L27-L42, core-libs/core/src/i18n/i18next/i18next-backend/i18next-resources-to-backend.initializer.ts:L23-L37. Il passaggio finale del ramo `loader` dipende dal plugin esterno (DA VERIFICARE, §3.6).

---

5. Estratti di codice reale

Il fallback dinamico viene attivato solo se `fallbackLang` non è configurato — core-libs/core/src/i18n/i18n.module.ts:L23-L56

```ts
export function initI18nConfig(
  configInitializer: I18nConfigInitializer,
  config: I18nConfig
): ConfigInitializer | null {
  /**
   * If `fallbackLang` was already configured statically
   */
  if (config?.i18n?.fallbackLang !== undefined) {
    return null;
  }
  return configInitializer;
}

@NgModule({
  imports: [TranslatePipe, CxDatePipe, CxNumericPipe],
  exports: [TranslatePipe, CxDatePipe, CxNumericPipe],
})
export class I18nModule {
  static forRoot(): ModuleWithProviders<I18nModule> {
    return {
      ngModule: I18nModule,
      providers: [
        provideDefaultConfig(defaultI18nConfig),
        { provide: TranslationService, useExisting: I18nextTranslationService },
        ...i18nextProviders,
        {
          provide: CONFIG_INITIALIZER,
          useFactory: initI18nConfig,
          deps: [I18nConfigInitializer, I18nConfig],
          multi: true,
        },
      ],
    };
  }
}
```

L'`APP_INITIALIZER` aspetta lo scope `i18n` — core-libs/core/src/i18n/i18next/i18next-providers.ts:L13-L28

```ts
export const i18nextProviders: Provider[] = [
  {
    provide: APP_INITIALIZER,
    useFactory: () => {
      const configInitializerService = inject(ConfigInitializerService);
      const i18nextInitializer = inject(I18nextInitializer);
      return () =>
        lastValueFrom(configInitializerService.getStable('i18n')).then(() =>
          i18nextInitializer.initialize()
        );
    },
    multi: true,
  },

  ...i18nextBackendProviders,
];
```

Init di i18next: risorse statiche aggiunte nel callback — core-libs/core/src/i18n/i18next/i18next-initializer.ts:L36-L71

```ts
  initialize(): Promise<any> {
    const i18nextConfig = this.getI18nextConfig();
    return this.i18next.use(this.loggerPlugin).init(i18nextConfig, () => {
      // Don't use i18next's 'resources' config key for adding static translations,
      // because it will disable loading chunks from backend. We add resources here, in the init's
callback.
      this.addTranslationResources();
      this.synchronizeLanguage();
    });
  }

  /**
   * Returns the configuration for initializing an i18next instance.
   */
  protected getI18nextConfig(): InitOptions {
    let i18nextConfig: InitOptions = {
      ns: [], // don't preload any namespaces
      fallbackLng: this.config.i18n?.fallbackLang,
      debug: this.config.i18n?.debug,
      interpolation: {
        escapeValue: false,
        skipOnVariables: false,
      },

      // @ts-ignore -- In CXSPA-12377 we'll remove this @ts-ignore, after we upgrade to the latest
i18next version
      showSupportNotice: false,
    };
```

```
  if (this.config.i18n?.backend) {
    i18nextConfig = {
      ...i18nextConfig,
      ...this.i18nextBackendService.initialize(),
    };
  }

  return i18nextConfig;
}
```

Il cuore di `translate`: emissione sincrona, spazio non separabile in attesa, ritraduzione al cambio lingua — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L55-L90

```ts
  return new Observable<string>((subscriber) => {
    const translate = () => {
      if (!this.i18next.isInitialized) {
        return;
      }

      // make sure chunks are loaded
      let namespacesLoaded = false;

      // loadNamespaces has cache under the hood, so it won't load the same chunk twice
      // when namespaces are already loaded, the callback will be called synchronously:
      this.i18next.loadNamespaces(
        Array.from(chunkNamesByKeys.values()),
        () => {
          namespacesLoaded = true;
          if (this.i18next.exists(namespacedKeys, options)) {
            subscriber.next(
              this.i18next.t(namespacedKeys, options) as string
            );
          } else {
            this.reportMissingKey(chunkNamesByKeys);
            subscriber.next(this.getFallbackValue(namespacedKeys));
          }
        }
      );

      // if namespaces are not loaded yet, we can emit a non-breaking space
      if (!namespacesLoaded && whitespaceUntilLoaded) {
        subscriber.next(this.NON_BREAKING_SPACE);
      }
    };

    translate();
```

```
    this.i18next.on('languageChanged', translate);
    return () => this.i18next.off('languageChanged', translate);
  });
```

Chiave → chunk, con ripiego sul namespace — core-libs/core/src/i18n/translation-chunk.service.ts:L39-L49

```ts
protected readonly KEY_SEPARATOR = '.';

getChunkNameForKey(key: string): string {
  const mainKey = (key || '').split(this.KEY_SEPARATOR)[0];
  const chunk = this.chunks && this.chunks[mainKey];

  if (!chunk) {
    return mainKey; // fallback to main key as a chunk
  }
  return chunk;
}
```

La pipe impura che si risottoscrive solo quando serve — core-libs/core/src/i18n/translate.pipe.ts:L69-L89

```ts
private translate(key: any, options: object) {
  if (
    key !== this.lastKey ||
    !ObjectComparisonUtils.shallowEqualObjects(options, this.lastOptions)
  ) {
    this.lastKey = key;
    this.lastOptions = options;

    if (this.sub) {
      this.sub.unsubscribe();
    }
    this.sub = this.service
      .translate(key, options, true)
      .subscribe((val) => this.markForCheck(val));
  }
}

private markForCheck(value: string) {
  this.translatedValue = value;
  this.cd.markForCheck();
}
```

HTTP backend: niente `setInterval` e URL assoluti in SSR — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L55-L91

```ts
  protected getBackendConfig(): HttpBackendOptions {
    if (!this.config.i18n?.backend?.loadPath) {
      throw new Error('Missing config `i18n.backend.loadPath`.');
    }

    const loadPath = this.getLoadPath(this.config.i18n.backend.loadPath);

    const backend: HttpBackendOptions = {
      loadPath,
      request: this.i18nextHttpClient,

      // Disable the periodical reloading. Otherwise SSR would not finish due to the pending task `setInterval()`
      // ...
      reloadInterval: false,
    };
    return backend;
  }
  // ...
  protected getLoadPath(path: string): string {
    if (!this.windowRef.isBrowser() && !path.match(/^http(s)?:\/\//)) {
      if (path.startsWith('/')) {
        path = path.slice('/'.length);
      }
      if (path.startsWith('./')) {
        path = path.slice('./'.length);
      }
      const serverRequestOrigin = this.windowRef.location.origin;
      return `${serverRequestOrigin}/${path}`;
    }
    return path;
  }
```

Il client HTTP usato da i18next è l'`HttpClient` di Angular — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L27-L42

```ts
    return (
      _options: HttpBackendOptions,
      url: string,
      _payload: string | object,
      callback: RequestCallback
    ) => {
      httpClient.get(url, { responseType: 'text' }).subscribe({
        next: (data) => callback(null, { status: 200, data }),
        error: (error) =>
          callback(error, {
            // a workaround for https://github.com/i18next/i18next-http-backend/issues/82
            data: null as any,
```

```
          status: error.status,
        }),
    });
  };
```

`cxDate`: **lingua attiva come locale, ripiego su** `en` — `core-libs/core/src/i18n/date.pipe.ts:L25-L47`

```ts
transform(value: any, format?: string, timezone?: string): string | null {
  return super.transform(value, format, timezone, this.getLang());
}

private getLang() {
  const lang = this.getActiveLang();
  try {
    getLocaleId(lang);
    return lang;
  } catch {
    this.reportMissingLocaleData(lang);
    return 'en';
  }
}

private getActiveLang(): string {
  let result = '';
  this.language
    .getActive()
    .subscribe((lang) => (result = lang))
    .unsubscribe();
  return result;
}
```

Configurazione dell'app demo — `projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L55-L67`

```ts
  provideConfig({
    // we bring in static translations to be up and running soon right away
    i18n: {
      resources: {
        en: translationsEn,
        ja: translationsJa,
        de: translationsDe,
        zh: translationsZh,
      },
      chunks: translationChunksConfig,
      fallbackLang: 'en',
    },
  }),
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
i18n.fallbackLang	string | false	undefined (core-libs/core/src/i18n/config/default-i18n-config.ts:L9-L14); se resta undefined diventa context.language[0]	core-libs/core/src/i18n/config/i18n-config.ts:L20-L23	core-libs/core/src/i18n/i18n.module.ts:L30-L32, core-libs/core/src/i18n/config/i18n-config-initializer.ts:L24-L33, core-libs/core/src/i18n/i18next/i18next-initializer.ts:L52-L52
i18n.backend.loadPath	string con {{lng}} e {{ns}}	assente	core-libs/core/src/i18n/config/i18n-config.ts:L31-L51	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L63-L63, core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L38-L40, core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L55-L60
i18n.backend.loader	(language, chunkName) => Promise<TranslationResourceKey>	assente	core-libs/core/src/i18n/config/i18n-config.ts:L53-L71	core-libs/core/src/i18n/i18next/i18next-backend/i18next-resources-to-backend.initializer.ts:L23-L34
i18n.resources	TranslationResources (lingua → chunk → oggetto)	assente (?? {})	core-libs/core/src/i18n/config/i18n-config.ts:L74-L78, tipo in core-libs/core/src/i18n/translation-resources.ts:L7-L18	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L79-L92
i18n.debug	boolean	false (core-libs/core/src/i18n/config/default-i18n-config.ts:L12-L12)	core-libs/core/src/i18n/config/i18n-config.ts:L80-L83	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L53-L53 (opzione debug di i18next, che sblocca log/warn del logger)
i18n.chunks	{ [chunk: string]: string[] }	assente (|| {})	core-libs/core/src/i18n/config/i18n-config.ts:L85-L90	core-libs/core/src/i18n/translation-chunk.service.ts:L21-L21
context.language (esterna, site context)	string[]	—	site context (13)	core-libs/core/src/i18n/config/i18n-config-initializer.ts:L25-L29
CmsComponentMapping.i18nKeys (esterna, CMS)	string[]	assente	core-libs/core/src/cms/config/cms-config.ts:L76-L76	core-libs/storefront/cms-structure/services/cms-components.service.ts:L266-L276

Opzioni i18next imposte da Spartacus e non configurabili via I18nConfig (si cambiano solo estendendo I18nextInitializer.getI18nextConfig): ns: [], interpolation.escapeValue: false, interpolation.skipOnVariables: false, showSupportNotice: false (core-libs/core/src/i18n/i18next/i18next-

initializer.ts:L50-L61); per il backend HTTP `reloadInterval: false` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L68-L68).

Esempio di config lazy, citato letteralmente dai JSDoc: `loadPath: 'assets/i18n-assets/{{lng}}/{{ns}}.json'` (core-libs/core/src/i18n/config/i18n-config.ts:L37-L40) e `loader: (language, chunkName) => import(`../../assets/i18n-assets/${language}/${chunkName}.json`)` (core-libs/core/src/i18n/config/i18n-config.ts:L63-L66). Il JSDoc di `loadPath` lo sconsiglia per gli asset locali e raccomanda `loader`, "more performant (especially in SSR)" (core-libs/core/src/i18n/config/i18n-config.ts:L42-L48).

---

7. Commenti e note notevoli dal sorgente

- *"Each SSR request gets its own instance of i18next. ... The reference to the static global instance of `i18next` ... should not be used anywhere else, because otherwise it would be shared in between all SSR requests and can cause concurrency issues."* — core-libs/core/src/i18n/i18next/i18next-instance.ts:L10-L18. È la regola d'oro per chi scrive estensioni: mai `import i18next from 'i18next'` per usare l'istanza globale, sempre `inject(I18NEXT_INSTANCE)`.
- *"Don't use i18next's 'resources' config key for adding static translations, because it will disable loading chunks from backend."* — core-libs/core/src/i18n/i18next/i18next-initializer.ts:L39-L40.
- *"@ts-ignore -- In CXSPA-12377 we'll remove this @ts-ignore, after we upgrade to the latest i18next version"* — core-libs/core/src/i18n/i18next/i18next-initializer.ts:L59-L60. In i18next 25.7.4 l'opzione `showSupportNotice` non esiste nel sorgente ESM, quindi oggi viene semplicemente ignorata. DA VERIFICARE: a quale versione di i18next si riferisca CXSPA-12377 non si può verificare qui.
- *"Moreover, we SHOULD emit a value (or a fallback value) synchronously (not in a promise/setTimeout). Otherwise, we the will trigger additional deferred change detection ... which together with `switchMap` operator may lead to an infinite loop."* — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L38-L40.
- *"Required param added in 3.0.x as a critical bug fix, not subject to the breaking changes policy"* — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L26-L27. È il parametro `I18NEXT_INSTANCE`. Con ogni probabilità il bug è la condivisione dell'istanza globale tra richieste SSR descritta nel JSDoc del token (core-libs/core/src/i18n/i18next/i18next-instance.ts:L10-L18); è un'inferenza, il commento non lo dice.
- *"Disable the periodical reloading. Otherwise SSR would not finish due to the pending task `setInterval()`"* — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L66-L68.
- *"It's needed, because Angular Universal doesn't support relative URLs in HttpClient."* — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L73-L78, con i link alle issue angular#19224 e universal#858.
- *"a workaround for https://github.com/i18next/i18next-http-backend/issues/82"* — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L37-L37.
- *"By default it configures and uses the `I18nextHttpBackendService`."* — core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.service.ts:L15-L15. Il nome è sbagliato: la classe reale si chiama `I18nextHttpBackendInitializer`.
- *"type CxDatePipe, not DatePipe, due to conflict with Angular's DatePipe - problem occurs for the backward compatibility compiler of Ivy"* — core-libs/core/src/i18n/date.pipe.ts:L12-L12.
- *"TODO: Replace `any` to match strict types from angular in 4.0"* — core-libs/core/src/i18n/date.pipe.ts:L21-L22: un TODO rimasto da molte major.

- *"the first language in the array is the default one"* — core-libs/core/src/i18n/config/i18n-config-initializer.ts:L28-L28.
- *"fallback to main key as a chunk"* — core-libs/core/src/i18n/translation-chunk.service.ts:L46-L46.
- *"Only returns the date if we have a valid format. We do this to avoid loads of console errors coming from the datePipe while the user is typing"* — core-libs/storefront/shared/components/form/date-picker/date-picker.component.ts:L60-L68.

8. Extension point

EXTENSION POINT	COME SI USA	RIFERIMENTO	PUBBLICO?
`provideConfig({ i18n: { resources, chunks } })`	aggiungere o sovrascrivere traduzioni (deep merge, `addResourceBundle` con overwrite)	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L79-L92	sì (`I18nConfig` in core-libs/core/src/i18n/index.ts:L7-L7)
`i18n.backend.loadPath` / `loader`	lazy loading via HTTP o `import()`	core-libs/core/src/i18n/config/i18n-config.ts:L30-L72	sì
`TranslationService`	sostituire l'intero motore (es. un altro framework i18n) con `{ provide: TranslationService, useClass: ... }`	core-libs/core/src/i18n/translation.service.ts:L9-L35; binding core-libs/core/src/i18n/i18n.module.ts:L46-L46	sì
`I18nextTranslationService`	estendere, es. `protected getFallbackValue` per cambiare il testo delle chiavi mancanti	core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L100-L106	sì
`TranslationChunkService`	cambiare la regola chiave → chunk (`getChunkNameForKey`)	core-libs/core/src/i18n/translation-chunk.service.ts:L41-L49	sì
`I18nextInitializer`	cambiare le opzioni di i18next (`protected getI18nextConfig`) o la sincronizzazione della lingua	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L49-L106	sì
`I18nConfigInitializer`	cambiare la regola del fallback dinamico (`protected resolveConfig`)	core-libs/core/src/i18n/config/i18n-config-initializer.ts:L24-L33	sì
`I18nextBackendInitializer` (multi)	aggiungere un altro plugin backend di i18next con `hasMatch` (dichiarato nella classe) e `getPriority` (opzionale, ereditato dall'interfaccia `Applicable`)	core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.initializer.ts:L15-L26, core-libs/core/src/util/applicable.ts:L44-L44; selezione core-libs/core/src/util/applicable.ts:L63-L86	sì
`I18NEXT_HTTP_BACKEND_CLIENT`	sostituire la funzione `request` (header, cache, retry)	core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L16-L44	sì

EXTENSION POINT	COME SI USA	RIFERIMENTO	PUBBLICO?
`I18nextHttpBackendInitializer`	estendere `protected getBackendConfig`/`getLoadPath` (es. altre opzioni del plugin)	`core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L55-L91`	sì
`I18NEXT_INSTANCE`	accedere all'istanza per richiesta (mai all'import globale)	`core-libs/core/src/i18n/i18next/i18next-instance.ts:L19-L22`	sì
`CmsComponentMapping.i18nKeys`	precaricare i chunk dei componenti CMS di una pagina	`core-libs/core/src/cms/config/cms-config.ts:L76-L76`	sì (cms)
`CxNumericPipe`	estendere (metodi `protected`)	`core-libs/core/src/i18n/numeric.pipe.ts:L26-L52`	sì
`I18nTestingModule` e mock	test di componenti con chiavi "parlanti"	`core-libs/core/src/i18n/testing/i18n-testing.module.ts:L13-L20`	sì

Non pubblici, quindi da non usare: `I18NEXT_LOGGER_PLUGIN` e `i18nextProviders` (non esportati da `core-libs/core/src/i18n/i18next/index.ts:L7-L10`), `initI18nConfig` (di `i18n.module` si esporta solo `I18nModule`, `core-libs/core/src/i18n/index.ts:L7-L18`), `defaultI18nConfig`, `mockTranslate` (`core-libs/core/src/i18n/testing/index.ts:L7-L10`).

Esempio: un backend personalizzato. Lo schema che segue usa solo i simboli pubblici visti sopra. È uno schema illustrativo, non codice del repo:

```ts
// Schema illustrativo (non presente nel repo)
@Injectable({ providedIn: 'root' })
export class MyBackendInitializer implements I18nextBackendInitializer {
  protected i18next = inject(I18NEXT_INSTANCE);
  protected config = inject(I18nConfig);
  hasMatch() { return !!this.config.i18n?.backend; }
  getPriority() { return Priority.HIGH; } // batte i due backend di default
  initialize(): InitOptions {
    this.i18next.use(/* plugin backend i18next */);
    return { backend: { /* opzioni del plugin */ } };
  }
}
// provider: { provide: I18nextBackendInitializer, useExisting: MyBackendInitializer, multi: true }
```

Ricette in Parte III:

- traduzioni, override delle chiavi, chunk per una feature custom: 53 — Ricette: feature lib, carrello/checkout, stili, traduzioni;
- override di servizi via DI e config: 50 — Ricette: configurazione, DI, interceptor, eventi;
- cosa è API pubblica e cosa no: 54 — API pubblica vs interna, upgrade.

9. Trappole, casi limite, comportamenti controintuitivi

1. `loadPath` **e** `loader` **insieme: vince** `loader`**.** Entrambi i backend corrispondono e nessuno dei due ha `getPriority`. `resolveApplicable` fa vincere, a parità di priorità, il provider registrato per ultimo, che è `I18nextResourcesToBackendInitializer`; `loadPath` viene ignorato senza avvisi (core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.providers.ts:L12-L23, core-libs/core/src/util/applicable.ts:L72-L86).
2. `i18n.backend: {}` **senza** `loadPath` **né** `loader`**.** Nessun initializer corrisponde e `I18nextBackendService` restituisce `{}`: niente backend e niente errore (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L63-L68, core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.service.ts:L33-L38).
3. **Le traduzioni eager non disattivano il lazy loading.** Proprio perché vengono aggiunte nel callback di init e non come opzione `resources` di i18next, risorse eager e chunk lazy convivono (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L38-L43, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1853-L1857).
4. `fallbackLang: false`**.** Impostato staticamente disattiva il fallback dinamico (il test è `!== undefined`) e arriva a i18next come `fallbackLng: false`, quindi nessuna lingua di ripiego (core-libs/core/src/i18n/i18n.module.ts:L30-L32, core-libs/core/src/i18n/i18next/i18next-initializer.ts:L52-L52).
5. **Namespace dimenticato in** `chunks`**.** Il chunk diventa il namespace stesso. Con un backend HTTP si vedrà una richiesta a `.../<namespace>.json`, probabilmente in 404, invece del chunk previsto (core-libs/core/src/i18n/translation-chunk.service.ts:L41-L49). Al contrario, le risorse eager sono indicizzate per *chunk*: se in `resources` metti `{ en: { order: {...} } }` ma la chiave `orderDetails.x` è mappata al chunk `order`, la chiave risolta è `order:orderDetails.x`, che cerca `orderDetails.x` dentro il bundle `order`.
6. **Namespace duplicati tra chunk.** Vince il primo in ordine di `Object.keys`, e l'avviso compare solo in dev mode (core-libs/core/src/i18n/translation-chunk.service.ts:L22-L36).
7. **Chiavi mancanti silenziose in produzione.** In produzione la chiave mancante diventa uno spazio non separabile, senza log; solo in dev mode si vede `[chunk:key]` con un warn (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L100-L119).
8. **Prima dell'init non arriva nulla.** Se ci si sottoscrive a `translate()` prima che i18next sia inizializzato, l'Observable non emette e si "risveglia" solo al primo `languageChanged`; `TranslatePipe` restituisce `translatedValue` non ancora assegnato (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L57-L59, core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L87-L89, core-libs/core/src/i18n/translate.pipe.ts:L65-L66). Dato che la Promise dell'`APP_INITIALIZER` si risolve solo a init completato (§3.3), il caso riguarda soprattutto codice che si sottoscrive durante l'inizializzazione dell'app.
9. `translate()` **non completa mai.** Chi lo usa nel codice TypeScript deve gestire l'unsubscribe, oppure usare `take(1)` o `firstValueFrom` quando basta un valore. La pipe lo fa in `ngOnDestroy` (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L87-L89, core-libs/core/src/i18n/translate.pipe.ts:L91-L95). Attenzione: con `whitespaceUntilLoaded = true` e un chunk non ancora caricato, il primo valore è lo spazio non separabile (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L81-L84).
10. **Array letterali nella pipe.** `{{ ['a.b', 'c.d'] | cxTranslate }}`: la chiave array è confrontata per riferimento, quindi se il template produce un array nuovo a ogni ciclo la pipe si risottoscrive a ogni change detection (core-libs/core/src/i18n/translate.pipe.ts:L70-L72). DA VERIFICARE: se il compilatore Angular memoizzi gli array letterali costanti nei template (pure function) non è stato verificato qui; il rischio concreto riguarda gli array costruiti nel componente con un getter.
11. `escapeValue: false`**.** i18next non esegue l'escaping dei valori interpolati (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L54-L57). Con il binding di testo `{{ }}` l'escaping lo fa Angular; con

`[innerHTML]="'x' | cxTranslate: { name }"` un parametro controllato dall'utente finisce nell'HTML, sanificato solo dal sanitizer di Angular.

12. `cxDate` **e** `cxNumeric` **non reagiscono al cambio lingua.** Sono pipe pure: se cambi lingua e la data non cambia, il testo resta formattato nella lingua precedente finché l'input non cambia (core-libs/core/src/i18n/date.pipe.ts:L13-L13, core-libs/core/src/i18n/numeric.pipe.ts:L12-L12, angular:packages/core/src/metadata/directives.ts:L752-L752).
13. **Lingua non ancora nello store.** `getActiveLang()` restituisce `''`; in Angular `getLocaleData('')` non trova nulla, il padre `''` neanche, e poiché `'' !== 'en'` `findLocaleData` lancia. La pipe ripiega quindi su `'en'` e in dev mode emette un warn con locale `''` (core-libs/core/src/i18n/date.pipe.ts:L40-L47, angular:packages/core/src/i18n/locale_data_api.ts:L105-L123, angular:packages/core/src/i18n/locale_data_api.ts:L49-L72).
14. **Locale non registrati.** Senza `registerLocaleData` della lingua, `cxDate` e `cxNumeric` usano `'en'`. L'app demo registra solo de, ja, zh (core-libs/core/src/i18n/date.pipe.ts:L29-L38, projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L29-L31). I codici con underscore (`es_CO`, `zh_TW`) vengono normalizzati da Angular in `es-co` / `zh-tw`, e si prova anche il padre (angular:packages/core/src/i18n/locale_data_api.ts:L181-L183).
15. **Il placeholder del date picker non è localizzato.** È sempre `yyyy-mm-dd` (core-libs/storefront/shared/components/form/date-picker/date-picker.service.ts:L24-L37).
16. `I18nTestingModule` **non ha** `cxNumeric`**.** I test di componenti che usano `cxNumeric` devono dichiararlo a parte (core-libs/core/src/i18n/testing/i18n-testing.module.ts:L13-L20). Inoltre `MockTranslationService` completa dopo il primo valore, a differenza dell'implementazione reale (core-libs/core/src/i18n/testing/mock-translation.service.ts:L14-L28).
17. **Il warn "No backend was added...".** Senza backend, ogni `loadNamespaces` passa dal connettore, che registra `'No backend was added via i18next.use. Will not load resources.'`; si vede solo con `debug: true`, perché `warn` del logger i18next è riservato al debug (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1548-L1552, deps:i18next-25.7.4/package/dist/esm/i18next.js:L219-L231).
18. **HTTP backend in SSR e origine.** Un `loadPath` relativo viene reso assoluto con `windowRef.location.origin`: se l'origine vista dal server è sbagliata (proxy, host interno) le richieste vanno all'host sbagliato (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L79-L91). DA VERIFICARE: come venga popolato `location.origin` lato server è materia di 39 — SSR ottimizzato.
19. **Cambi di lingua rapidi.** `changeLanguage` notifica `languageChanged` solo per l'ultima lingua richiesta (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1955-L1962).
20. **Traduzioni delle feature lazy caricate eager.** Nell'app demo le risorse di una feature lazy sono fornite nel modulo root della feature, quindi finiscono nel bundle iniziale anche se il codice della feature è lazy (projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L26-L48). Per evitarlo serve un backend lazy.
21. **Script di assets rotti.** `generate-translations-ts-2-properties.ts` e `generate-translations-properties-2-ts.ts` importano `{ translations }` da `./src/translations/translations` (core-libs/assets/generate-translations-ts-2-properties.ts:L8-L8, core-libs/assets/generate-translations-properties-2-ts.ts:L8-L8), ma quel file esporta solo `translationChunksConfig` e `translationsXx` (core-libs/assets/src/translations/translations.ts:L7-L84): così non compilano. `properties-2-ts`, inoltre, presuppone file di chunk `.ts` della forma `export const x = {...}` (core-libs/assets/generate-translations-properties-2-ts.ts:L56-L71), mentre le cartelle lingua contengono JSON importati da `index.ts` (core-libs/assets/src/translations/en/index.ts:L7-L18). Anche il README di assets cita `import { translations } from '@spartacus/assets';` (core-libs/assets/README.md:L1-L5), un export che non esiste. La cartella `lang/properties` contiene solo `dummy.txt`.
22. **Piccoli bug negli script.** In `ts-2-properties`, `_prefix + '=' + jsonObj[key] || ''` applica `|| ''` all'intera stringa concatenata, quindi il fallback non ha mai effetto (core-libs/assets/generate-translations-ts-

2-properties.ts:L28-L28). In `properties-2-ts`, `property.split('=')[1]` tronca i valori che contengono `=` (core-libs/assets/generate-translations-properties-2-ts.ts:L14-L15).

23. **JSON lazy solo per** `en`**.** `ts-2-json` produce gli asset solo per l'inglese (core-libs/assets/generate-translations-ts-2-json.ts:L32-L40): chi vuole il lazy loading HTTP per altre lingue deve generarli da sé.
24. **Il validatore non guarda** `core-libs`**.** Le traduzioni core in `core-libs/assets/src/translations/**` non vengono validate da `build:libs` (scripts/i18n/validate-translations-json-files.ts:L97-L105).
25. **Tipi duplicati.** `TranslationChunksConfig` è definita sia in core sia in assets (core-libs/core/src/i18n/translation-resources.ts:L16-L18, core-libs/assets/src/translations/translation-chunks-config.ts:L7-L9), e `translationChunksConfig` del core non è tipizzato, mentre quelli delle feature lib sì (core-libs/assets/src/translations/translations.ts:L7-L7, feature-libs/order/assets/translations/translations.ts:L7-L9).

---

10. Come fare debug

- **Vedere cosa fa i18next.** Imposta `provideConfig({ i18n: { debug: true } })`: l'opzione arriva a i18next (core-libs/core/src/i18n/i18next/i18next-initializer.ts:L53-L53) e sblocca `log`/`warn` del logger, che passano dal plugin a `LoggerService` (core-libs/core/src/i18n/i18next/i18next-plugins/i18next-logger-plugin.ts:L17-L40). Vedrai `initialized`, `languageChanged` (deps:i18next-25.7.4/package/dist/esm/i18next.js:L1834-L1840, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1955-L1963) e i warn del connettore. Il JSDoc avverte: "Don't use in production!" (core-libs/core/src/i18n/config/i18n-config.ts:L80-L83).
- **Chiavi mancanti.** In dev mode compaiono come `[chunk:key]` nella UI, con un warn `Translation keys missing [...] Attempted to load 'key' from chunk 'chunk'` (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L108-L119). Se il chunk indicato non è quello atteso, controlla `i18n.chunks` e cerca nei log l'avviso sui duplicati (core-libs/core/src/i18n/translation-chunk.service.ts:L51-L65).
- **Ispezionare l'istanza.** Da un componente o servizio di debug, `inject(I18NEXT_INSTANCE)` restituisce l'istanza per root injector (core-libs/core/src/i18n/i18next/i18next-instance.ts:L19-L22); proprietà come `isInitialized` sono quelle che il servizio di traduzione controlla (core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L57-L57).
- **Testo vuoto o "spazio" persistente.** Se resta uno spazio non separabile: in produzione può essere una chiave mancante (§9.7); in qualunque modalità può essere un chunk che non arriva (backend che fallisce: con l'HTTP backend l'errore passa da `HttpClient`, core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L33-L41, ed è quindi visibile nel tab Network).
- **Backend HTTP in SSR.** Controlla che l'URL calcolato sia assoluto e con l'origine giusta (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L79-L91); se il rendering SSR non termina, verifica di non aver sovrascritto `reloadInterval` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L66-L68).
- **Date in inglese inattese.** Cerca in dev mode il warn `cxDate pipe: No locale data registered for '<lang>'` (core-libs/core/src/i18n/date.pipe.ts:L49-L55) e aggiungi `registerLocaleData` per quella lingua.
- **Backend sbagliato.** Se hai sia `loadPath` sia `loader`, ricorda §9.1. Per capire quale vince, metti un breakpoint in `I18nextBackendService.initialize` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.service.ts:L33-L38).
- **Nei test** usa `I18nTestingModule`: le chiavi diventano stringhe leggibili come `key name:value` (core-libs/core/src/i18n/testing/mock-translate.ts:L7-L18), comode da cercare nel DOM. Vedi 44 — Test e

tooling.

---

11. Domande di autoverifica

1. **Perché `I18NEXT_INSTANCE` usa `createInstance()` invece dell'import di default di i18next?** Perché in SSR ogni richiesta ha il proprio root injector e quindi la propria istanza; l'istanza globale sarebbe condivisa tra richieste concorrenti. — core-libs/core/src/i18n/i18next/i18next-instance.ts:L10-L22
2. **Se non configuro `i18n.fallbackLang`, quale lingua di fallback userà i18next?** La prima lingua di `context.language`, calcolata da `I18nConfigInitializer` dopo che lo scope `context.language` è stabile. — core-libs/core/src/i18n/i18n.module.ts:L23-L34, core-libs/core/src/i18n/config/i18n-config-initializer.ts:L24-L33
3. **Perché Spartacus non passa le traduzioni statiche nell'opzione `resources` di `init`?** Perché con `resources` i18next non userebbe più il backend per i chunk lazy; Spartacus le aggiunge con `addResourceBundle` nel callback di init. — core-libs/core/src/i18n/i18next/i18next-initializer.ts:L38-L43, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1853-L1857
4. **Come viene risolta la chiave `searchBox.placeholder` con la config di `@spartacus/assets`?** Il namespace è `searchBox`, mappato al chunk `common`; la chiave passata a i18next è `common:searchBox.placeholder`. — core-libs/core/src/i18n/translation-chunk.service.ts:L41-L49, core-libs/assets/src/translations/translations.ts:L7-L27, core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L121-L123
5. **Perché `translate()` deve emettere in modo sincrono quando il chunk è già caricato, e come ci riesce?** Un'emissione differita innescherebbe change detection aggiuntive che, con `switchMap`, possono causare loop infiniti. Ci riesce perché `loadNamespaces` chiama il callback in modo sincrono quando non c'è nulla da caricare. — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L35-L40, deps:i18next-25.7.4/package/dist/esm/i18next.js:L1555-L1559
6. **Cosa mostra `cxTranslate` per una chiave mancante, in dev e in produzione?** In dev `[chunk:key]` più un warn; in produzione uno spazio non separabile, senza log. — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L100-L119
7. **Configuro sia `backend.loadPath` sia `backend.loader`: quale backend viene usato?** `I18nextResourcesToBackendInitializer` (il `loader`): entrambi corrispondono con pari priorità e vince il provider registrato per ultimo. — core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.providers.ts:L12-L23, core-libs/core/src/util/applicable.ts:L72-L86
8. **Perché il backend HTTP imposta `reloadInterval: false`?** Per non lasciare un `setInterval` pendente, che impedirebbe all'SSR di terminare. — core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L66-L68
9. **Cambio lingua da `en` a `de`: perché i testi cambiano ma le date formattate con `cxDate` possono restare in inglese?** `cxTranslate` è impura e il servizio ritraduce a ogni `languageChanged`; `cxDate` è pura e si ricalcola solo se cambia l'input. Inoltre senza `registerLocaleData` per `de` ripiega comunque su `en`. — core-libs/core/src/i18n/translate.pipe.ts:L25-L29, core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L87-L89, angular:packages/core/src/metadata/directives.ts:L752-L752, core-libs/core/src/i18n/date.pipe.ts:L29-L38
10. **Come si fa a far partire il download dei chunk di una pagina CMS prima che i componenti vengano creati?** Dichiarando `i18nKeys` nel mapping del componente CMS: quando le guard della pagina passano, `CmsPageGuardService` chiama `CmsI18nService.loadForComponents`, che chiama `TranslationService.loadChunks`. — core-libs/core/src/cms/config/cms-config.ts:L76-L76, core-libs/storefront/cms-

structure/guards/cms-page-guard.service.ts:L81-L85, core-libs/storefront/cms-structure/services/cms-i18n.service.ts:L21-L28

TAVOLA 37.1 **4.1 Bootstrap: dalla config all'istanza i18next pronta**

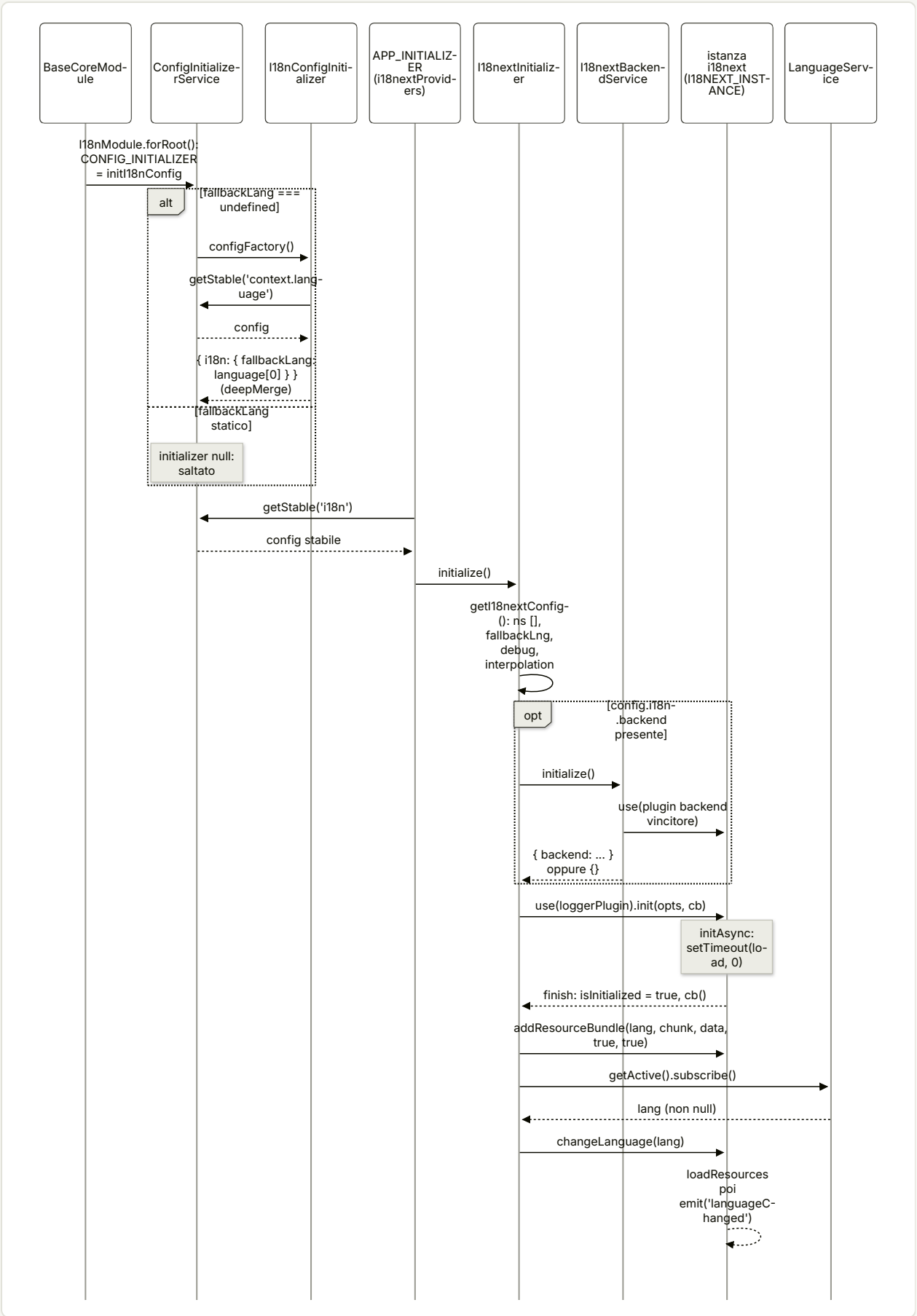
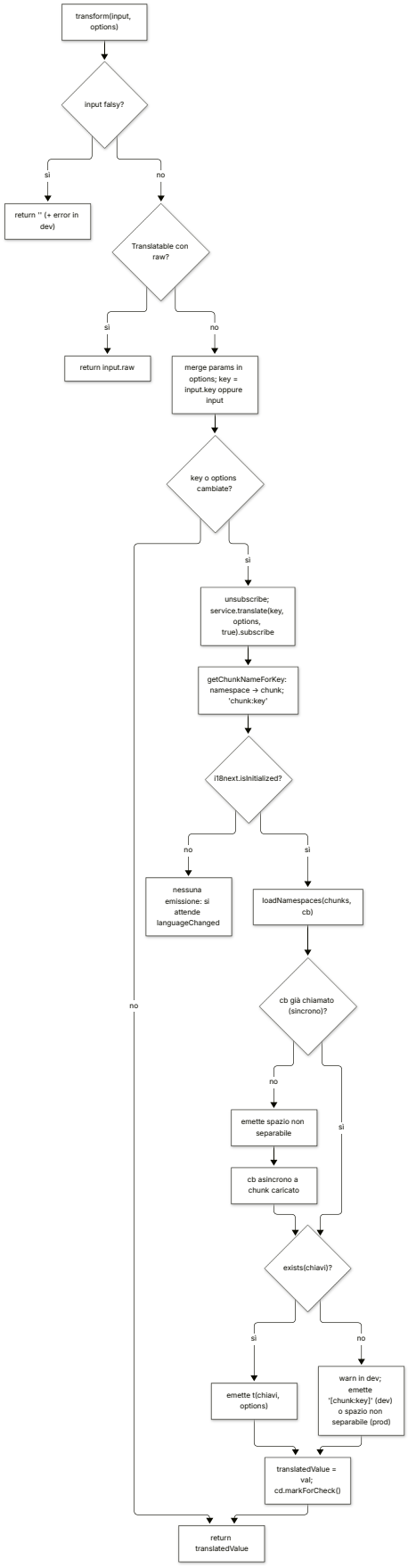


TAVOLA 37.2 **4.2 {{ 'key' | cxTranslate }}: dal template al testo**



38

Stili e theming

`@spartacus/styles`, una libreria Sass distribuita come sorgente, e il theming.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Spartacus separa nettamente la logica (librerie Angular in `@spartacus/core`, `@spartacus/storefront` e nelle feature lib) dall'aspetto grafico, che vive quasi per intero in un pacchetto a sé: `@spartacus/styles`, sorgente in `core-libs/styles/`. Il pacchetto non contiene TypeScript né CSS precompilato: è una libreria **Sass** distribuita come sorgente. Il `package.json` dichiara come entry Sass `./_index.scss` (campo `"sass"`) — `core-libs/styles/package.json:L2-L3`, `core-libs/styles/package.json:L31-L31` — il progetto Nx `storefrontstyles` ha solo un target di test e nessun target di build — `core-libs/styles/project.json:L2-L14` — e in fase di release i file vengono copiati tali e quali (`cp -r core-libs/styles/* ../.`) — `ci-scripts/release-packer.sh:L42-L43`. La compilazione avviene quindi **nell'applicazione consumer**, dentro la pipeline di build Angular, che vede tutte le variabili Sass e può modificarle prima che il CSS venga emesso.

Questa scelta abilita il modello di personalizzazione di Spartacus, che lavora su tre livelli:

1. **Compile time (Sass)**: variabili `!default` del tema (colori, tipografia, flag di componente come `$useTabsOnPDP`), liste di selettori da emettere (`$<area>-components-allowlist`, `$skipComponentStyles`, `$page-template-allowlist`) e placeholder `%cx-*` ridefinibili — `core-libs/styles/scss/theme/santorini/_variables.scss:L1-L9`, `core-libs/styles/scss/_components.scss:L1-L53`.
2. **Runtime (CSS custom property)**: `root.scss` traduce la mappa Sass `$theme-colors` in proprietà `--cx-color-*` su `:root`, che i componenti leggono con `var(...)`; basta ridefinirle su una classe per ottenere un tema diverso senza ricompilare — `core-libs/styles/scss/root.scss:L1-L6`.
3. **Runtime (classi sul DOM gestite da TypeScript)**: `ThemeService` mette sul root component la classe del *site theme* attivo (per esempio `cx-theme-high-contrast-dark`) — `core-libs/storefront/layout/theme/theme.service.ts:L52-L61` —, `FeatureStylesService` aggiunge classi `cxFeat_<flag>` per gli stili condizionati da feature toggle — `core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L95` — e `DirectionService` imposta l'attributo `dir` su `<html>` in base alla lingua — `core-libs/storefront/layout/direction/direction.service.ts:L89-L95`.

Sotto tutto c'è una copia **vendorizzata di Bootstrap 4.6.2** (`core-libs/styles/vendor/bootstrap/`) — `core-libs/styles/vendor/bootstrap/scss/bootstrap.scss:L1-L7` — di cui Spartacus riusa funzioni, mixin, variabili e una selezione di parziali (reboot, grid, utilities, buttons, forms, modal…), personalizzandoli nei blocchi `cxbase`.

Il capitolo copre: struttura e ordine di import del pacchetto Sass, il meccanismo placeholder/allowlist, temi compilati (santorini, santorini-updated, sparta, lambda) e temi runtime (classi CMS e high-contrast), feature flag e versioning in SCSS, integrazione nell'app e nelle feature lib, e i servizi TypeScript che collegano il DOM al CSS (`ThemeService`, `FeatureStylesService`, `DirectionService`). Il lato "dati" del site theme (store NgRx, persistenza, precedenza CMS) è trattato in 13 — Site context e site theme; i breakpoint TypeScript in 18 — Layout, breakpoint, launch dialog e accessibilità; i feature toggle in generale in 42 — Feature toggle e deprecazioni.

2. Mappa dei file

PATH	RUOLO
`core-libs/styles/package.json`	Pacchetto `@spartacus/styles`, entry Sass `./_index.scss`, peer dependency Font Awesome / Open Sans / ng-select
`core-libs/styles/_index.scss`	Entry di default: `app` → `root` → `page-template` → `components`
`core-libs/styles/theme-santorini.scss`, `core-libs/styles/theme-sparta.scss`, `core-libs/styles/theme-lambda.scss`	Entry alternativi: variabili di un tema + `./index`
`core-libs/styles/scss/_core.scss`	Solo strumenti (features, versioning, functions, mixins, theme): da importare in ogni bundle
`core-libs/styles/scss/_theme.scss`	Seleziona il tema compilato (santorini)
`core-libs/styles/scss/app.scss`	Stili globali: font, Font Awesome, ng-select, blocchi cxbase, layout, misc, site theme
`core-libs/styles/scss/root.scss`	Custom property su `:root` e override a11y via classi `cxFeat_*`
`core-libs/styles/scss/_components.scss`	Allowlist, `$skipComponentStyles`, cicli `@extend %<selettore>` e `%<selettore>__body`
`core-libs/styles/scss/_page-template.scss`	Allowlist/blocklist dei page template e regole `cx-page-layout.<Template>`
`core-libs/styles/scss/_features.scss`	`getFeatureCssClass`, mixin `forFeature`
`core-libs/styles/scss/_versioning.scss`	Mixin `forVersion` (deprecato) e funzioni di versione
`core-libs/styles/scss/_site-theme.scss`	Import dei temi high-contrast
`core-libs/styles/scss/cxbase/_variables.scss`	Variabili aggiuntive rispetto a Bootstrap ("NOT a theme")
`core-libs/styles/scss/cxbase/_mixins.scss`	Mixin Bootstrap + mixin Spartacus (`type`, `weight`, `visible-focus`, …)
`core-libs/styles/scss/cxbase/functions/merge.scss`, `core-libs/styles/scss/cxbase/functions/spacer.scss`	Funzioni `mergeAll` e `spacer`
`core-libs/styles/scss/cxbase/blocks/buttons.scss` (e gli altri `blocks/*`)	Personalizzazione globale di `.btn`, form, modal, alert, …
`core-libs/styles/scss/components/index.scss`	Importa le nove aree di componenti (placeholder `%cx-*` + allowlist)
`core-libs/styles/scss/layout/_index.scss`	page-fold, ghost, screen-reader, placeholder dei page template
`core-libs/styles/scss/theme/santorini/_variables.scss`	Tema di default (+ classe `.santorini-updated`)
`core-libs/styles/scss/theme/santorini-updated/_variables.scss`	Tema aggiornato, non attivo di default
`core-libs/styles/scss/theme/sparta/_variables.scss`, `core-libs/styles/scss/theme/lambda/_variables.scss`	Temi alternativi (Lambda è un tema di test)
`core-libs/styles/scss/theme/high-contrast/high-contrast.scss` + `dark/`, `light/`, `mixins/`	Temi runtime ad alto contrasto e mixin `cx-highContrastTheme*`
`core-libs/styles/vendor/bootstrap/scss/`	Bootstrap 4.6.2 vendorizzato

PATH	RUOLO
core-libs/styles/test/scss.spec.js	Runner Jest + sass-true per i test *.spec.scss
core-libs/storefront/layout/theme/theme.service.ts, core-libs/storefront/layout/theme/theme.module.ts	Classe del site theme sul root component (APP_BOOTSTRAP_LISTENER)
core-libs/storefront/layout/direction/direction.service.ts, core-libs/storefront/layout/direction/direction.module.ts	Attributo dir su <html> (APP_INITIALIZER)
core-libs/storefront/layout/direction/config/direction.model.ts, core-libs/storefront/layout/direction/config/default-direction.config.ts	DirectionMode, Direction, default detect/rtlLanguages
core-libs/core/src/features-config/services/feature-styles.service.ts, core-libs/core/src/features-config/utils/use-feature-styles.ts	Classi cxFeat_<flag> sul root element
core-libs/storefront/layout/main/storefront.component.ts	Registra gli stili feature-flag globali; caso speciale su <html>
projects/storefrontapp/src/styles.scss, projects/storefrontapp/src/styles-config.scss, projects/storefrontapp/src/cms-themes.scss	Integrazione nell'app demo
core-libs/schematics/src/add-spartacus/index.ts	Schematic che genera styles-config.scss e gli import nel foglio principale

3. Come funziona passo per passo

3.1 Packaging e test

@spartacus/styles dichiara come dipendenza runtime soltanto hamburgers e come peer dependency @fontsource/open-sans, @fortawesome/fontawesome-free 7.1.0 e @ng-select/ng-select — core-libs/styles/package.json:L16-L24. Il pacchetto si porta dietro i sorgenti scss/ (189 file tracciati) e vendor/ (91 file) — core-libs/styles/scss/, core-libs/styles/vendor/. I test sono scritti in Sass con sass-true: un unico spec Jest cerca tutti i test/**/*.spec.scss e li esegue con sassTrue.runSass — core-libs/styles/test/scss.spec.js:L1-L15.

DA VERIFICARE: il modo in cui la build Angular risolve @import '@spartacus/styles' attraverso il campo "sass" del package.json non è verificabile qui (i sorgenti di Angular CLI non sono in /tmp/angular, che contiene solo il framework).

3.2 Entry point e ordine di import

L'entry di default core-libs/styles/_index.scss è composto da quattro import, in quest'ordine: scss/app, scss/root, scss/page-template, scss/components — core-libs/styles/_index.scss:L1-L4. Ciascuno ha un compito preciso:

- app.scss carica font, il CSS di Font Awesome, il tema di default di ng-select, cxbase/index, i blocchi cx-base (buttons, forms, modal, alert, message, tooltip, tables, card, accordion, tabs, carousel, link, dropdown), i layout cxbase, paragraph, animations, misc/* e per ultimo site-theme — core-libs/styles/scss/app.scss:L1-L39.
- root.scss emette le custom property su :root (vedi §3.5) — core-libs/styles/scss/root.scss:L1-L45.

- `_page-template.scss` emette le regole `cx-page-layout.<Template>` (vedi §3.7) — core-libs/styles/scss/_page-template.scss:L21-L27.
- `_components.scss` emette le regole per ogni elemento `cx-*` (vedi §3.6) — core-libs/styles/scss/_components.scss:L37-L53.

Esistono tre entry alternativi che antepongono le variabili di un tema: `theme-sparta.scss` (font + variabili Sparta, poi `./index`) — core-libs/styles/theme-sparta.scss:L1-L4 —, `theme-santorini.scss` (font Santorini **prima** delle variabili, poi `./index`) — core-libs/styles/theme-santorini.scss:L1-L4 — e `theme-lambda.scss` (solo variabili, nessun font) — core-libs/styles/theme-lambda.scss:L1-L3. Nessun file del repo li referenzia (vedi §9).

3.3 Il "core" Sass: strumenti senza output

`scss/_core.scss` importa features, versioning, functions, mixins e theme — core-libs/styles/scss/_core.scss:L1-L5. È ciò che ogni bundle (app o feature lib) importa per primo, perché porta variabili, funzioni e mixin senza emettere (in teoria) regole CSS:

- `_functions.scss` → `cxbase/functions`, che carica le functions del Bootstrap vendorizzato più `spacer` e `merge` — core-libs/styles/scss/_functions.scss:L1-L1, core-libs/styles/scss/cxbase/_functions.scss:L1-L3.
- `_mixins.scss` → `cxbase/mixins` + mixin high-contrast — core-libs/styles/scss/_mixins.scss:L1-L2. `cxbase/_mixins.scss` importa a sua volta `../theme`, "because of some lower level bootstrap variables", poi i mixin di Bootstrap e quelli Spartacus (reset, weight, type, button-reset, placeholder, transformations, visible-focus) — core-libs/styles/scss/cxbase/_mixins.scss:L1-L15.
- `_theme.scss` punta al tema compilato. Di default è attivo `theme/santorini/variables`; `santorini-updated` è commentato ("to uncomment on next major or with impex"), sparta e lambda pure — core-libs/styles/scss/_theme.scss:L5-L12.

In realtà il file di variabili santorini emette anche CSS (la classe `.santorini-updated`, §3.4), quindi "senza output" va preso con cautela (vedi §9).

3.4 Tema, cxbase e Bootstrap: chi vince sulle variabili

La catena delle variabili è la parte più delicata. L'ordine reale, a partire da `_theme.scss`, è:

1. **Variabili del tema** (`santorini/_variables.scss`). Il file dichiara il contratto: si possono ridefinire variabili Bootstrap, cxbase e di componente, ma non aggiungerne di nuove — core-libs/styles/scss/theme/santorini/_variables.scss:L1-L9. I colori base usano `!default` (`$primary: #1f7bc0`, `$text: #14293a`, `$visual-focus: #6d9df7`, …) — core-libs/styles/scss/theme/santorini/_variables.scss:L31-L50; `$disabled` e `$disabled-accent` sono separati per superfici chiare e riempimenti, per consentire il 7:1 nei temi high-contrast — core-libs/styles/scss/theme/santorini/_variables.scss:L53-L58. Le mappe `$theme-colors` (24 chiavi) e `$type` (scala tipografica 1–8 come coppie dimensione/peso) sono invece assegnate **senza** `!default` — core-libs/styles/scss/theme/santorini/_variables.scss:L60-L85, core-libs/styles/scss/theme/santorini/_variables.scss:L102-L112.
2. **cxbase** (`@import '../../cxbase/variables'`, "bring in variable defaults") — core-libs/styles/scss/theme/santorini/_variables.scss:L114-L115. `cxbase/_variables.scss` si autodefinisce "NOT a theme": carica le variabili Bootstrap e aggiunge solo ciò che Bootstrap non ha — core-libs/styles/scss/cxbase/_variables.scss:L1-L13. Fonde in `$theme-colors` le chiavi `background`, `text`, `inverse`, `transparent` (mettendo le chiavi del tema per seconde, quindi vincenti) e dichiara con `!default` `$font-weight-semi: 600`, `$btn-text-transform`, `$direction: ltr`, `$header-height: 67px` — core-libs/styles/scss/cxbase/_variables.scss:L15-L50. Con santorini il default di `$font-weight-semi` è ignorato: il tema, prima di im-

portare cxbase, assegna senza `!default` `$font-weight-light/normal/semi/bold` come stringhe (`'light'`, `'normal'`, `'semi'`, `'bold'`), quindi il valore effettivo è la stringa `'semi'` — core-libs/styles/scss/theme/santorini/_variables.scss:L97-L100, core-libs/styles/scss/cxbase/_variables.scss:L30-L30.

3. **Bootstrap** inizializza `$theme-colors: () !default` e poi la fonde con i propri colori, quindi le chiavi del tema già definite prevalgono — core-libs/styles/vendor/bootstrap/scss/_variables.scss:L85-L86.
4. **Override globali del tema**, scritti dopo l'import di cxbase: `$grid-gutter-width: 30px`, `$input-height: 48px`, `$body-color: $text`, `$body-bg: $inverse`, … — core-libs/styles/scss/theme/santorini/_variables.scss:L114-L145. Il tema poi rimuove da `$colors` i colori nominali di Bootstrap (indigo, purple, …, white) — core-libs/styles/scss/theme/santorini/_variables.scss:L147-L163.
5. **Flag di componente**: il tema imposta `$spinWhileLoading`, `$spinWhileWaiting`, `$zoomThumbOnHover`, `$overlayWhileLoading`, `$useTabsOnPDP`, `$useAccordionOnly` a `true !default` — core-libs/styles/scss/theme/santorini/_variables.scss:L169-L174 — mentre i file dei componenti dichiarano gli stessi flag con default `false` — core-libs/styles/scss/components/product/details/_product-images.scss:L1-L4, core-libs/styles/scss/components/content/tab-paragraph-container/_tab-paragraph-container.scss:L1-L2. Poiché il tema viene valutato per primo, l'assegnazione `false !default` del componente non ha effetto. `$useTabsOnPDP` decide a compile time se `%cx-tab-paragraph-container` estende `%pdpTabs` o `%pdpFlat` — core-libs/styles/scss/components/content/tab-paragraph-container/_tab-paragraph-container.scss:L221-L227 — e `$useAccordionOnly` se includere il mixin `accordion()` — core-libs/styles/scss/components/content/tab-paragraph-container/_tab-paragraph-container.scss:L140-L142.
6. **Classe** `.santorini-updated`: in coda al file di variabili santorini c'è un blocco CSS vero e proprio con tutte le custom property del tema aggiornato ("for impex file, remove on next major") — core-libs/styles/scss/theme/santorini/_variables.scss:L176-L215.

Gli altri temi seguono lo stesso schema con palette diverse: `santorini-updated` usa `$primary: #055f9f`, `$visual-focus: #1f3a93` e una scala tipografica ridotta (h1 = 2rem, h2 = 1.5rem) — core-libs/styles/scss/theme/santorini-updated/_variables.scss:L31-L50, core-libs/styles/scss/theme/santorini-updated/_variables.scss:L80-L86; Sparta usa `$primary: #fe5757` e `$text: #212738`, e la sua `$theme-colors` non contiene `disabled`/`disabled-accent` — core-libs/styles/scss/theme/sparta/_variables.scss:L27-L73; Lambda è un "test theme" in Comic Sans MS che assegna a `$theme-colors` solo `brand`/`accent`, chiavi che poi cxbase e Bootstrap integrano con le proprie (`background`, `text`, …, `primary`, `secondary`, …) — core-libs/styles/scss/cxbase/_variables.scss:L20-L28, core-libs/styles/vendor/bootstrap/scss/_variables.scss:L85-L86 —; usa inoltre `$cx-hamburger-type: arrow-r` — core-libs/styles/scss/theme/lambda/_variables.scss:L1-L3, core-libs/styles/scss/theme/lambda/_variables.scss:L16-L16, core-libs/styles/scss/theme/lambda/_variables.scss:L32-L52.

Font. `_fonts.scss` punta ai font Sparta, che importano i CSS di `@fontsource/open-sans` (pesi 300/400/600/700, font self-hosted) — core-libs/styles/scss/_fonts.scss:L1-L2, core-libs/styles/scss/theme/sparta/_fonts.scss:L1-L4. I font Santorini invece fanno `@import url('#{$font-url}')` verso Google Fonts — core-libs/styles/scss/theme/santorini/_fonts.scss:L1-L1, core-libs/styles/scss/theme/santorini/_variables.scss:L26-L28. Poiché `app.scss` importa `fonts` (cioè Sparta) — core-libs/styles/scss/app.scss:L2-L2 —, l'entry di default usa i font self-hosted pur compilando le variabili santorini.

3.5 Custom property su `:root`

`root.scss` è il ponte fra Sass e runtime. Per ogni voce di `$theme-colors` emette `--cx-color-<nome>` — core-libs/styles/scss/root.scss:L1-L6 — e aggiunge costanti: `--cx-page-width-max: 1140px`, i pesi `--cx-font-weight-*`, `--cx-text-transform`, `--cx-direction`, durate di transizione/animazione, border radius e `--cx-border-style` — core-libs/styles/scss/root.scss:L7-L25. `--fa-width: auto` ripristina la larghezza automatica delle icone dopo il breaking change di Font Awesome 7 — core-libs/styles/scss/root.scss:L27-L30.

Tre blocchi sono condizionati da feature flag:

- `forFeature('a11yPreventWindowsHighContrastOverride', 'currentSelector')` annidato in `:root` applica `forced-color-adjust: none` sotto `forced-colors: active` — core-libs/styles/scss/root.scss:L36-L44. Essendo `currentSelector`, il selettore risultante è `:root.cxFeat_…`, cioè la classe deve stare su `<html>` — core-libs/styles/scss/_features.scss:L57-L61.
- `.cxFeat_a11yFocusIndicatorContrast:not(.cx-theme-high-contrast-dark):not(.cx-theme-high-contrast-light)` porta `--cx-color-visual-focus` a `#1f3a93 !important` — core-libs/styles/scss/root.scss:L47-L64.
- `.cxFeat_a11yPrimaryColorContrast:not(...)` porta `--cx-color-primary` a `#055f9f !important` — core-libs/styles/scss/root.scss:L66-L70.

Altre custom property globali nascono altrove: `--cx-spatial-*` (base 0.5rem) — core-libs/styles/scss/cxbase/_spatial.scss:L1-L8 —, `--btf-delay`/`--btf-min-height`/`--btf-margin-top` per le slot sotto la piega — core-libs/styles/scss/layout/_page-fold.scss:L3-L43 —, `--cx-spinner-size` con il keyframe `spin` e il placeholder `%spinner` — core-libs/styles/scss/cxbase/_animations.scss:L1-L25 —, `--cx-color-ghost*` per lo skeleton `.ghost` — core-libs/styles/scss/layout/_ghost.scss:L1-L24.

3.6 Componenti: placeholder `%cx-*`, allowlist e `@extend`

Gli stili di un componente non sono scritti sul suo selettore, ma su un **placeholder** Sass con lo stesso nome preceduto da `%`: per esempio `%cx-site-theme-switcher` o `%cx-storefront` — core-libs/styles/scss/components/misc/site-theme-switcher/_site-theme-switcher.scss:L3-L8, core-libs/styles/scss/components/layout/_storefront.scss:L3-L10. Un placeholder non produce CSS finché qualcuno non lo estende.

`components/index.scss` importa nove aree: content, product, misc, myaccount, layout, pwa, cds, digital-payments, cpq-quote — core-libs/styles/scss/components/index.scss:L1-L9. Ogni area, oltre ai placeholder, definisce una **allowlist** `!default` di selettori:

- `$content-components-allowlist` (cx-banner, cx-link, cx-media, cx-carousel, …, cx-tab) — core-libs/styles/scss/components/content/_index.scss:L10-L14;
- `$misc-components-allowlist` (cx-table, cx-icon, cx-spinner, …, cx-site-theme-switcher, cx-read-more) — core-libs/styles/scss/components/misc/_index.scss:L13-L16;
- `$layout-components-allowlist`, che include anche l'elemento HTML `header` (placeholder `%header`) — core-libs/styles/scss/components/layout/_index.scss:L6-L7, core-libs/styles/scss/components/layout/header/_header.scss:L3-L5;
- `$product-components-allowlist`, costruita con `mergeAll` a partire da `$product-list-allowlist` — core-libs/styles/scss/components/product/index.scss:L7-L24, core-libs/styles/scss/components/product/list/_index.scss:L11-L14;
- `$myaccount-components-allowlist` (14 selettori) — core-libs/styles/scss/components/myaccount/_index.scss:L16-L22;
- le aree minori pwa, cds, digital-payments, cpq-quote — core-libs/styles/scss/components/pwa/add-to-home-screen-banner/_index.scss:L3-L3, core-libs/styles/scss/components/cds/index.scss:L3-L3, core-libs/styles/scss/components/digital-payments/index.scss:L3-L3, core-libs/styles/scss/components/cpq-quote/index.scss:L3-L3.

`_components.scss` mette tutto insieme:

1. importa `components/index` (placeholder e allowlist reali) — core-libs/styles/scss/_components.scss:L1-L2;
2. **dopo**, ridichiara `$skipComponentStyles` e tutte le 11 allowlist come `() !default`: sono fallback per le liste non definite da nessuna area del core: `$epd-visualization-components-allowlist` è definita dall'integra-

tion lib epd-visualization, mentre `$user-components-allowlist` non è definita da nessun file del repo (è un residuo e resta vuota se non la imposta il consumer) — core-libs/styles/scss/_components.scss:L4-L19, core-libs/styles/scss/_components.scss:L11-L11, integration-libs/epd-visualization/_index.scss:L5-L9;

3. unisce le 11 liste in `$selectors` con `mergeAll`, che concatena con `join` — core-libs/styles/scss/_components.scss:L21-L35, core-libs/styles/scss/cxbase/functions/merge.scss:L1-L7;
4. per ogni selettore emette `<selettore> { @extend %<selettore> !optional; }` a meno che il selettore sia in `$skipComponentStyles` — core-libs/styles/scss/_components.scss:L37-L44;
5. per lo stesso elenco fa estendere a `body` i placeholder `%<selettore>__body` — core-libs/styles/scss/_components.scss:L46-L53.

Il secondo ciclo è il canale per gli stili che un componente deve applicare **fuori da sé**, a livello di pagina. Nel core styles l'unico placeholder `__body` è `%cx-searchbox__body`, che reagisce alla classe `searchbox-is-active` messa sul body dal componente di ricerca — core-libs/styles/scss/components/product/search/_searchbox.scss:L3-L21, core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L73-L73. La classe viene aggiunta da `setSearchBoxActive()` tramite `toggleBodyClass`, che opera su `this.winRef.document.body.classList` — core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L470-L471, core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L185-L196.

Il `!optional` rende innocuo un selettore senza placeholder corrispondente. Alcuni componenti offrono anche una classe che estende direttamente il placeholder (es. `.cx-icon { @extend %cx-icon; }`), usabile fuori dall'elemento `cx-icon` — core-libs/styles/scss/components/misc/_icons.scss:L9-L13.

Le feature lib replicano lo schema nel proprio bundle: epd-visualization importa `@spartacus/styles/scss/core`, definisce `$epd-visualization-components-allowlist` e ripete i due cicli — integration-libs/epd-visualization/_index.scss:L1-L29.

3.7 Page template

`_page-template.scss` funziona come `_components.scss`, ma per i layout di pagina: definisce `$page-template-blocklist: () !default` e `$page-template-allowlist` con 14 template — core-libs/styles/scss/_page-template.scss:L7-L19 — e per ogni template non blocklistato emette `cx-page-layout.<Template> { @extend %<Template> !optional; }` — core-libs/styles/scss/_page-template.scss:L21-L27. La classe `<Template>` su `cx-page-layout` è quella del template CMS della pagina (vedi 16 — CMS (storefront): rendering dinamico delle pagine).

I placeholder stanno in `layout/page-templates/`: `layout/_index.scss` importa page-fold, ghost, screen-reader e 13 file di template, e definisce `%CheckoutLoginPageTemplate` come estensione di `%LoginPageTemplate` — core-libs/styles/scss/layout/_index.scss:L1-L21. Le slot di base sono flex container in cui `--cx-flex-basis` (default 100%) regola la larghezza dei figli; `%LandingPage2Template` la porta al 50% per `Section2A`/`Section2B` — core-libs/styles/scss/layout/page-templates/_common-template.scss:L12-L20, core-libs/styles/scss/layout/page-templates/_landing-page-2.scss:L35-L38. `%ProductDetailsPageTemplate` usa un grid a due colonne da `lg` per la slot `Summary` — core-libs/styles/scss/layout/page-templates/_product-detail.scss:L1-L26. Le feature lib aggiungono allowlist dedicate, per esempio checkout con `$page-template-allowlist-multistep-checkout` — feature-libs/checkout/base/styles/layout/_index.scss:L5-L12.

3.8 Mixin e funzioni di base

- `spacer($key: 3)` restituisce `map-get($spacers, $key)` — core-libs/styles/scss/cxbase/functions/spacer.scss:L1-L3.
- `type($size, $weight)` legge la coppia dimensione/peso da `$type` ed emette `font-size: var(--cx-font-size, …)` e `line-height: var(--cx-line-height, …)`, delegando il peso a `weight` — core-libs/styles/scss/cxbase/mixins/_type.scss:L3-L29. `weight` traduce `light`/`semi`/`normal`/`bold` nelle custom property `--cx-font-weight-*` ed emette tale e quale un valore non stringa — core-libs/styles/scss/cxbase/mixins/_weight.scss:L1-L15. Il risultato è che ogni testo tipizzato con `type()` è sovrascrivibile a runtime con `--cx-font-size`/`--cx-line-height`.
- `visible-focus` disegna un outline `var(--cx-color-visual-focus)` di larghezza `var(--cx-visual-focus-width, 2px)` con offset 2px (−2px sotto `md`, salvo eccezioni) — core-libs/styles/scss/cxbase/mixins/_visible-focus.scss:L1-L24; `button` e `.btn` lo applicano su `:focus` — core-libs/styles/scss/cxbase/blocks/buttons.scss:L356-L361; quando l'utente usa il mouse, `%cx-storefront .mouse-focus` azzera `--cx-visual-focus-width` — core-libs/styles/scss/components/layout/_storefront.scss:L32-L44.
- Altri mixin: `button-reset`, `placeholder`, `reset($display)`, `cx-lighten`/`cx-darken` (filtro brightness) — core-libs/styles/scss/cxbase/mixins/_button-reset.scss:L1-L15, core-libs/styles/scss/cxbase/mixins/_placeholder.scss:L1-L14, core-libs/styles/scss/cxbase/mixins/_reset.scss:L1-L3, core-libs/styles/scss/cxbase/mixins/_transformations.scss:L1-L9.
- `.btn` usa `--cx-buttons-border-radius`, `--cx-text-transform`, `min-height: 48px` e `type('4')` — core-libs/styles/scss/cxbase/blocks/buttons.scss:L23-L32.
- L'hamburger usa la lib npm `hamburgers`: `$cx-hamburger-type` (default `squeeze`) seleziona il tipo e `.cx-hamburger` estende `.hamburger` e `.hamburger--<tipo>` — core-libs/styles/scss/misc/_hamburger.scss:L1-L20.

3.9 Feature flag in SCSS: `forFeature` + `FeatureStylesService`

Le modifiche stilistiche potenzialmente "breaking" sono protette da feature toggle. Il lato Sass è in `_features.scss`: `$CSS_FEATURE_FLAG_PREFIX: 'cxFeat_'` ("The same as the one used in the TS service FeatureStylesService") e `getFeatureCssClass` che lo antepone al nome — core-libs/styles/scss/_features.scss:L3-L14. Il mixin `forFeature($feature, $classTarget: 'ancestorSelector')` valida i parametri con `@error` e genera, con `@at-root`, `.cxFeat_<flag> &` (classe su un antenato) oppure, solo con `'currentSelector'`, `&.cxFeat_<flag>` (classe sull'elemento stesso, riservato al root element) — core-libs/styles/scss/_features.scss:L38-L61. Un test Sass verifica la prima forma — core-libs/styles/test/features.spec.scss:L1-L30. Nel core ci sono 39 `@include forFeature` in 19 file — core-libs/styles/scss/ (grep).

Il lato TypeScript deve mettere la classe sul DOM. `FeatureStylesService` usa lo stesso prefisso — core-libs/core/src/features-config/services/feature-styles.service.ts:L31-L36 — e un contatore per flag:

1. `useFeatureStyles(feature)` va chiamato nel costruttore del componente: richiede un injection context, chiama `registerUsage` e registra `unregisterUsage` su `DestroyRef.onDestroy` — core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40.
2. `registerUsage` aggiunge la classe al primo utilizzo se il flag è attivo — core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L95; `addClass` non fa nulla se il root element non è ancora noto — core-libs/core/src/features-config/services/feature-styles.service.ts:L130-L136.
3. Al bootstrap, `FeaturesConfigModule.forRoot` registra un `APP_BOOTSTRAP_LISTENER` che chiama `init(compRef)` — core-libs/core/src/features-config/features-config.module.ts:L41-L49; `init` memorizza l'elemento nativo del root component e aggiunge le classi già registrate (tipicamente dal costruttore del root component, che gira prima del listener) — core-libs/core/src/features-config/services/feature-styles.service.ts:L65-L77.

4. `unregisterUsage` rimuove la classe quando il contatore passa da 1 a 0 — core-libs/core/src/features-config/services/feature-styles.service.ts:L103-L115.

Angular invoca i listener `APP_BOOTSTRAP_LISTENER` con il `ComponentRef` del componente bootstrappato dopo `attachView` e un primo `tick()` — angular:packages/core/src/application/application_ref.ts:L775-L795. Nell'app demo il root component è `app-root` — projects/storefrontapp/src/app/app.component.ts:L15-L15, projects/storefrontapp/src/main.ts:L17-L17 —, quindi le classi `cxFeat_*` finiscono su `<app-root>`.

`StorefrontComponent` registra nel costruttore i flag stilistici globali (`a11yPreventWindowsHighContrastOverride`, `alignNavigationMenuWithHeader`, `cdsBottomHeaderSlotAdjustPosition`, `a11yFocusIndicatorContrast`, `a11yDisabledButtonContrast`, `a11yPrimaryColorContrast`) — core-libs/storefront/layout/main/storefront.component.ts:L117-L124 — e, per l'unico blocco su `:root`, aggiunge a mano `cxFeat_a11yPreventWindowsHighContrastOverride` su `document.documentElement` — core-libs/storefront/layout/main/storefront.component.ts:L141-L147. Un esempio d'uso a livello di componente è il bottone primario disabilitato, che con `a11yDisabledButtonContrast` usa `--cx-color-disabled-accent` — core-libs/styles/scss/cxbase/blocks/buttons.scss:L73-L84.

3.10 Versioning SCSS (`forVersion`, deprecato)

Prima dei feature flag nominati, Spartacus usava un versioning degli stili. `_versioning.scss` si apre con un avviso: `forVersion($level)` è deprecato e superato da `forFeature` — core-libs/styles/scss/_versioning.scss:L1-L3. La versione corrente è `$_fullVersion: (major: 221121, minor: 22, patch: 0)` con `$_majorVersion: 2211` — core-libs/styles/scss/_versioning.scss:L29-L37; il consumer può impostare `$styleVersion` (default `$_majorVersion`) e `$useLatestStyles` (default `false`) — core-libs/styles/scss/_versioning.scss:L39-L46. `forVersion($from, $to: 0)` (`@deprecated since 2211.20`) emette il contenuto solo se `$from` non supera la versione corrente e la versione è "stabile e valida" o è un breaking change richiesto — core-libs/styles/scss/_versioning.scss:L63-L77; i breaking change sono richiesti se `$useLatestStyles` è vero o `$styleVersion` supera `$_majorVersion` — core-libs/styles/scss/_versioning.scss:L228-L231. Nel core, `@include forVersion` compare solo nel commento d'esempio; il meccanismo resta coperto dai test — core-libs/styles/scss/_versioning.scss:L18-L18, core-libs/styles/test/versioning-test-util.scss:L1-L10.

3.11 Integrazione nell'applicazione

Il foglio principale dell'app demo importa, con commenti "ORDER IMPORTANT": `styles-config`, `@spartacus/styles/scss/core`, 15 parziali del Bootstrap vendorizzato, `@spartacus/styles` e infine `cms-themes` — projects/storefrontapp/src/styles.scss:L1-L29. L'ordine conta:

- `styles-config` viene prima di tutto perché contiene le variabili del consumer, che devono precedere le assegnazioni `!default` della libreria; nell'app demo imposta `$useLatestStyles: true` — projects/storefrontapp/src/styles-config.scss:L1-L2.
- `scss/core` prima di Bootstrap, perché i parziali Bootstrap leggono le variabili già personalizzate dal tema.
- `cms-themes` per ultimo: il commento di `styles.scss` dice che serve a far vincere le custom property `--cx-color-*` delle classi `.santorini`, `.lambda`, `.sparta` sui default di `:root`, ma rispetto a `:root` vincerebbero comunque, perché la classe è messa sul root component (un discendente di `<html>`) — core-libs/storefront/layout/theme/theme.service.ts:L52-L58. L'ordine conta contro le altre regole che dichiarano le stesse proprietà sullo **stesso** root component con pari specificità, come gli override `.cxFeat_*` di `root.scss`, che per questo usano `!important` — core-libs/styles/scss/root.scss:L55-L57, projects/storefrontapp/src/styles.scss:L27-L29, projects/storefrontapp/src/cms-themes.scss:L1-L60. L'app demo abilita `applyBaseSiteThemeFromCms`, quindi la classe arriva dal campo `theme` del base site — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L378-L385.

Ogni feature lib ha un bundle CSS separato in `project.json` (es. `lib-asm.scss` → bundle `asm`) e l'app silenzia la deprecazione Sass di `@import` — projects/storefrontapp/project.json:L43-L54, projects/storefrontapp/project.json:L137-L141. Quasi tutti i bundle (21 su 24) ripartono da `styles-config` e dall'entry della lib — projects/storefrontapp/src/styles/lib-cart.scss:L1-L2 —; `lib-customer-ticketing`, `lib-s4om` e `lib-subscription-billing` importano solo l'entry della lib, quindi lì non arriva neppure `styles-config` — projects/storefrontapp/src/styles/lib-customer-ticketing.scss:L1-L1. L'entry della lib, che a sua volta reimporta `@spartacus/styles/scss/core` più eventuali blocchi cxbase — feature-libs/cart/_index.scss:L1-L3. Ogni bundle è quindi una compilazione Sass indipendente: le variabili impostate nel foglio principale non arrivano nei bundle delle lib, a meno di stare in `styles-config`.

Nelle app nuove, lo schematic `add-spartacus` crea `styles-config.scss` con `$styleVersion: <featureLevel>` (default: major.minor della versione corrente) — core-libs/schematics/src/add-spartacus/index.ts:L70-L86, core-libs/schematics/src/shared/utils/package-utils.ts:L147-L149 — e inserisce nel foglio principale la stessa sequenza (config, core, Bootstrap, `@spartacus/styles/index`); con `--theme` aggiunge `@import '@spartacus/styles/scss/theme/${options.theme}'` — core-libs/schematics/src/add-spartacus/index.ts:L146-L173. L'opzione `theme` è una stringa libera ("E.g: santorini") — core-libs/schematics/src/add-spartacus/schema.json:L386-L389. Per gli schematics in generale vedi 43 — Schematics e migrazioni di Spartacus.

3.12 Site theme lato CSS: temi high-contrast

`app.scss` include per ultimo `_site-theme.scss`, che carica le variabili dei due temi ad alto contrasto e `high-contrast.scss` — core-libs/styles/scss/_site-theme.scss:L1-L3. `.cx-theme-high-contrast-dark` ridefinisce le custom property (`--cx-color-text: #ffffff`, `--cx-color-background: #000000`, `--cx-color-primary: #6bd3ff`, …) — core-libs/styles/scss/theme/high-contrast/dark/_variables.scss:L1-L27 —, `.cx-theme-high-contrast-light` usa testo nero su bianco con `--cx-color-primary: #004ccb` — core-libs/styles/scss/theme/high-contrast/light/_variables.scss:L1-L26 — e `high-contrast.scss` applica `color` e `background-color` a entrambe le classi — core-libs/styles/scss/theme/high-contrast/high-contrast.scss:L1-L5.

Dove le sole custom property non bastano, i componenti usano `cx-highContrastTheme($feature: null)`: senza flag genera `.cx-theme-high-contrast-dark &, .cx-theme-high-contrast-light &` annidato; con un flag usa `@at-root` per la forma combinata `.cx-theme-high-contrast-dark.cxFeat_<flag> &`; esistono varianti `-dark` e `-light` — core-libs/styles/scss/theme/high-contrast/mixins/_mixins.scss:L1-L40. Compare 67 volte nel core — `core-libs/styles/scss/` (grep). `%cx-storefront` disattiva i forced colors di Windows quando porta una delle classi high-contrast — core-libs/styles/scss/components/layout/_storefront.scss:L52-L59.

3.13 Site theme lato TypeScript: `ThemeService`

La catena che porta la classe sul DOM:

1. `BaseCoreModule` importa `SiteThemeModule.forRoot()`, che fornisce `SiteThemeService`, la config di default e un `APP_INITIALIZER` che chiama `SiteThemeInitializer.initialize()` — core-libs/core/src/base-core.module.ts:L43-L43, core-libs/core/src/site-theme/site-theme.module.ts:L18-L29, core-libs/core/src/site-theme/providers/site-theme-initializer-providers.ts:L10-L23.
2. `initialize` attende la config `context` stabile, ripristina il valore persistito e imposta il default se non inizializzato; con `applyBaseSiteThemeFromCms` segue anche `BaseSite.theme` — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L36-L50 — con precedenza: `config.context.theme` statico, poi tema opzionale scelto dall'utente, poi tema CMS — core-libs/core/src/site-theme/services/site-theme-initializer.ts:L93-L127. Il default arriva da `getDefaultClassName` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L46-L63 — e i temi opzionali di default sono i due high-contrast — core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22.

3. `setActive` fa dispatch di `SetActiveSiteTheme` solo se la classe cambia ed è valida — core-libs/core/src/site-theme/facade/site-theme.service.ts:L109-L119; `getActive()` legge `getActiveSiteTheme` dallo store filtrando i null — core-libs/core/src/site-theme/facade/site-theme.service.ts:L99-L104.
4. `LayoutModule` (importato da `BaseStorefrontModule`) importa `ThemeModule` — core-libs/storefront/layout/layout.module.ts:L15-L25, core-libs/storefront/base-storefront.module.ts:L40-L47 —, che registra `initTheme` come `APP_BOOTSTRAP_LISTENER` — core-libs/storefront/layout/theme/theme.module.ts:L10-L25.
5. `ThemeService.init` crea un `Renderer2`, memorizza il root component e sottoscrive `getActive()` — core-libs/storefront/layout/theme/theme.service.ts:L35-L46; `setTheme` rimuove la classe precedente e aggiunge la nuova se non vuota — core-libs/storefront/layout/theme/theme.service.ts:L52-L61.
6. L'utente può cambiare tema da `SiteThemeSwitcherComponent`, che scrive la scelta con `setActive` e registra `useFeatureStyles('a11ySiteContextCaretClick')` — core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.component.ts:L44-L60, core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.component.service.ts:L14-L25.

Il risultato è che il root component porta contemporaneamente la classe del tema (es. `santorini` o `cx-theme-high-contrast-dark`) e le classi `cxFeat_*`. È per questo che `root.scss` deve usare `!important` per battere `.santorini`, caricata dopo con la stessa specificità, e le guardie `:not(.cx-theme-high-contrast-*)` per non toccare i temi high-contrast — core-libs/styles/scss/root.scss:L47-L64.

3.14 Direzione (RTL)

`DirectionMode` ha solo `LTR = 'ltr'` e `RTL = 'rtl'`; il valore HTML5 `auto` non è supportato perché "too fragile" — core-libs/storefront/layout/direction/config/direction.model.ts:L7-L23. La config `direction` (`default`, `detect`, `rtlLanguages`, `ltrLanguages`, tutte opzionali) — core-libs/storefront/layout/direction/config/direction.model.ts:L29-L56 — è esposta da `DirectionConfig`, classe astratta `useExisting: Config` con module augmentation — core-libs/storefront/layout/direction/config/direction.config.ts:L25-L35. Il default è `detect: true`, `default: LTR`, `rtlLanguages: ['he', 'ar']` — core-libs/storefront/layout/direction/config/default-direction.config.ts:L10-L19.

Flusso:

1. `LayoutModule` importa `DirectionModule` — core-libs/storefront/layout/layout.module.ts:L15-L25 —, che registra un `APP_INITIALIZER` (`initHtmlDirAttribute`) che restituisce la Promise di `DirectionService.initialize()` — core-libs/storefront/layout/direction/direction.module.ts:L12-L35. `APP_INITIALIZER` è deprecato in Angular dalla v19 a favore di `provideAppInitializer` — angular:packages/core/src/application/application_init.ts:L36-L42.
2. `initialize()` converte con `lastValueFrom` la config stabile `direction`: se `detect` chiama `detect()`, altrimenti imposta `config.default` su `document.documentElement` — core-libs/storefront/layout/direction/direction.service.ts:L46-L62.
3. `detect()` (protetto da `startsDetecting`) sottoscrive `LanguageService.getActive()` e a ogni lingua imposta `dir` — core-libs/storefront/layout/direction/direction.service.ts:L68-L83.
4. `getDirection` controlla `rtlLanguages`, poi `ltrLanguages`, poi `default` — core-libs/storefront/layout/direction/direction.service.ts:L104-L112; `setDirection` imposta l'attributo o lo rimuove se `undefined` — core-libs/storefront/layout/direction/direction.service.ts:L89-L95.
5. `ngOnDestroy` disiscrive "to avoid memory leaks in SSR" — core-libs/storefront/layout/direction/direction.service.ts:L114-L117.

Il CSS reagisce soprattutto con proprietà logiche (216 righe con `*-inline-start`/`*-inline-end` e nessuna proprietà fisica `margin|padding-left|right`: le sole occorrenze sono il nome della custom property `--cx-margin-right`) — `core-libs/styles/scss/` (grep), core-libs/styles/scss/components/content/video/_video.

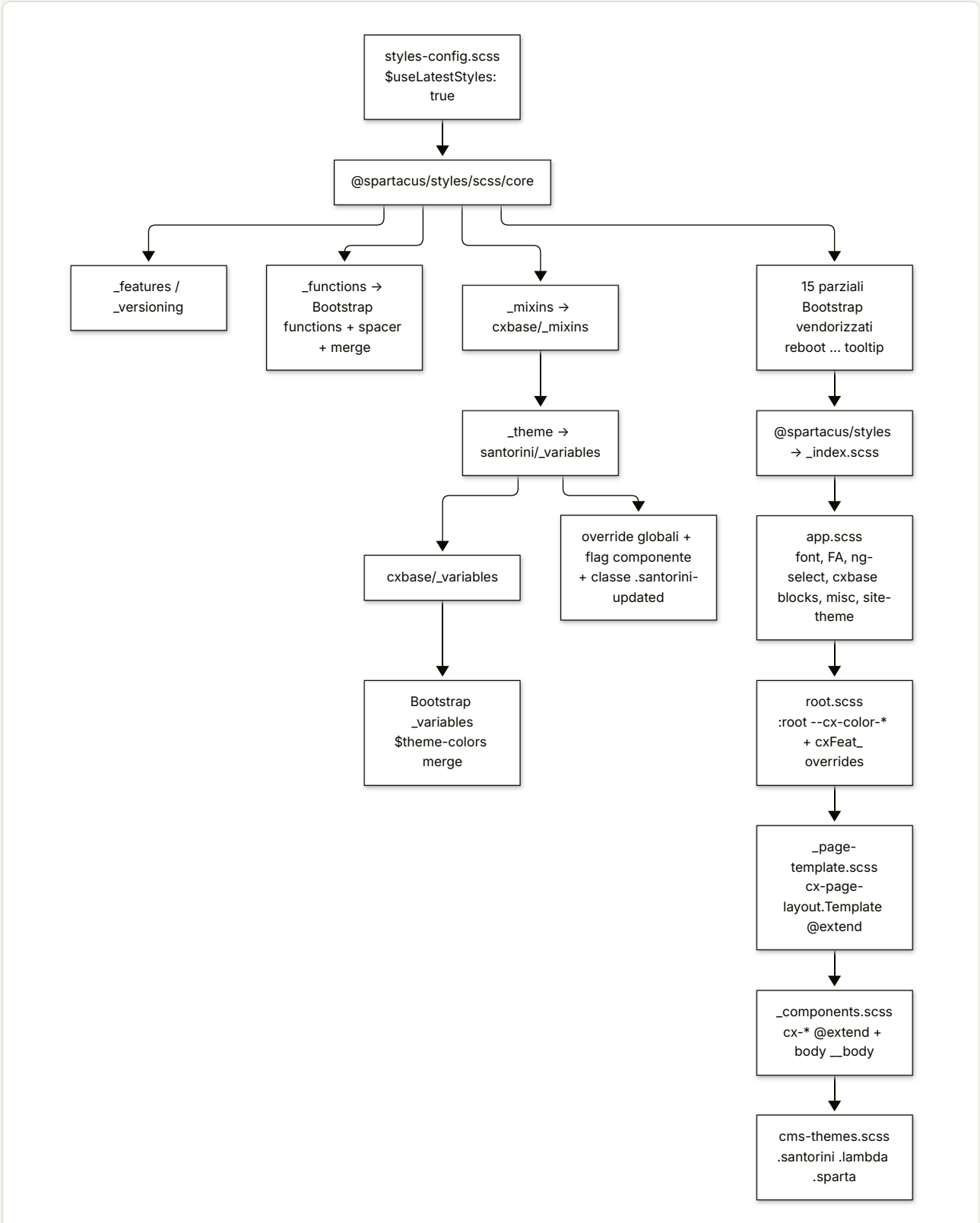
scss:L19-L19, core-libs/styles/scss/cxbase/blocks/alert.scss:L26-L26, core-libs/styles/scss/cxbase/blocks/message.scss:L61-L61 — e, dove serve, con selettori `[dir='rtl']`: icone con `flip-at-rtl`/`flip-at-ltr`, separatore del breadcrumb, verso delle stelle di rating — core-libs/styles/scss/components/misc/_icons.scss:L1-L7, core-libs/styles/scss/components/content/navigation/_breadcrumb.scss:L1-L3, core-libs/styles/scss/components/product/_star-rating.scss:L35-L38. `IconComponent` espone le due classi come `@HostBinding` "for the style layer" — core-libs/storefront/cms-components/misc/icon/icon.component.ts:L69-L77. La lingua attiva è descritta in 13 — Site context e site theme e in 37 — Internazionalizzazione (i18n).

3.15 Breakpoint: Sass vs TypeScript

I breakpoint Sass di Bootstrap sono valori **minimi**: xs 0, sm 576px, md 768px, lg 992px, xl 1200px — core-libs/styles/vendor/bootstrap/scss/_variables.scss:L207-L213. Il `defaultLayoutConfig` TypeScript usa numeri xs 576, sm 768, md 992, lg 1200, xl `{min: 1200}`, e `BreakpointService` tratta i numeri come larghezza **massima** — core-libs/storefront/layout/config/default-layout.config.ts:L9-L19, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L161-L176. Le due scale coincidono come confini ma sono sfalsate di un nome (vedi §9 e 18 — Layout, breakpoint, launch dialog e accessibilità).

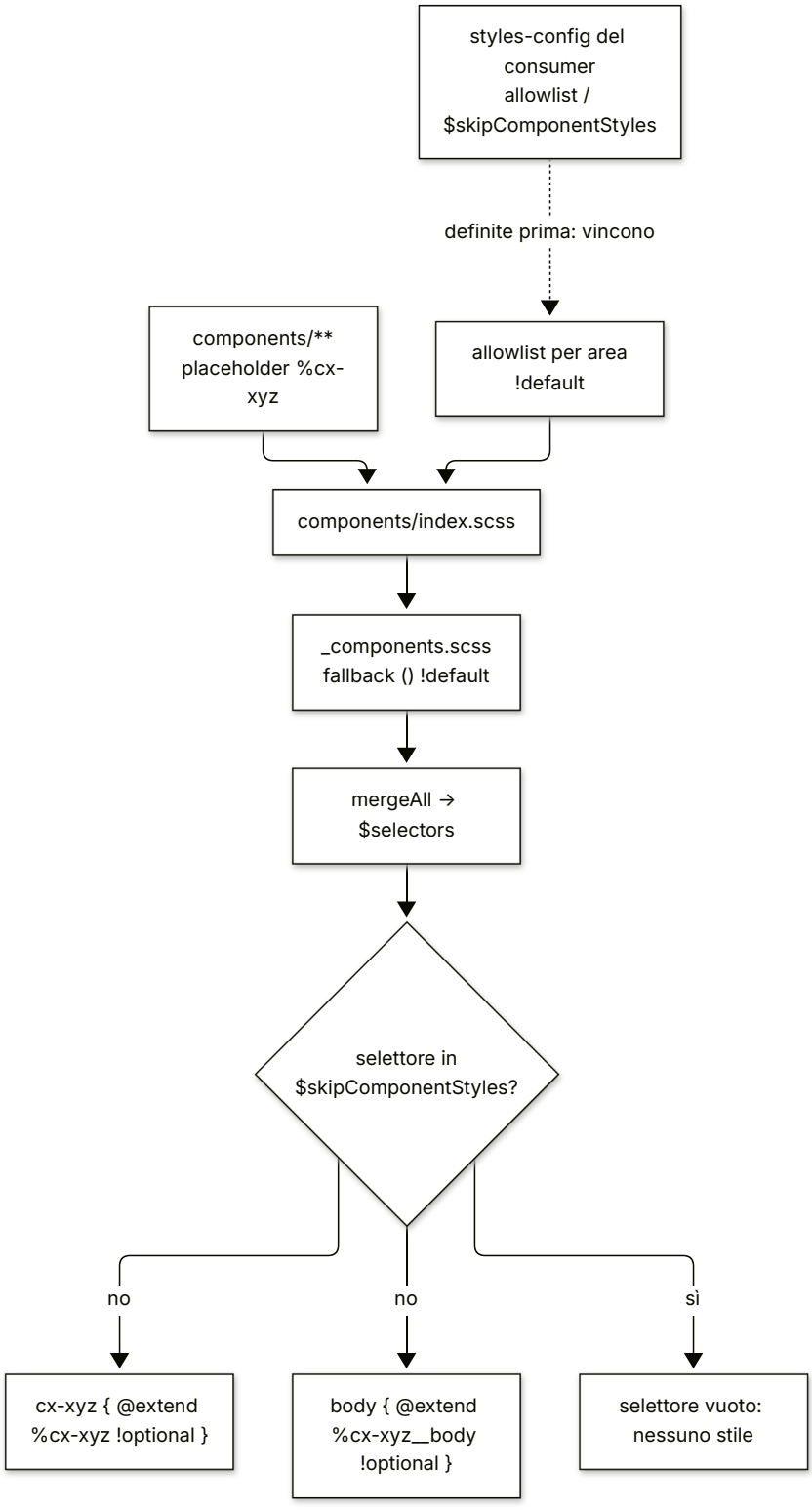
4. Diagrammi

4.1 Ordine di compilazione del bundle principale (app demo)



Riferimenti: projects/storefrontapp/src/styles.scss:L1-L29, core-libs/styles/scss/_core.scss:L1-L5, core-libs/styles/scss/cxbase/_mixins.scss:L1-L15, core-libs/styles/scss/theme/santorini/_variables.scss:L114-L215, core-libs/styles/_index.scss:L1-L4.

4.2 Dalla allowlist al CSS emesso



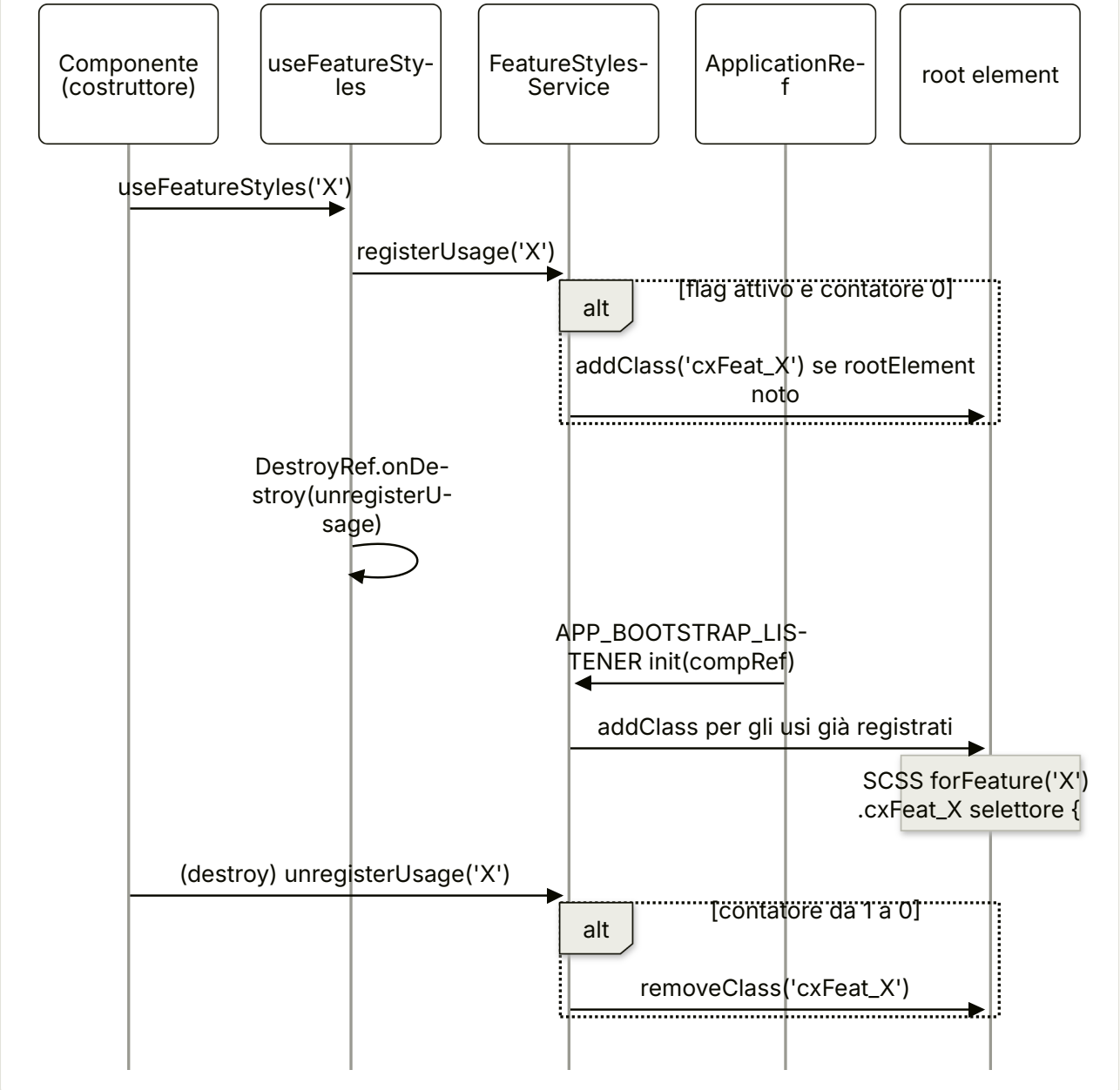
Riferimenti: core-libs/styles/scss/_components.scss:L1-L53, core-libs/styles/scss/cxbase/functions/merge.scss:L1-L7.

4.3 Site theme: dallo store alla classe sul root component

■ TAVOLA 38.1 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/site-theme/services/site-theme-initializer.ts:L36-L127, core-libs/core/src/site-theme/facade/site-theme.service.ts:L99-L119, angular:packages/core/src/application/application_ref.ts:L775-L795, core-libs/storefront/layout/theme/theme.service.ts:L35-L61.

4.4 Stili condizionati da feature flag



Riferimenti: core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40, core-libs/core/src/features-config/services/feature-styles.service.ts:L65-L136, core-libs/core/src/features-config/features-config.module.ts:L41-L49, core-libs/styles/scss/_features.scss:L41-L62.

4.5 Direzione al bootstrap e al cambio lingua

■ **TAVOLA 38.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/layout/direction/direction.module.ts:L12-L35, core-libs/storefront/layout/direction/direction.service.ts:L46-L112.

---

5. Estratti di codice reale

5.1 Emissione dei componenti

core-libs/styles/scss/_components.scss:L1-L19 (estratto)

```scss
@import './functions';
@import './components/index';

// the $skipComponentStyles is used as a list of component styles that can be ignore
// during the sass compilation. Whenever the list contains a component selector, the
// selector is not used.
$skipComponentStyles: () !default;

$layout-components-allowlist: () !default;
$product-components-allowlist: () !default;
// ...
$cpq-quote-allowlist: () !default;
```

core-libs/styles/scss/_components.scss:L37-L53

```scss
@each $selector in $selectors {
  #{$selector} {
    // skip selectors if they're added to the $skipComponentStyles list
    @if (index($skipComponentStyles, $selector) ==null) {
      @extend %#{$selector} !optional;
    }
  }
}

// add body specific selectors
body {
  @each $selector in $selectors {
    @if (index($skipComponentStyles, $selector) ==null) {
      @extend %#{$selector}__body !optional;
```

```
    }
  }
}
```

5.2 Page template

core-libs/styles/scss/_page-template.scss:L7-L27

```scss
// The $page-template-blocklist can be used to prevent template CSS
// being generated.
$page-template-blocklist: () !default;

// allowlisted page template selectors are processed
// unless (some) that are blocklisted using `$page-template-blocklist`
$page-template-allowlist:
  LandingPage2Template, ContentPage1Template, CategoryPageTemplate,
  ProductListPageTemplate, ProductGridPageTemplate,
  SearchResultsListPageTemplate, ProductDetailsPageTemplate, LoginPageTemplate,
  MultiStepCheckoutSummaryPageTemplate, ErrorPageTemplate,
  CheckoutLoginPageTemplate, AccountPageTemplate, OrderConfirmationPageTemplate,
  MyAccountViewPageTemplate !default;

@each $selector in $page-template-allowlist {
  cx-page-layout.#{$selector} {
    @if (index($page-template-blocklist, $selector) == null) {
      @extend %#{$selector} !optional;
    }
  }
}
```

5.3 `:root` e gli override a11y

core-libs/styles/scss/root.scss:L1-L16 (estratto)

```scss
@import 'theme';

:root {
  @each $color, $value in $theme-colors {
    --cx-color-#{$color}: #{$value};
  }
  --cx-page-width-max: 1140px;

  --cx-font-weight-light: 300;
  // ...
  --cx-direction: #{$direction};
```

core-libs/styles/scss/root.scss:L55-L70

```scss
// `!important` is required because CMS theme classes (e.g. `.santorini`) also
// set this property on the same element with equal specificity and are loaded
// afterwards, so they would otherwise win. The `:not()` guards exclude the
// high-contrast themes, which define their own compliant focus colors that
// must not be overridden.
.#{getFeatureCssClass('a11yFocusIndicatorContrast')}:not(
    .cx-theme-high-contrast-dark
  ):not(.cx-theme-high-contrast-light) {
  --cx-color-visual-focus: #1f3a93 !important;
}

.#{getFeatureCssClass('a11yPrimaryColorContrast')}:not(
    .cx-theme-high-contrast-dark
  ):not(.cx-theme-high-contrast-light) {
  --cx-color-primary: #055f9f !important;
}
```

5.4 Il mixin `forFeature`

core-libs/styles/scss/_features.scss:L41-L62

```scss
@mixin forFeature($feature, $classTarget: 'ancestorSelector') {
  // Validate params
  @if (type-of($feature) != 'string') {
    @error "forFeature: $feature param must be a string, but #{type-of($feature)} was provided.";
  }
  @if (not index(('ancestorSelector', 'currentSelector'), $classTarget)) {
    @error "forFeature: $classTarget must be either 'ancestorSelector' or 'currentSelector', but #
{$classTarget} was provided.";
  }
  // Validate params end

  $featureClass: getFeatureCssClass($feature);
  @if ($classTarget == 'ancestorSelector') {
    @at-root .#{$featureClass} & {
      @content;
    }
  }
  @if ($classTarget == 'currentSelector') {
    @at-root &.#{$featureClass} {
      @content;
    }
  }
}
```

5.5 Il contatore di `FeatureStylesService`

core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L115

```ts
registerUsage(feature: FeatureToggleKey): void {
  const currentCounter = this.usagesCounter.get(feature) ?? 0;
  if (this.isEnabled(feature) && currentCounter === 0) {
    this.addClass(feature);
  }
  this.usagesCounter.set(feature, currentCounter + 1);
}

/**
 * Unregisters the usage of a feature flag.
 *
 * If the feature flag is enabled, this method removes the feature's CSS class
 * from the root element (unless there are other registered usages of the same feature yet).
 */
unregisterUsage(feature: FeatureToggleKey): void {
  const currentCounter = this.usagesCounter.get(feature) ?? 0;
  if (currentCounter === 0 && isDevMode()) {
    this.logger.warn(
      `Feature flag CSS: "${feature}" is already not used, so it cannot be unregistered.`
    );
  }

  if (this.isEnabled(feature) && currentCounter === 1) {
    this.removeClass(feature);
  }
  this.usagesCounter.set(feature, Math.max(currentCounter - 1, 0));
}
```

5.6 ThemeService

core-libs/storefront/layout/theme/theme.service.ts:L35-L61

```ts
init(rootComponent: ComponentRef<any>): void {
  this.renderer = this.rendererFactory.createRenderer(null, null);
  this.rootComponent = rootComponent;
  // Theme value is a string. It is put in the generic multi-value
  // property of the SiteContextConfig. So the array's first item
  // is the theme value.
  this.subscription.add(
    this.siteThemeService
      .getActive()
      .subscribe((activeTheme) => this.setTheme(activeTheme))
  );
}

ngOnDestroy(): void {
  this.subscription.unsubscribe();
}
```

```
  setTheme(theme: string | undefined): void {
    const element = this.rootComponent.location.nativeElement;
    if (this.existingTheme) {
      this.renderer.removeClass(element, this.existingTheme);
    }
    if (theme) {
      this.renderer.addClass(element, theme);
      this.existingTheme = theme;
    }
  }
```

5.7 DirectionService

core-libs/storefront/layout/direction/direction.service.ts:L46-L62

```ts
  initialize(): Promise<any> {
    return lastValueFrom(
      this.configInit.getStable('direction').pipe(
        tap((config: DirectionConfig) => {
          this.config = config?.direction;
          if (this.config?.detect) {
            this.detect();
          } else {
            this.setDirection(
              this.winRef.document.documentElement,
              this.config?.default
            );
          }
        })
      )
    );
  }
```

core-libs/storefront/layout/direction/direction.service.ts:L104-L112

```ts
  getDirection(language?: string): DirectionMode | undefined {
    if (language && this.config?.rtlLanguages?.includes(language)) {
      return DirectionMode.RTL;
    }
    if (language && this.config?.ltrLanguages?.includes(language)) {
      return DirectionMode.LTR;
    }
    return this.config?.default;
  }
```

5.8 Integrazione nell'app demo

projects/storefrontapp/src/styles.scss:L1-L29 (estratto)

```scss
@import 'styles-config';

// ORDER IMPORTANT: Spartacus core first
@import '@spartacus/styles/scss/core';

// ORDER IMPORTANT: Bootstrap next

@import '@spartacus/styles/vendor/bootstrap/scss/reboot';
// ...
@import '@spartacus/styles/vendor/bootstrap/scss/tooltip';

// ORDER IMPORTANT: Spartacus styles last
@import '@spartacus/styles';

// CMS-driven base-site theme rules (must come after Spartacus styles so
// the CSS custom properties override the defaults in `:root`).
@import 'cms-themes';
```

5.9 Il tema compilato e i suoi flag

core-libs/styles/scss/_theme.scss:L5-L12

```scss
//point to the theme variables
//@import 'theme/sparta/variables';
//@import "theme/lambda/variables";

//old theme to be commented out on next major or with impex
@import 'theme/santorini/variables';
// new theme below to uncomment on next major or with impex
// @import 'theme/santorini-updated/variables';
```

core-libs/styles/scss/components/content/tab-paragraph-container/_tab-paragraph-container.scss:L221-L227

```scss
%cx-tab-paragraph-container {
  @if $useTabsOnPDP == true {
    @extend %pdpTabs !optional;
  } @else {
    @extend %pdpFlat !optional;
  }
  @extend %cx-tab !optional;
```

5.10 Ridefinizione di un mixin in una feature lib

feature-libs/cart/_index.scss:L1-L11

```scss
@import '@spartacus/styles/scss/core';
@import '@spartacus/styles/scss/cxbase/blocks/modal';
@import '@spartacus/styles/scss/misc/table';

@mixin visible-focus {
  outline-style: solid;
  outline-color: var(--cx-color-visual-focus);
  outline-width: var(--cx-visual-focus-width, 2px);
  outline-offset: 4px;
  transition: none;
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
direction.detect	boolean	true	core-libs/storefront/layout/direction/config/default-direction.config.ts:L12-L12	core-libs/storefront/layout/direction/direction.service.ts:L51-L51
direction.default	DirectionMode	LTR	core-libs/storefront/layout/direction/config/default-direction.config.ts:L13-L13	core-libs/storefront/layout/direction/direction.service.ts:L54-L57, core-libs/storefront/layout/direction/direction.service.ts:L111-L111
direction.rtlLanguages	string[]	['he','ar']	core-libs/storefront/layout/direction/config/default-direction.config.ts:L17-L17	core-libs/storefront/layout/direction/direction.service.ts:L105-L105
direction.ltrLanguages	string[]	—	core-libs/storefront/layout/direction/config/direction.model.ts:L52-L55	core-libs/storefront/layout/direction/direction.service.ts:L108-L108
siteTheme.optionalThemes	SiteTheme[]	high-contrast dark/light	core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22	core-libs/core/src/site-theme/facade/site-theme.service.ts:L86-L89
context.theme	string[]	—	core-libs/core/src/site-context/providers/context-ids.ts:L10-L10 (THEME_CONTEXT_ID)	core-libs/core/src/site-theme/facade/site-theme.service.ts:L46-L63
feature applyBaseSiteThemeFromCms	boolean	false (demo: true)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1092-L1092	core-libs/core/src/site-theme/services/site-theme-initializer.ts:L36-L50
feature a11yFocusIndicatorContrast / a11yPrimaryColorContrast / a11yDisabledButtonContrast	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1101-L1103	core-libs/styles/scss/root.scss:L60-L70, core-libs/styles/scss/cxbase/blocks/buttons.scss:L77-L84
breakpoints	per screen	xs 576, sm 768, md 992, lg 1200, xl {min:1200}	core-libs/storefront/layout/config/default-layout.config.ts:L9-L19	core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L161-L176

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
Sass `$styleVersion`	number	`$_majorVersion` (2211)	`core-libs/styles/scss/_versioning.scss:L41-L41`	`core-libs/styles/scss/_versioning.scss:L213-L231`
Sass `$useLatestStyles`	bool	`false` (demo: `true`)	`core-libs/styles/scss/_versioning.scss:L46-L46`; `projects/storefrontapp/src/styles-config.scss:L2-L2`	`core-libs/styles/scss/_versioning.scss:L213-L231`
Sass `$skipComponentStyles`	list	`()`	`core-libs/styles/scss/_components.scss:L7-L7`	`core-libs/styles/scss/_components.scss:L40-L49`
Sass `$<area>-components-allowlist` (11)	list	per area	`core-libs/styles/scss/components/content/_index.scss:L10-L14` (e analoghi); fallback `core-libs/styles/scss/_components.scss:L9-L19`	`core-libs/styles/scss/_components.scss:L21-L53`
Sass `$page-template-allowlist` / `$page-template-blocklist`	list	14 template / `()`	`core-libs/styles/scss/_page-template.scss:L9-L19`	`core-libs/styles/scss/_page-template.scss:L21-L27`
Sass `$theme-colors`	map	santorini (24 chiavi)	`core-libs/styles/scss/theme/santorini/_variables.scss:L60-L85`	`core-libs/styles/scss/root.scss:L4-L6`
Sass `$type`	map	scala 1–8	`core-libs/styles/scss/theme/santorini/_variables.scss:L102-L112`	`core-libs/styles/scss/cxbase/mixins/_type.scss:L3-L29`
Sass `$direction`	`ltr`/`rtl`	`ltr`	`core-libs/styles/scss/cxbase/_variables.scss:L32-L32`	`core-libs/styles/scss/root.scss:L16-L16`
Sass `$useTabsOnPDP`, `$useAccordionOnly`	bool	`true` (tema) / `false` (componente)	`core-libs/styles/scss/theme/santorini/_variables.scss:L173-L174`	`core-libs/styles/scss/components/content/tab-paragraph-container/_tab-paragraph-container.scss:L140-L140`, `core-libs/styles/scss/components/content/tab-paragraph-container/_tab-paragraph-container.scss:L222-L222`
Sass `$spinWhileLoading`, `$spinWhileWaiting`, `$zoomThumbOnHover`, `$overlayWhileLoading`	bool	`true` (tema)	`core-libs/styles/scss/theme/santorini/_variables.scss:L169-L172`	`core-libs/styles/scss/components/product/details/_product-images.scss:L44-L64`
Sass `$cx-hamburger-type`	string	`squeeze`	`core-libs/styles/scss/misc/_hamburger.scss:L6-L6`	`core-libs/styles/scss/misc/_hamburger.scss:L8-L19`
Sass `$font-url`	string	Google Fonts Open Sans	`core-libs/styles/scss/theme/santorini/_variables.scss:L28-L28`	`core-libs/styles/scss/theme/santorini/_fonts.scss:L1-L1`
CSS `--cx-color-*`, `--cx-font-weight-*`, `--cx-page-width-max`, `--cx-buttons-border-radius`, …	custom property	da `$theme-colors` / costanti	`core-libs/styles/scss/root.scss:L3-L45`	componenti via `var(...)`
CSS `--cx-visual-focus-width`	length	`2px` (fallback)	`core-libs/styles/scss/cxbase/mixins/_visible-focus.scss:L4-L4`	`core-libs/styles/scss/components/layout/_storefront.scss:L36-L42`
CSS `--cx-flex-basis`	length	`100%` (fallback)	`core-libs/styles/scss/layout/page-templates/_common-template.scss:L12-L20`	idem

7. Commenti e note notevoli dal sorgente

- "🚨 CAUTION. The approach of using `forVersion($level)` in this file is deprecated and superseded by the recommendation of using named feature flags with mixin `forFeature($featureName)`." — `core-libs/styles/scss/_versioning.scss:L1-L3`; e `@deprecated since 2211.20, use forFeature($feature) mixin instead` — `core-libs/styles/scss/_versioning.scss:L63-L63`. Il versioning numerico è ancora nel pacchetto (e nei test) ma non va usato per nuovi stili.
- "IMPORTANT: Please mind to call `useFeatureStyles(feature)` from `@spartacus/core` in the constructor of your styled component. Otherwise the feature-flagged styles MIGHT NOT be activated EVEN if the flag is enabled" — `core-libs/styles/scss/_features.scss:L16-L31`. È il punto di contatto obbligatorio fra SCSS e TypeScript.
- "The second argument with value `'currentSelector'` should be passed ONLY in one edge case, when the feature-flagged styles are defined for the Angular root element, i.e. `app-root`." — `core-libs/styles/scss/_features.scss:L38-L40`.
- "TODO: Required to use feature flag with root styles. Remove this entire block once the a11yPreventWindowsHighContrastOverride feature flag is removed" — `core-libs/storefront/layout/main/storefront.component.ts:L141-L142`.
- Commento sul `!important` di `--cx-color-visual-focus`: spiega WCAG 1.4.11 (≥3:1) e la competizione con `.santorini` caricata dopo — `core-libs/styles/scss/root.scss:L47-L59`.
- "This is NOT a theme. This file should include ONLY new variables that Bootstrap does not provide." — `core-libs/styles/scss/cxbase/_variables.scss:L1-L13`.
- "You can: 1) Set new values for Bootstrap variables ... You cannot: 1) Add new variables" — `core-libs/styles/scss/theme/santorini/_variables.scss:L1-L9`.
- "old theme to be commented out on next major or with impex" / "new theme below to uncomment on next major or with impex" — `core-libs/styles/scss/_theme.scss:L9-L12`; e "santorini-updated variables for impex file, remove on next major" — `core-libs/styles/scss/theme/santorini/_variables.scss:L176-L176`. Il passaggio a santorini-updated è pianificato ma non ancora fatto.
- "TODO: Replace `$font-url` with swap url in 5.0 to use swap strategy by default." — `core-libs/styles/scss/theme/santorini/_variables.scss:L26-L26`: TODO storico mai chiuso.
- `// TODO (#11607) remove LoginPageTemplate from allowlist` — `core-libs/styles/scss/_page-template.scss:L5-L5`.
- "this change is not kicking-in, minor issue, it does work work when unwraping, unwrap in next major" — `core-libs/styles/scss/components/layout/_storefront.scss:L37-L37`, accanto alla regola `.mouse-focus`.
- "Font Awesome breaking change on v7: Fixed Width is now default for icons." — `core-libs/styles/scss/root.scss:L27-L29`.
- "The HTML5 "auto" value is not supported in Spartacus, as it's considered to be too fragile" — `core-libs/storefront/layout/direction/config/direction.model.ts:L11-L12`.
- "we don't export the factory to avoid a breaking change when we drop the initHtmlDirAttribute" — `core-libs/storefront/layout/direction/index.ts:L9-L9`.
- "Cleans up the subscription, to avoid memory leaks in SSR." — `core-libs/storefront/layout/direction/direction.service.ts:L115-L115`.
- "Note: The default Site Theme can be driven by the CMS, but additional possible selectable themes (like a11y high contrast themes) can be configured only in the separate Spartacus configuration `config.siteTheme.optionalThemes`." — `core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L90-L93`.

8. Extension point

EXTENSION POINT	LIVELLO	COME SI USA	RIFERIMENTO
Variabili tema (`$primary`, `$theme-colors`, `$type`, `$font-url`, flag di componente)	Sass, compile time	Assegnarle in `styles-config.scss` (prima di `scss/core`) per le `!default`; per le mappe senza `!default` vedi §9	`core-libs/styles/scss/theme/santorini/_variables.scss:L28-L174`
`$skipComponentStyles`	Sass	Lista di selettori `cx-*` per cui non emettere lo stile di libreria (per riscriverlo da zero)	`core-libs/styles/scss/_components.scss:L4-L7`
`$<area>-components-allowlist`	Sass	Ridefinire la lista prima dell'import per restringere o ampliare i selettori emessi	`core-libs/styles/scss/components/content/_index.scss:L10-L14`, `core-libs/styles/scss/_components.scss:L9-L19`
`$page-template-allowlist` / `$page-template-blocklist`	Sass	Escludere o aggiungere page template; un template nuovo richiede anche il suo placeholder `%<Template>`	`core-libs/styles/scss/_page-template.scss:L7-L27`
Placeholder `%cx-*`, `%cx-*__body`, `%<Template>`	Sass	Aggiungere regole al placeholder dopo l'import di `scss/core`, oppure creare il placeholder di un componente custom e aggiungerne il selettore alla allowlist	`core-libs/styles/scss/_components.scss:L37-L53`
Custom property `--cx-*`	CSS runtime	Ridefinirle su `:root` o su una classe tema (pattern di `cms-themes.scss`)	`core-libs/styles/scss/root.scss:L3-L45`, `projects/storefrontapp/src/cms-themes.scss:L1-L60`
Classi tema (`.santorini`, `.cx-theme-high-contrast-*`, temi custom)	CSS + config	Aggiungere un tema a `siteTheme.optionalThemes` e la classe CSS corrispondente; oppure `context.theme` / `BaseSite.theme`	`core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22`
Mixin `forFeature`, `cx-highContrastTheme*`, `visible-focus`, `type`, `weight`	Sass	Riusarli negli stili custom; ridefinirli dopo `scss/core` (come fa cart con `visible-focus`)	`core-libs/styles/scss/_features.scss:L41-L62`, `core-libs/styles/scss/theme/high-contrast/mixins/_mixins.scss:L1-L40`, `feature-libs/cart/_index.scss:L5-L11`
`useFeatureStyles`, `FeatureStylesService`	TS (`@spartacus/core`)	Chiamare `useFeatureStyles` nel costruttore dei componenti custom con stili `forFeature`	`core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40`
`ThemeService` (`init`, `setTheme`), `ThemeModule`	TS (`@spartacus/storefront`)	Sostituire il servizio via DI per mettere la classe su un altro elemento	`core-libs/storefront/layout/theme/theme.service.ts:L18-L62`
`DirectionConfig`, `DirectionService` (`setDirection`, `getDirection`, `detect`)	TS (`@spartacus/storefront`)	`provideConfig({ direction: { … } })` o sostituzione del servizio (es. match su prefisso lingua)	`core-libs/storefront/layout/direction/config/direction.config.ts:L25-L35`, `core-libs/storefront/layout/direction/direction.service.ts:L28-L118`
`$styleVersion`, `$useLatestStyles`	Sass (deprecato)	Solo per stili legacy con `forVersion`	`core-libs/styles/scss/_versioning.scss:L39-L46`

Ricette pratiche (Parte III):

- Override SCSS, placeholder, allowlist/skip e temi custom: 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni.
- Configurazione `direction` e `siteTheme` con `provideConfig`, sostituzione di servizi come `ThemeService`/`DirectionService` via DI: 50 — Ricette: configurazione, DI, interceptor, eventi.
- Stili di un componente CMS custom e di nuovi page template/layout: 51 — Ricette: componenti CMS, outlet, route e layout.
- Cosa è API pubblica (custom property, variabili `!default`) e cosa è dettaglio interno (struttura dei file SCSS) in vista degli upgrade: 54 — API pubblica vs interna, upgrade, anti-pattern, checklist.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Le allowlist vanno ridefinite PRIMA dell'import.** In `_components.scss` le dichiarazioni `() !default` vengono dopo `@import './components/index'`, quindi sono solo fallback; e le allowlist delle aree sono `!default`. Per cambiarne una bisogna assegnarla prima di importare `@spartacus/styles` (es. in `styles-config`) — `core-libs/styles/scss/_components.scss:L1-L19`, `core-libs/styles/scss/components/content/_index.scss:L10-L14`.
2. `$skipComponentStyles` **non elimina tutto.** Filtra solo l'`@extend` sull'elemento: classi che estendono direttamente il placeholder (es. `.cx-icon`) restano emesse — `core-libs/styles/scss/_components.scss:L37-L44`, `core-libs/styles/scss/components/misc/_icons.scss:L9-L11`.
3. **Template senza CSS.** `%SearchResultsGridPageTemplate` esiste ma `SearchResultsGridPageTemplate` non è in `$page-template-allowlist`, quindi il suo CSS non viene generato di default, pur esistendo il template nel layout TS — `core-libs/styles/scss/layout/page-templates/_search-results-grid.scss:L1-L1`, `core-libs/styles/scss/_page-template.scss:L13-L19`, `core-libs/storefront/recipes/config/layout-config.ts:L88-L88`.
4. **Selettori in allowlist senza placeholder.** `cx-page-header` è in `$content-components-allowlist` ma esiste solo la classe `.cx-page-header`; `!optional` evita l'errore — `core-libs/styles/scss/components/content/_index.scss:L13-L13`, `core-libs/styles/scss/cxbase/layout/page.scss:L7-L11`. Analogamente `MultiStepCheckoutSummaryPageTemplate` è nell'allowlist core ma il placeholder sta nel bundle della lib checkout — `core-libs/styles/scss/_page-template.scss:L17-L17`, `feature-libs/checkout/base/styles/layout/_index.scss:L9-L12`.
5. **Bundle indipendenti.** Ogni bundle di feature lib è una compilazione separata che riparte da `scss/core` e, in 21 bundle su 24, da `styles-config` (non in `lib-customer-ticketing`, `lib-s4om`, `lib-subscription-billing`): una variabile impostata solo nel foglio principale non arriva nei bundle delle lib — `projects/storefrontapp/src/styles/lib-cart.scss:L1-L2`, `projects/storefrontapp/src/styles/lib-customer-ticketing.scss:L1-L1`, `feature-libs/cart/_index.scss:L1-L3`. Inoltre le lib possono ridefinire mixin core: cart ridefinisce `visible-focus` con `outline-offset: 4px`, quindi il focus nel bundle cart differisce da quello core — `feature-libs/cart/_index.scss:L5-L11`, `core-libs/styles/scss/cxbase/mixins/_visible-focus.scss:L1-L7`.
6. **Mappe senza** `!default`**.** `$theme-colors` e `$type` nei temi sono assegnate senza `!default`; poiché `_theme.scss` viene reimportato da molti file (circa 20 file sotto `scss/` importano `../../theme` o simili), un valore del consumer per queste mappe impostato in `styles-config` verrebbe sovrascritto dal reimport. DA VERIFICARE con una compilazione reale (con `@import` Sass non deduplica i file) — `core-libs/styles/scss/theme/santorini/_variables.scss:L60-L112`, `core-libs/styles/scss/cxbase/layout/app.scss:L1-L1`. Per lo stesso motivo la classe `.santorini-updated` potrebbe essere emessa più volte — `core-libs/styles/scss/theme/santorini/_variables.scss:L177-L215`.

7. **Entry** `theme-*.scss` **probabilmente inefficaci.** Caricano le variabili del tema e poi `./index`, che reimporta `_theme` → santorini: le `!default` del primo tema restano, ma le assegnazioni senza `!default` di santorini (`$theme-colors`, `$type`, …) le sovrascrivono; per Lambda la mappa `brand`/`accent` andrebbe persa e, anzi, il file di variabili importa `cxbase/mixins` → `../theme` → santorini **prima** dei propri colori, quindi le `!default` di Lambda (`$primary: #987654 !default`, …) non prevalgono neppure al primo import — core-libs/styles/scss/theme/lambda/_variables.scss:L12-L19, core-libs/styles/scss/cxbase/_mixins.scss:L5-L5, core-libs/styles/scss/_theme.scss:L10-L10 —; per Sparta mancherebbero comunque `disabled`/`disabled-accent` nella sua mappa. DA VERIFICARE con compilazione — core-libs/styles/theme-lambda.scss:L1-L3, core-libs/styles/scss/root.scss:L1-L1, core-libs/styles/scss/theme/sparta/_variables.scss:L50-L73. Nessun file del repo referenzia questi entry.
8. **Ordine font/variabili in** `theme-santorini.scss`**.** Il file dei font (che usa `$font-url`) è importato prima delle variabili che lo definiscono. DA VERIFICARE: senza compilazione non è confermato se Sass segnali variabile non definita quando il consumer non la imposta prima — core-libs/styles/theme-santorini.scss:L1-L2, core-libs/styles/scss/theme/santorini/_fonts.scss:L1-L1, core-libs/styles/scss/theme/santorini/_variables.scss:L28-L28.
9. `--theme` **dello schematic.** Con `--theme=<x>` lo schematic aggiunge `@import '@spartacus/styles/scss/theme/<x>'`, ma `scss/theme/santorini/` contiene solo `_fonts.scss` e `_variables.scss`, nessun `_index.scss`. DA VERIFICARE se l'import si risolve — core-libs/schematics/src/add-spartacus/index.ts:L170-L172.
10. `$styleVersion` **dello schematic.** Il default generato è nel formato `221121.22`, mentre `$_majorVersion` è `2211`: `compareVersion` lo considera maggiore, quindi `useBreakingChanges()` risulta vero. Effetto pratico nullo nel core, dove `forVersion` non è più usato. DA VERIFICARE con compilazione — core-libs/schematics/src/add-spartacus/index.ts:L74-L76, core-libs/styles/scss/_versioning.scss:L228-L231.
11. `forFeature` **senza** `useFeatureStyles` **non fa nulla.** Anche con il flag attivo, la classe `cxFeat_<flag>` compare solo se un componente vivo ha registrato l'uso; e per i selettori su `:root` serve la gestione manuale come in `StorefrontComponent` — core-libs/styles/scss/_features.scss:L18-L21, core-libs/storefront/layout/main/storefront.component.ts:L141-L147. `unregisterUsage` con contatore già a 0 in dev mode produce solo un warning — core-libs/core/src/features-config/services/feature-styles.service.ts:L103-L109.
12. **La classe tema sta su** `app-root`**, non su** `cx-storefront`**.** La regola `%cx-storefront { &.cx-theme-high-contrast-* { … } }` scatta solo se la classe è su `cx-storefront` stesso, mentre `ThemeService` la mette sul root component. DA VERIFICARE: non trovato altro `addClass` della classe tema in `core-libs/storefront`; potrebbe essere efficace solo nelle app che bootstrappano direttamente `cx-storefront` — core-libs/styles/scss/components/layout/_storefront.scss:L52-L59, core-libs/storefront/layout/theme/theme.service.ts:L52-L61.
13. **Dettagli di** `ThemeService`**.** `setTheme('')`/`undefined` rimuove la classe precedente ma non azzera `existingTheme` (alla chiamata successiva prova di nuovo a rimuoverla, innocuo); `setTheme` usa `rootComponent` senza guard, quindi va in errore se chiamato prima di `init`; il costruttore inietta `SiteContextConfig` senza usarlo e il commento in `init` descrive una vecchia logica — core-libs/storefront/layout/theme/theme.service.ts:L25-L61.
14. **Direzione: match esatto e rimozione di** `dir`**.** Il confronto è `includes(isocode)`: `ar_SA` non corrisponde ad `ar` — core-libs/storefront/layout/direction/direction.service.ts:L105-L108. Con lingua non mappata e senza `default`, `getDirection` restituisce `undefined` e l'attributo `dir` viene rimosso — core-libs/storefront/layout/direction/direction.service.ts:L89-L112. Una seconda chiamata a `initialize()` aggiorna `this.config` ma non riavvia la subscription (guardia `startsDetecting`) — core-libs/storefront/layout/direction/direction.service.ts:L46-L83.
15. **Direzione Sass statica.** `$direction` è `ltr` a compile time e indipendente da `DirectionService`; anche `--cx-direction` non è letta da nessuna regola — core-libs/styles/scss/cxbase/_variables.scss:L32-L32,

core-libs/styles/scss/root.scss:L16-L16. Altre variabili mai lette: `$featureMessagingItems` e le variabili header/footer di Lambda — core-libs/styles/scss/_features.scss:L1-L1, core-libs/styles/scss/theme/lambda/_variables.scss:L46-L49.

16. **Breakpoint sfalsati.** In TS `md: 992` è il massimo di md; in Bootstrap `media-breakpoint-up(md)` parte da 768px. Confondere le due semantiche sposta di uno step — core-libs/storefront/layout/config/default-layout.config.ts:L10-L18, core-libs/styles/vendor/bootstrap/scss/_variables.scss:L207-L213.
17. **SSR.** `ThemeService` e `FeatureStylesService` aggiungono classi via `Renderer2` sul root component; `DirectionService` usa `WindowRef.document`. DA VERIFICARE che classi e `dir` compaiano nell'HTML generato dal server prima dell'idratazione (non verificato sul codice di rendering) — core-libs/storefront/layout/theme/theme.service.ts:L35-L46, core-libs/core/src/features-config/services/feature-styles.service.ts:L130-L136, core-libs/storefront/layout/direction/direction.service.ts:L54-L57. Vedi 07 — SSR, hydration e TransferState e 39 — SSR ottimizzato.

---

10. Come fare debug

- **Il CSS di un componente non c'è.** Ispezionare il CSS compilato (DevTools → Sources sul bundle `styles`): cercare il selettore `cx-xyz`. Se manca, verificare che sia nell'allowlist della sua area e non in `$skipComponentStyles` — core-libs/styles/scss/_components.scss:L37-L44. Se il componente appartiene a una feature lib, cercarlo nel bundle della lib (es. `cart`), non in `styles` — projects/storefrontapp/project.json:L43-L54.
- **Un page template non ha layout.** Controllare la classe su `cx-page-layout` nel DOM e confrontarla con `$page-template-allowlist` — core-libs/styles/scss/_page-template.scss:L13-L19 (caso tipico: `SearchResultsGridPageTemplate`, §9 punto 3).
- **Colori sbagliati.** In DevTools selezionare `<app-root>` (o il root component) e guardare il pannello "Computed" per `--cx-color-primary`: si vede quale regola vince fra `:root` (`root.scss`), la classe tema (`.santorini`, `.cx-theme-high-contrast-*`) e gli override `cxFeat_*` con `!important` — core-libs/styles/scss/root.scss:L3-L70, projects/storefrontapp/src/cms-themes.scss:L1-L60.
- **Quale tema è attivo.** Leggere la classe sul root component; in parallelo controllare lo stato NgRx del site theme (Redux DevTools, selettore `getActiveSiteTheme`) — core-libs/core/src/site-theme/facade/site-theme.service.ts:L99-L104. Se la classe manca, verificare che `ThemeModule` sia caricato tramite `BaseStorefrontModule` → `LayoutModule` — core-libs/storefront/layout/layout.module.ts:L15-L25.
- **Stili `forFeature` che non si attivano.** Verificare (1) il flag nei feature toggle, (2) la presenza di `cxFeat_<flag>` sul root component, (3) che il componente chiami `useFeatureStyles('<flag>')` nel costruttore. Un breakpoint in `FeatureStylesService.registerUsage/addClass` mostra se `rootElement` è già impostato — core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L136. In dev mode il warning "Feature flag CSS: ... is already not used" segnala unregister sbilanciati — core-libs/core/src/features-config/services/feature-styles.service.ts:L105-L109.
- **Direzione.** Ispezionare `document.documentElement.getAttribute('dir')`; se manca, controllare `direction.default` e il codice lingua esatto restituito da `LanguageService.getActive()` (match esatto) — core-libs/storefront/layout/direction/direction.service.ts:L104-L112.
- **Ordine e override di variabili Sass.** Inserire temporaneamente `@debug $theme-colors;` o `@debug $useTabsOnPDP;` subito dopo `@import '@spartacus/styles/scss/core'` nel foglio dell'app per vedere il valore effettivo durante la build (tecnica generica di Sass, non specifica di Spartacus). I test `sass-true` (`npm run test` in `core-libs/styles`) sono un buon modello per isolare il comportamento di un mixin — core-libs/styles/project.json:L2-L14, core-libs/styles/test/features.spec.scss:L1-L30.

11. Domande di autoverifica

1. **Perché** `@spartacus/styles` **non viene compilato in fase di build della libreria?** Perché è distribuito come sorgente Sass (nessun target di build, copia tale e quale in release): la compilazione avviene nell'app consumer, che può così impostare le variabili `!default` prima dell'import. — core-libs/styles/project.json:L2-L14, ci-scripts/release-packer.sh:L42-L43.
2. **Come fa un elemento** `cx-banner` **a ricevere i suoi stili?** Lo stile è nel placeholder `%cx-banner`; `cx-banner` è nella `$content-components-allowlist`, che confluisce in `$selectors`; il ciclo di `_components.scss` emette `cx-banner { @extend %cx-banner !optional; }`. — core-libs/styles/scss/components/content/_index.scss:L10-L14, core-libs/styles/scss/_components.scss:L21-L44.
3. **A cosa servono i placeholder** `%<selettore>__body`**?** A permettere a un componente di applicare regole a livello di `body`, che estende `%<selettore>__body` per ogni selettore in allowlist; nel core l'unico è `%cx-searchbox__body`, legato alla classe `searchbox-is-active`. — core-libs/styles/scss/_components.scss:L46-L53, core-libs/styles/scss/components/product/search/_searchbox.scss:L3-L21.
4. **Perché, pur dichiarando** `$useTabsOnPDP: false !default` **nel componente, i tab del PDP sono attivi di default?** Perché il tema santorini, valutato prima, assegna `$useTabsOnPDP: true !default`; la successiva assegnazione `!default` del componente non ha effetto. — core-libs/styles/scss/theme/santorini/_variables.scss:L169-L174, core-libs/styles/scss/components/content/tab-paragraph-container/_tab-paragraph-container.scss:L1-L2.
5. **Cosa succede se uso** `@include forFeature('myFlag')` **ma non chiamo** `useFeatureStyles('myFlag')`**?** La classe `cxFeat_myFlag` non viene aggiunta al root element (il contatore resta a 0), quindi gli stili non si attivano anche con il flag abilitato. — core-libs/styles/scss/_features.scss:L16-L31, core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L95.
6. **Su quale elemento finisce la classe del site theme e chi la mette?** Sul root component bootstrappato (nell'app demo `app-root`), aggiunta da `ThemeService.setTheme`, avviato da `ThemeModule` come `APP_BOOTSTRAP_LISTENER`. — core-libs/storefront/layout/theme/theme.module.ts:L10-L25, core-libs/storefront/layout/theme/theme.service.ts:L52-L61, projects/storefrontapp/src/app/app.component.ts:L15-L15.
7. **Perché in** `root.scss` **gli override** `a11yFocusIndicatorContrast` **usano** `!important` **e** `:not(.cx-theme-high-contrast-*)`**?** `!important` serve a battere le classi tema CMS (es. `.santorini`) che stanno sullo stesso elemento con pari specificità e sono caricate dopo; le guardie `:not` evitano di toccare i temi high-contrast, che hanno colori di focus propri. — core-libs/styles/scss/root.scss:L47-L64.
8. **Con la config di default, che** `dir` **ottiene** `<html>` **per la lingua** `ar_SA`**?** `ltr`: `rtlLanguages` è `['he','ar']` e il confronto è esatto, quindi `ar_SA` non corrisponde e si ricade su `default: LTR`. — core-libs/storefront/layout/direction/config/default-direction.config.ts:L10-L19, core-libs/storefront/layout/direction/direction.service.ts:L104-L112.
9. **Perché le variabili Sass impostate solo in** `styles.scss` **non influenzano il bundle CSS di** `@spartacus/cart`**?** Perché ogni feature lib ha un bundle separato, compilato a partire dall'entry della lib che reimporta `scss/core` e, in 21 bundle su 24, da `styles-config`: è una compilazione indipendente. — projects/storefrontapp/project.json:L43-L54, projects/storefrontapp/src/styles/lib-cart.scss:L1-L2, feature-libs/cart/_index.scss:L1-L3.
10. **Il breakpoint** `md` **vale 992 in TypeScript e 768px in Bootstrap: è un bug?** No: in `BreakpointService` i numeri sono larghezze massime (md termina a 992), in Bootstrap sono minimi (md parte da 768px). I confini coincidono, la semantica è diversa. — core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L161-L176, core-libs/styles/vendor/bootstrap/scss/_variables.scss:L207-L213.

TAVOLA 38.1 **4.3 Site theme: dallo store alla classe sul root component**

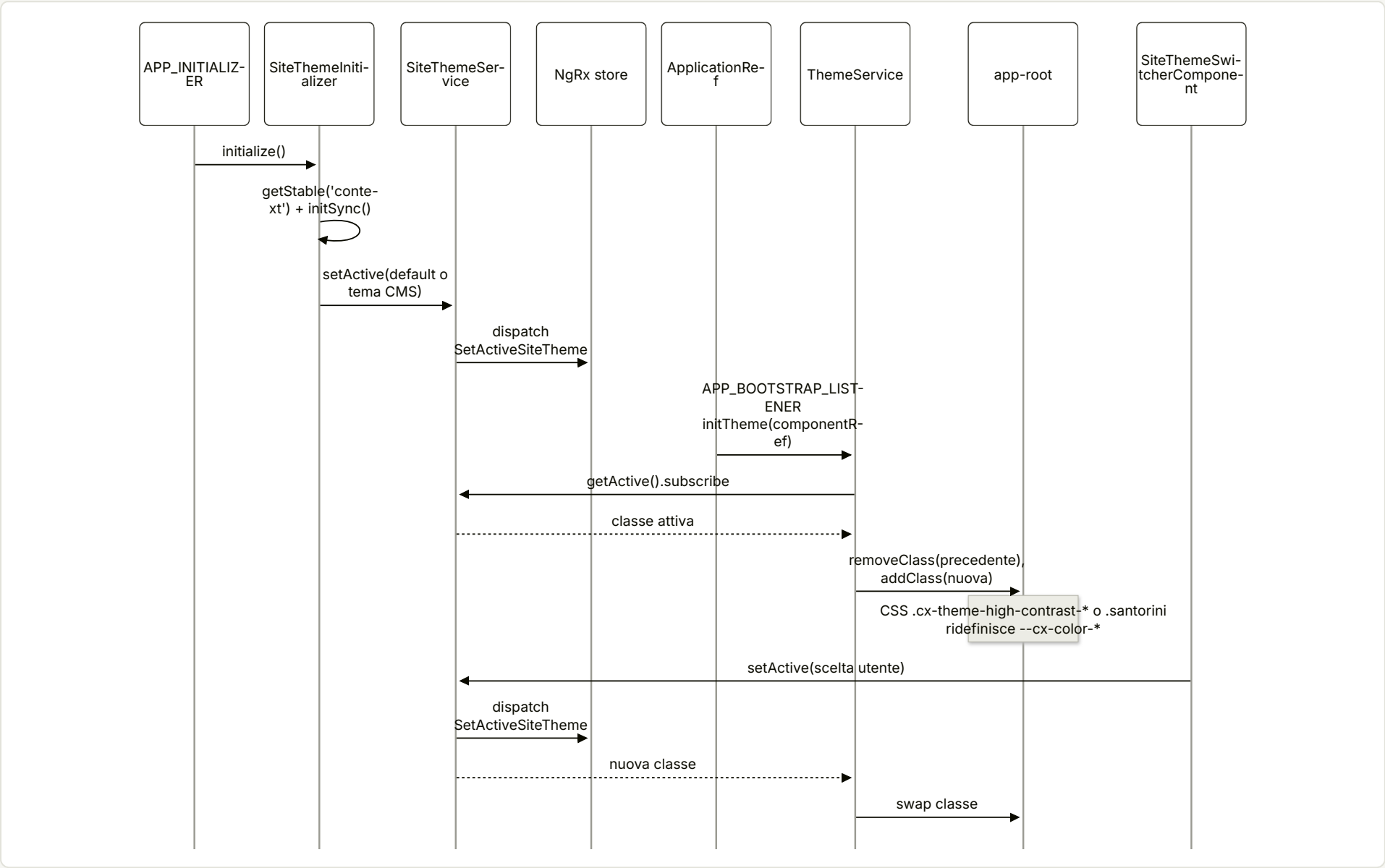
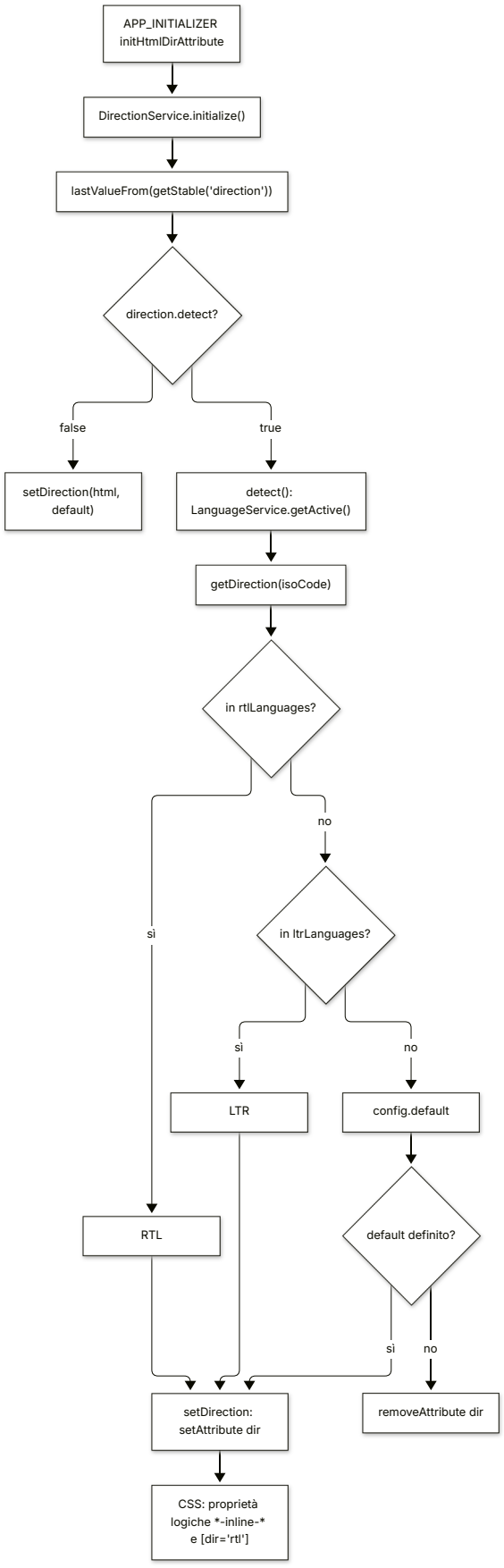


TAVOLA 38.2 4.5 Direzione al bootstrap e al cambio lingua



39

SSR ottimizzato

Cosa succede attorno al render SSR sotto traffico reale.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta). Angular @ `v21.2.23` (prefisso `angular:`); `@angular/ssr` 21.2.23 dal pacchetto npm pubblicato (prefisso `deps:`, path relativi a `/tmp/deps`).

---

1. A cosa serve e perché esiste

Il capitolo 07 spiega **come** Angular e Spartacus producono HTML sul server: `CommonEngine`, la catena `ngExpressEngine` → `CxCommonEngine`, i token di richiesta, TransferState e hydration. Qui ci occupiamo di **cosa succede attorno a quel render** quando il server riceve traffico reale:

- un render SSR di Spartacus può durare secondi (chiamate OCC, CMS, lazy loading) e Angular **non permette di abortirlo** (`core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L380-L388`);
- molte richieste contemporanee possono saturare CPU e memoria del processo Node;
- alcune pagine (checkout, my-account, ASM, SmartEdit) non hanno senso, o sono dannose, se renderizzate sul server;
- gli errori avvenuti dentro l'app Angular devono diventare **status HTTP corretti** (404/500) per Express e per le cache esterne;
- i log devono essere correlabili per richiesta e leggibili da strumenti di monitoring.

La risposta di Spartacus è un insieme di componenti dell'entry point `@spartacus/setup/ssr` (`core-libs/setup/ssr/public_api.ts:L7-L16`):

1. `OptimizedSsrEngine`: un wrapper attorno alla funzione view-engine Express che aggiunge timeout con **fallback CSR**, limite di **concorrenza**, **cache in memoria** dei render, **rendering strategy** per richiesta, condivisione di un render tra richieste identiche (`reuseCurrentRendering`) e protezione dai render "appesi" (`maxRenderTime`) (`core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L37-L57`, `core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L220-L237`).
2. `SsrOptimizationOptions` e i suoi default (`core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L18-L181`).
3. **Logger SSR**: `ExpressServerLogger`/`DefaultExpressServerLogger` (log JSON a riga singola in prod), `ExpressLoggerService` che collega il `LoggerService` dell'app al logger del server, contesto di richiesta con UUID e W3C `traceparent` (`core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L28-L133`, `core-libs/setup/ssr/optimized-engine/request-context.ts:L42-L51`).
4. **Error handling verso Express**: `PROPAGATE_ERROR_TO_SERVER`, `PropagatingToServerErrorHandler`, `defaultExpressErrorHandlers` (`core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32`).
5. **Express utils**: calcolo dell'origin (con `X-Forwarded-Host` fidato) e middleware di validazione dell'origin contro Host header injection e cache poisoning (`core-libs/setup/ssr/express-utils/express-request-origin.ts:L9-L26`, `core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L14-L67`).

Tutto questo è codice **solo-Node**: `optimized-ssr-engine.ts` importa `fs` ed `express` con `/* webpackIgnore: true */` (`core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L7-L9`). La gestione errori lato `@spartacus/core` (interceptor HTTP, `CxErrorHandler`, global message) è in 41 — Errori HTTP, error handling, logger e global message; qui ne vediamo solo l'aggancio al server.

2. Mappa dei file

PATH	RUOLO
core-libs/setup/ssr/public_api.ts	Barrel di @spartacus/setup/ssr
core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts	NgExpressEngineDecorator, decorateExpressEngine, tipi NgExpressEngine/NgExpressEngineInstance
core-libs/setup/ssr/engine-decorator/index.ts	Esporta solo NgExpressEngineDecorator
core-libs/setup/ssr/engine/ng-express-engine.ts	ngExpressEngine (porting da Angular Universal) — vedi cap. 07
core-libs/setup/ssr/engine/cx-common-engine.ts	CxCommonEngine: CommonEngine + PROPAGATE_ERROR_TO_SERVER — vedi cap. 07
core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts	OptimizedSsrEngine, SsrCallbackFn
core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts	SsrOptimizationOptions, RenderingStrategy, defaultSsrOptimizationOptions, getDefaultRenderKey
core-libs/setup/ssr/optimized-engine/get-loggable-ssr-optimization-options.ts	Serializza le opzioni per il log di avvio (non esportato)
core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts	defaultRenderingStrategyResolver
core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver-options.ts	RenderingStrategyResolverOptions, defaultRenderingStrategyResolverOptions
core-libs/setup/ssr/optimized-engine/request-context.ts	RequestContext (uuid, timeReceived, traceContext) in res.locals.cx.request
core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts	RenderingCache: Map FIFO con limite in byte e TTL
core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts	RenderingCacheSizeManager: contabilità dei byte
core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts	DefaultCacheEntrySizeCalculator (2 byte/carattere)
core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.model.ts	RenderingEntry, CacheEntrySizeCalculator
core-libs/setup/ssr/optimized-engine/index.ts, core-libs/setup/ssr/optimized-engine/rendering-cache/index.ts	Barrel
core-libs/setup/ssr/logger/loggers/express-server-logger.ts	Interfaccia ExpressServerLogger, token EXPRESS_SERVER_LOGGER
core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts	DefaultExpressServerLogger (JSON in prod, inspect in dev)
core-libs/setup/ssr/logger/get-logger-inspect-options.ts	Opzioni util.inspect per dev/prod
core-libs/setup/ssr/logger/loggers/w3c-trace-context/parse-traceparent.ts	Parsing header traceparent
core-libs/setup/ssr/logger/loggers/w3c-trace-context/w3c-trace-context.model.ts	W3cTraceContext

PATH	RUOLO
core-libs/setup/ssr/logger/loggers/w3c-trace-context/errors/invalid-traceparent-format-error.ts, core-libs/setup/ssr/logger/loggers/w3c-trace-context/errors/invalid-traceparent-length-error.ts	Errori di parsing del `traceparent`
core-libs/setup/ssr/logger/services/express-logger.service.ts	`ExpressLoggerService` (il `LoggerService` in SSR)
core-libs/setup/ssr/logger/services/prerendering-logger.service.ts	`PrerenderingLoggerService` (console)
core-libs/setup/ssr/logger/services/server-logger-service-factory.ts	Scelta Express vs prerendering
core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts	Token `PROPAGATE_ERROR_TO_SERVER`
core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts	`PropagatingToServerErrorHandler`
core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts	`defaultExpressErrorHandlers` (404/500 + documento CSR)
core-libs/setup/ssr/express-utils/express-request-origin.ts	`getRequestOrigin` (trust proxy / `X-Forwarded-Host`)
core-libs/setup/ssr/express-utils/express-request-url.ts	`getRequestUrl` (= chiave di rendering di default)
core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts	`getOriginValidationMiddleware`, `OriginValidationOptions`
core-libs/setup/ssr/providers/ssr-providers.ts	`provideServer`, `getServerRequestProviders` — vedi cap. 07
core-libs/setup/ssr/providers/model.ts, core-libs/setup/ssr/providers/server-request-origin.ts, core-libs/setup/ssr/providers/server-request-url.ts	`ServerOptions` e factory dei token di richiesta — vedi cap. 07
core-libs/setup/ssr/tokens/express.tokens.ts	Token `REQUEST`, `RESPONSE` (Express)
core-libs/setup/ssr/testing/test-config-server.module.ts	`TestConfigServerModule` (solo e2e)
core-libs/storefront/utils/directive-state-transfer.service.ts	`DirectiveStateTransferService` — vedi cap. 07
projects/storefrontapp/src/server.ts	Server Express della demo: uso di riferimento di tutto quanto sopra

3. Come funziona passo per passo

3.1 Bootstrap: dal `server.ts` all'engine ottimizzato

1. La demo costruisce `ssrOptions` da variabili d'ambiente: `timeout` da `SSR_TIMEOUT` (default `defaultSsrOptimizationOptions.timeout`) e `cache` da `SSR_CACHE === 'true'` (projects/storefrontapp/src/server.ts:L22-L27).
2. `NgExpressEngineDecorator.get(engine, ssrOptions)` delega a `decorateExpressEngine` (projects/storefrontapp/src/server.ts:L29-L29, core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L43-L48). Il secondo argomento accetta `null`: con `undefined` scatta il valore di default del parametro (`defaultSsrOptimizationOptions`, quindi ottimizzazione attiva), con `null` l'ottimizzazione è di-

sattivata (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L57, core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L69-L72).

3. Quando Express registra il view engine (`server.engine('html', ngExpressEngine({ bootstrap }))`) la funzione decorata crea l'istanza di `ngExpressEngine` anteponendo `getServerRequestProviders()` ai provider di setup (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L58-L66, projects/storefrontapp/src/server.ts:L51-L56). `ngExpressEngine` crea un unico `CxCommonEngine` (core-libs/setup/ssr/engine/ng-express-engine.ts:L82-L88), il cui costruttore legge **NG_ALLOWED_HOSTS** (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L112). I dettagli della catena sono nel cap. 07 §3.3.
4. Con opzioni truthy si istanzia `new OptimizedSsrEngine(engineInstance, optimizationOptions)` e si restituisce a Express il suo `engineInstance` (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L69-L72), cioè `renderResponse` legato all'istanza (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L55-L57).
5. Il costruttore fonde **in modo shallow** i default con le opzioni utente, fondendo a parte `ssrFeatureToggles` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L63-L74); se dopo il merge manca `logger` lancia `'SsrOptimizationOptions.logger is not defined'`; poi crea la `RenderingCache` e logga le opzioni (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L76-L81).
6. `logOptions()` scrive "[spartacus] SSR optimization engine initialized" con le opzioni rese stampabili da `getLoggableSsrOptimizationOptions` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L84-L96), che sostituisce le istanze di classe con il nome del costruttore e le funzioni con il loro `toString()` (core-libs/setup/ssr/optimized-engine/get-loggable-ssr-optimization-options.ts:L16-L29). "Istanza di classe" significa `typeof value === 'object' && value.constructor !== Object` (core-libs/setup/ssr/optimized-engine/get-loggable-ssr-optimization-options.ts:L37-L39). Il tipo `DefaultSsrOptimizationOptions` rende obbligatorie tutte le chiavi proprio per poterle stampare tutte all'avvio (core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L201-L218).

Il resto della demo: `trust proxy = loopback` e middleware di validazione dell'origin **prima** dell'engine (projects/storefrontapp/src/server.ts:L39-L49), statici con `maxAge: '1y'` per i path con estensione, tutte le altre rotte renderizzate con `res.render(indexHtml, { req, providers: [APP_BASE_HREF] })` (projects/storefrontapp/src/server.ts:L61-L75), e infine `defaultExpressErrorHandlers(indexHtmlContent)` come ultimo middleware (projects/storefrontapp/src/server.ts:L77-L77). `indexHtml` è `index.server.html` nella cartella server di `dist` (projects/storefrontapp/src/server.ts:L34-L37): lo stesso file è il template del render **e** il documento del fallback CSR.

3.2 `renderResponse`: la pipeline di ogni richiesta

`renderResponse(filePath, options, callback)` produce uno di tre esiti: fallback CSR con `index.html`, render dell'engine Angular, risultato dalla cache in memoria (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L232-L239).

1. **Contesto di log.** `preprocessRequestForLogger(options.req, this.logger)` assegna alla richiesta UUID, timestamp e trace context; poi legge `request = options.req` e `response = options.req.res` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L245-L248). Dettagli in §3.8.
2. **Cache.** `returnCachedRender` calcola la chiave e, se `renderingCache.isReady(key)`, chiama `callback(cached.err, cached.html)`; con `cache` disabilitata cancella subito l'entry (consumo singolo). Se ha risposto, `renderResponse` logga "Render from cache" e termina (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L212-L230, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L250-L255).
3. **Si deve renderizzare?** Se `shouldRender(request)` è falso si risponde con `fallbackToCsr` e si termina (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L256-L259). Nessun render parte in

background: lo spec lo verifica per `ALWAYS_CSR` anche con `cache: true` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.spec.ts:L731-L749).

4. **Timeout o CSR immediato.**
   - Se `shouldTimeout(request)` è vero, si arma un `setTimeout` di `getTimeout(request)` ms che, a scadenza, azzera `requestTimeout`, invia il fallback CSR e logga "SSR rendering exceeded timeout ..." (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L261-L272).
   - Altrimenti si risponde **subito** con il CSR ma **senza** `return`: il render prosegue in background per riempire la cache (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L273-L279).
5. **Callback del render.** Si costruisce `renderCallback(err, html)` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L281-L300):
   - se `requestTimeout` è ancora attivo (il client attende): `clearTimeout`, `callback(err, html)` al client, log del risultato, poi `store` solo con `cache: true`, altrimenti `clear` della chiave;
   - se la richiesta ha già ricevuto il CSR: `store` **sempre**, "for future use".
6. **Delega.** `handleRender({ filePath, options, renderCallback, request })` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L302-L307), che a sua volta usa `startRender` (§3.5, §3.6).

Il fallback CSR imposta `Cache-Control: no-store` (così le cache esterne non riusano la pagina CSR) e invia il documento letto da `filePath` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L98-L110). `getDocument` legge il file con `fs.readFileSync` solo la prima volta e lo memorizza in `templateCache` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L310-L320).

3.3 Le decisioni: chiave, strategia, concorrenza, timeout

- **Chiave di rendering.** `getRenderingKey` usa `renderKeyResolver` se presente, altrimenti `getDefaultRenderKey` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L112-L116). `getDefaultRenderKey` è un alias di `getRequestUrl` (core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L196-L199), cioè `getRequestOrigin(req) + req.originalUrl` (core-libs/setup/ssr/express-utils/express-request-url.ts:L10-L12): **include host e query string**.
- **Strategia.** `getRenderingStrategy` usa `renderingStrategyResolver` se presente, altrimenti `RenderingStrategy.DEFAULT` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L118-L122). L'enum è numerico: `ALWAYS_CSR = -1`, `DEFAULT = 0`, `ALWAYS_SSR = 1` (core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L183-L187).
- `shouldRender`. Calcola `fallBack` (render già in corso per la chiave **e** `reuseCurrentRendering` disattivo) e `concurrencyLimitExceeded`, logga la causa (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L133-L151), e restituisce: render se nessun impedimento e strategia ≠ `ALWAYS_CSR`, **oppure sempre** se la strategia è `ALWAYS_SSR` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L153-L158).
- **Concorrenza.** `isConcurrencyLimitExceeded` restituisce `false` se con `reuseCurrentRendering` esiste già un render per la chiave (la richiesta non occupa un nuovo slot); altrimenti confronta `currentConcurrency >= concurrency`, e con `concurrency` falsy non c'è limite (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L166-L180).
- **Timeout.** `shouldTimeout` è vero se `timeout` è non zero **o** la strategia è `ALWAYS_SSR` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L187-L192); `getTimeout` restituisce `forcedSsrTimeout ?? 60000` per `ALWAYS_SSR`, altrimenti `timeout ?? 0` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L200-L204).

Con `timeout: 0` si ottiene quindi "CSR sempre, SSR per la prossima": lo spec verifica che la prima risposta sia il CSR, che il render resti in corso (`currentConcurrency: 1`) e che la seconda richiesta allo stesso URL riceva l'HTML (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.spec.ts:L292-L316).

3.4 Rendering strategy resolver di default

`defaultRenderingStrategyResolver(options)` è una factory che restituisce `(request) => RenderingStrategy` (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L62-L72):

1. se l'URL contiene una voce della lista fissa interna `defaultAlwaysCsrOptions.excludedUrls = ['cx-preview']` → `ALWAYS_CSR` (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L11-L13, core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L65-L67);
2. altrimenti, se `shouldFallbackToCsr` (parametri query o URL esclusi) → `ALWAYS_CSR`, altrimenti `DEFAULT` (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L41-L49, core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L69-L71).

Non restituisce mai `ALWAYS_SSR`. `hasExcludedParams` confronta i **nomi** delle proprietà di `request.query` con `excludedParams` (match esatto) (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L15-L30); `hasExcludedUrl` usa `request.url.search(url) > -1`, quindi ogni voce è una **espressione regolare** cercata come sottostringa (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L32-L39). I default escludono gli URL `checkout`, `my-account`, `punchout`, `opf` e il parametro `asm` (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver-options.ts:L12-L16).

3.5 `handleRender` e `reuseCurrentRendering`

- Senza `reuseCurrentRendering`, `handleRender` chiama direttamente `startRender` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L340-L348).
- Con `reuseCurrentRendering` (default `true`, core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L229-L229), il `renderCallback` della richiesta viene accodato in `renderCallbacks[renderingKey]` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L350-L354). Solo se non c'è già un render in corso si chiama `startRender` con un callback che, a fine render, **rilegge la Map** (non una copia locale, perché `maxRenderTime` può averla svuotata nel frattempo), invoca tutte le callback con lo stesso `(err, html)` e cancella la voce (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L356-L372).
- In questo ramo viene sempre loggato "Request is waiting for the SSR rendering to complete", anche per la richiesta che ha avviato il render (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L374-L377).

Ogni richiesta in attesa mantiene **il proprio** timer di timeout (§3.2 passo 4): il JSDoc fa l'esempio di timeout 3 s, render 4 s, seconda richiesta arrivata dopo 2 s che quindi riceve l'SSR (core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L110-L132). `renderCallbacks` è usata solo in questa modalità (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L43-L53).

3.6 `startRender` e `maxRenderTime`

1. Arma un timer `maxRenderTime` (default 300000 ms) che, a scadenza, libera la chiave in cache, azzera il timer, decrementa `currentConcurrency`, elimina le callback in attesa (con `reuseCurrentRendering`) e logga "... This might cause memory leaks!" (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L402-L417). Il JSDoc ammette che non esiste modo di abortire il render: superato il limite, il task è considerato "hanging" e il suo esito sarà ignorato (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L380-L388).
2. Logga "Rendering started", marca la chiave come `rendering` e incrementa `currentConcurrency` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L419-L421).
3. Antepone ai `providers` del render un provider `EXPRESS_SERVER_LOGGER` con il logger dell'engine (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L423-L432). È l'**unico** provider di quel token

nel codice non-spec del repo (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L426-L429).

4. Chiama `this.expressEngine(filePath, options, cb)`, cioè la funzione di `ngExpressEngine`. Al completamento: se il timer `maxRenderTime` è già scaduto, logga "completed after the specified maxRenderTime, therefore it was ignored" ed esce; altrimenti `clearTimeout`, log "Rendering completed", `currentConcurrency--`, `renderCallback(err, html)` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L434-L451).

`ngExpressEngine` calcola l'URL da `req.protocol`, header `host`, `req.baseUrl`, `req.url`, aggiunge `REQUEST`/`RESPONSE` e chiama `CxCommonEngine.render`; il risultato arriva come `callback(null, html)` o `.catch(callback)`, e gli errori sincroni sono passati solo se `instanceof Error` (core-libs/setup/ssr/engine/ng-express-engine.ts:L101-L126). Tutti i provider del render (compreso `EXPRESS_SERVER_LOGGER`) finiscono nel platform injector di `CommonEngine` (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L185-L210; vedi cap. 07 §3.2).

`logForRenderResult` usa `log` se c'è HTML ed `error` (con `error: err` nel contesto) se c'è un errore (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L457-L474).

3.7 `RenderingCache` e dimensionamento

`RenderingCache` è una `Map<string, RenderingEntry>` con un `RenderingCacheSizeManager` creato pigramente (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L11-L23). `RenderingEntry` ha `html?`, `err?`, `time?`, `rendering?`, `_size?` (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.model.ts:L10-L20).

- `setAsRendering(key)`: prima `clear(key)` (per rilasciare la dimensione dell'entry precedente), poi il placeholder `{ rendering: true }` (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L25-L32). Gli spec verificano che sovrascrivere una chiave con `store` rilasci la dimensione precedente, altrimenti `usedSize` "drifts upward forever" (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.spec.ts:L178-L190), e che `setAsRendering` seguito da `store` non accumuli la dimensione (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.spec.ts:L192-L200).
- `isRendering(key)`: entry con `rendering` truthy (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L34-L36).
- `store(key, err, html)`: crea `{ err, html }`, aggiunge `time` solo se `ttl` è impostato, chiama `clear(key)`, e se `shouldCacheRenderingResult` restituisce falso (o manca) **non salva**; altrimenti `storeUsingMemoryLimit` (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L41-L64).
- `storeUsingMemoryLimit`: calcola la dimensione, scarta l'entry se più grande dell'intero limite, libera le entry più vecchie finché c'è spazio, salva con `_size` e traccia (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L71-L83). La liberazione rimuove la **prima chiave in ordine di inserimento** (FIFO) (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L88-L103); `get` non riordina, quindi non è LRU (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L105-L107).
- `clear(key)`: elimina e sottrae `entry?._size ?? 0` (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L109-L114).
- `isReady(key)`: entry con `html` o `err` truthy **e** fresca; `isFresh` è sempre vero senza `ttl`, altrimenti `Date.now() - time < ttl` (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L116-L127).

`RenderingCacheSizeManager` tiene `usedSize` e `CACHE_SIZE_LIMIT = options?.cacheSizeMemory ?? 0` (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L14-L29); valida le opzioni **loggando** (non lanciando) se il calculator non ha `calculateSize` o se `cacheSizeMemory` non è un numero

> 0 (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L35-L54). `hasSpaceForEntrySize(n)` è `n <= LIMIT - usedSize`, `isEntryLargerThanCacheLimit(n)` è `n > LIMIT` (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L66-L76); `trackEntrySize` satura a `Number.MAX_SAFE_INTEGER` e `untrackEntrySize` riporta a 0 i valori negativi, loggando in entrambi i casi (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L81-L109).

`DefaultCacheEntrySizeCalculator.calculateSize` usa la dimensione dell'HTML, altrimenti dell'errore, altrimenti 0 (core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L27-L36); per gli errori somma solo `name`, `message`, `stack` e avverte della sottostima (core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L18-L23, core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L51-L60); le stringhe valgono `2 * str.length` byte, scelta motivata da heap snapshot che mostrano `SeqTwoByteString` (core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L62-L74).

3.8 Contesto di richiesta e trace

- `RequestContext = { uuid, timeReceived, traceContext? }` (core-libs/setup/ssr/optimized-engine/request-context.ts:L17-L21), letto da `getRequestContext(request)` in `request.res?.locals?.cx?.request` (core-libs/setup/ssr/optimized-engine/request-context.ts:L28-L30).
- `preprocessRequestForLogger` crea il contesto con `randomUUID()` e timestamp ISO più il trace context e lo scrive in `res.locals.cx.request` (core-libs/setup/ssr/optimized-engine/request-context.ts:L42-L51, core-libs/setup/ssr/optimized-engine/request-context.ts:L75-L81); `setRequestContext` scrive solo se `request.res` esiste (core-libs/setup/ssr/optimized-engine/request-context.ts:L58-L68). Il tipo `Locals` di Express è esteso per module augmentation (core-libs/setup/ssr/optimized-engine/request-context.ts:L105-L111).
- `getTraceContext` chiama `parseTraceparent(request.get('traceparent'))`; un'eccezione viene loggata con `logger.error` e il contesto resta senza trace (core-libs/setup/ssr/optimized-engine/request-context.ts:L90-L103).
- `parseTraceparent` restituisce `undefined` se l'header non è stringa, lancia `InvalidTraceparentLengthError` se la lunghezza ≠ 55, `InvalidTraceparentFormatError` se non rispetta il pattern esadecimale, altrimenti `{ version, traceId, parentId, traceFlags }` (core-libs/setup/ssr/logger/loggers/w3c-trace-context/parse-traceparent.ts:L11-L47, core-libs/setup/ssr/logger/loggers/w3c-trace-context/w3c-trace-context.model.ts:L18-L23). I messaggi sono "Traceparent header has invalid format." e "Traceparent header has invalid length: N. Expected 55 characters." (core-libs/setup/ssr/logger/loggers/w3c-trace-context/errors/invalid-traceparent-format-error.ts:L10-L14, core-libs/setup/ssr/logger/loggers/w3c-trace-context/errors/invalid-traceparent-length-error.ts:L11-L17).

3.9 Logger SSR

Due livelli distinti:

Logger del server (`ExpressServerLogger`). Interfaccia con `log/warn/error/info/debug(message, context)`, dove `ExpressServerLoggerContext` ha `request?` e chiavi libere (core-libs/setup/ssr/logger/loggers/express-server-logger.ts:L14-L36). `DefaultExpressServerLogger` instrada ogni livello al `console.*` corrispondente (core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L28-L48). `stringifyWithContext` produce, in dev mode, un output `formatWithOptions` multi-linea; in prod un **JSON a riga singola** con le `Error` convertite da `stringifyError` (core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L63-L86). `mapContext` aggiunge `timestamp` e sostituisce `request` con `mapRequest`, che produce `{ url: request.originalUrl, ...getRequestContext(request) }` (core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L95-L122). `getLoggerInspectOptions` usa `depth: 10` e in prod `breakLength: Infinity`, per

evitare che strumenti come Kibana interpretino più righe come log diversi (core-libs/setup/ssr/logger/get-logger-inspect-options.ts:L16-L58). La modalità dipende da `isDevMode()` di Angular, vero se `ngDevMode` è `undefined` o truthy (angular:packages/core/src/util/is_dev_mode.ts:L20-L22).

Logger dell'app (`LoggerService`). `provideServer` registra `LoggerService` con `useFactory: serverLoggerServiceFactory` (core-libs/setup/ssr/providers/ssr-providers.ts:L37-L40). La factory sceglie `ExpressLoggerService` se `REQUEST` è iniettabile, altrimenti `PrerenderingLoggerService` (core-libs/setup/ssr/logger/services/server-logger-service-factory.ts:L12-L17). `ExpressLoggerService` inietta `REQUEST` ed `EXPRESS_SERVER_LOGGER`, formatta gli argomenti e passa `{ request }` come contesto (core-libs/setup/ssr/logger/services/express-logger.service.ts:L24-L61); `PrerenderingLoggerService` inoltra a `console.*` (core-libs/setup/ssr/logger/services/prerendering-logger.service.ts:L15-L39). Il ponte tra i due livelli è il provider `EXPRESS_SERVER_LOGGER` iniettato da `startRender` (§3.6).

3.10 Errori: dall'app Angular allo status HTTP

1. Sul server, `HttpErrorHandlerInterceptor` passa all'`ErrorHandler` gli errori HTTP, incapsulati in `CmsPageNotFoundOutboundHttpError` (404 su endpoint `pages`) o `OutboundHttpError` (core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L48-L98).
2. `CxErrorHandler` inoltra a tutti i `MULTI_ERROR_HANDLER` (core-libs/core/src/error-handling/cx-error-handler.ts:L18-L30); `provideServer` registra tra questi `PropagatingToServerErrorHandler` via `useExisting` (core-libs/setup/ssr/providers/ssr-providers.ts:L41-L45), che chiama `PROPAGATE_ERROR_TO_SERVER` o un no-op se assente (core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts:L26-L38). Il token è fornito OOTB solo da `CxCommonEngine` (core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L9-L22).
3. `CxCommonEngine.render` memorizza il **primo** errore e rigetta la promise solo a render completato (core-libs/setup/ssr/engine/cx-common-engine.ts:L41-L66).
4. `ngExpressEngine` → `.catch(callback)` → `startRender` → `renderCallback(err, undefined)` (core-libs/setup/ssr/engine/ng-express-engine.ts:L120-L123, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L450-L450).
5. Se il client attende ancora: `clearTimeout`, `callback(err)` verso Express, poi log `error`. Con `cache: true` `store` scarta l'errore per il default `shouldCacheRenderingResult` (`!err`); con `cache: false` (default) la chiave viene semplicemente cancellata con `clear` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L283-L295, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L54-L61).
6. Express inoltra l'errore al middleware d'errore. DA VERIFICARE: è il comportamento di `res.render` senza callback in Express, i cui sorgenti non sono disponibili in `/tmp/deps`. `defaultExpressErrorHandlers` — se gli header non sono già inviati — imposta `Cache-Control: no-store` e invia il documento CSR con 404 per `CmsPageNotFoundOutboundHttpError`, 500 per ogni altro errore (core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32).

I test SSR E2E verificano proprio 404 per pagina inesistente e per errore su `cms/pages`, 500 per altri errori HTTP e per timeout del backend (projects/ssr-tests/src/ssr-testing.spec.ts:L19-L130). Il resto della catena lato `@spartacus/core` è descritto in 41.

3.11 Express utils: origin e sua validazione

`getRequestOrigin` usa `X-Forwarded-Host` (prima voce se CSV) solo se la `trust proxy fn` di Express si fida dell'indirizzo remoto, altrimenti l'header `host` (core-libs/setup/ssr/express-utils/express-request-origin.ts:L9-L26). Poiché l'origin entra nella chiave di cache (§3.3), un Host header manipolato potrebbe creare o avvelenare entry: da qui `getOriginValidationMiddleware`, che restituisce un middleware no-op se la allowlist

è vuota e altrimenti risponde `400 Bad Request` con `Cache-Control: no-store` quando l'origin non è ammesso (core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L51-L67). La allowlist accetta array o CSV (tipicamente `SSR_ALLOWED_ORIGINS`), origin completi senza slash finale, confronto case-insensitive, e `*` che copre **esattamente un'etichetta** e mai l'apex (core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L14-L50, core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L69-L99).

A un livello diverso opera `allowedHosts` di `CommonEngine` (variabile **NG_ALLOWED_HOSTS** con precedenza sull'opzione): se l'hostname non è ammesso e nessuna lista è configurata, `CommonEngine.render` logga un errore e **restituisce il documento non renderizzato come successo**; se una lista è configurata, lancia (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L135). Gli script `serve:ssr*` impostano `NG_ALLOWED_HOSTS=localhost` (package.json:L152-L155).

---

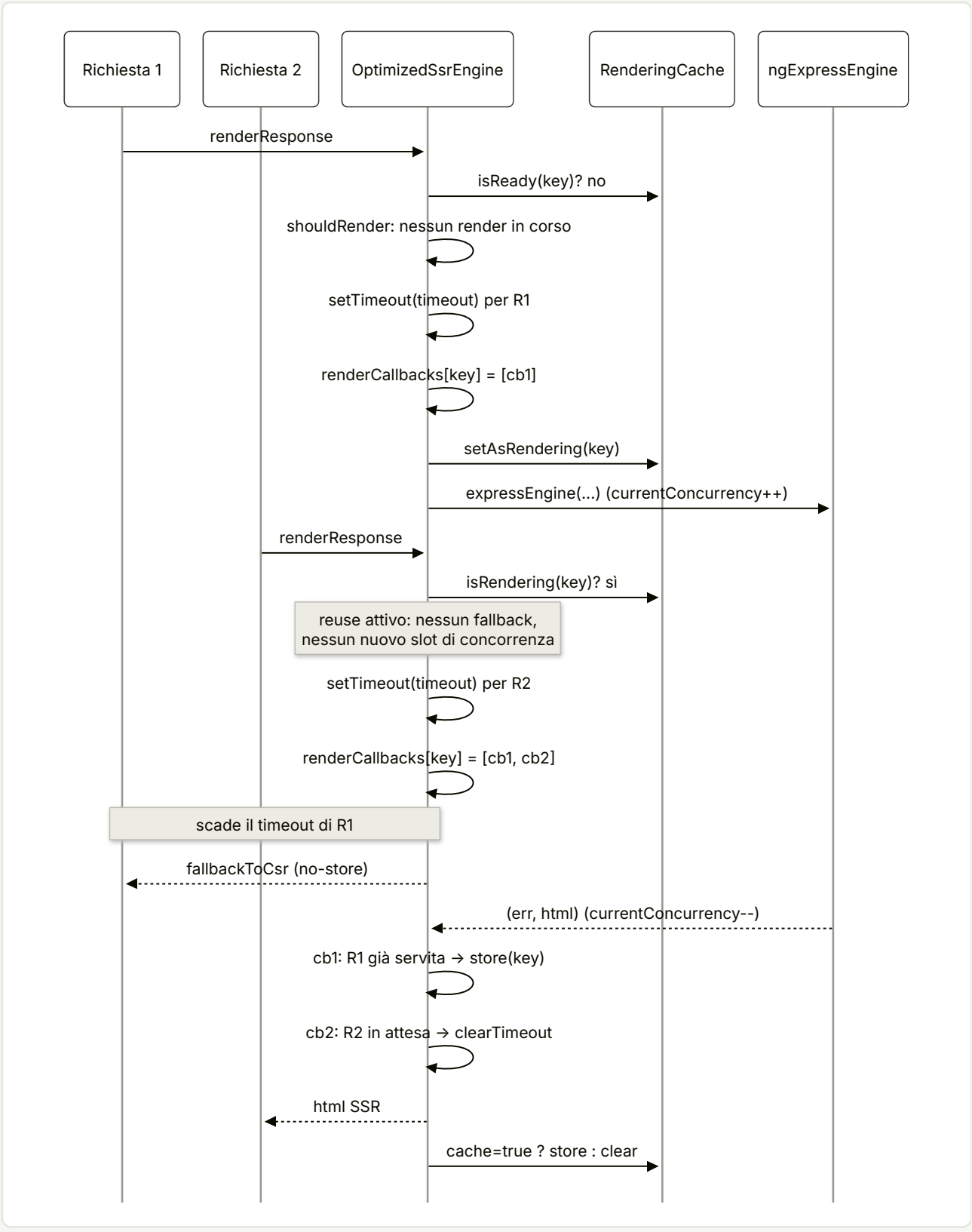
4. Diagrammi

4.1 Decisioni di `renderResponse`

■ **TAVOLA 39.1** diagramma a tutta pagina, in fondo al capitolo

Ogni nodo corrisponde a §3.2–§3.6: cache (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L212-L230), `shouldRender` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L133-L158), timeout (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L261-L279), callback (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L281-L300), reuse (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L340-L372), `startRender` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L400-L451).

4.2 `reuseCurrentRendering` con timeout (due richieste, stessa chiave)



Fondamento: `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L137-L139`, `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L170-L175`, `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L261-L272`, `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.`

ts:L282-L300, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L350-L372. Si noti l'ultimo passo: con `cache: false`, la `clear` di cb2 cancella l'entry appena salvata da cb1 (vedi §9, trappola 13).

4.3 Errore applicativo → risposta HTTP

■ **TAVOLA 39.2** diagramma a tutta pagina, in fondo al capitolo

Fondamento: §3.10.

4.4 Inserimento in `RenderingCache`

■ **TAVOLA 39.3** diagramma a tutta pagina, in fondo al capitolo

Fondamento: core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L41-L103.

---

5. Estratti di codice reale

5.1 Merge delle opzioni e logger obbligatorio

core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L59-L82

```ts
constructor(
  protected expressEngine: NgExpressEngineInstance,
  protected ssrOptions?: SsrOptimizationOptions
) {
  this.ssrOptions = ssrOptions
    ? {
        ...defaultSsrOptimizationOptions,
        // overrides the default options
        ...ssrOptions,
        // merge feature toggles
        ssrFeatureToggles: {
          ...defaultSsrOptimizationOptions.ssrFeatureToggles,
          ...ssrOptions.ssrFeatureToggles,
        },
      }
    : undefined;

  if (!this.ssrOptions?.logger) {
    throw new Error(``SsrOptimizationOptions.logger` is not defined');
  }
  this.logger = this.ssrOptions?.logger;
  this.renderingCache = new RenderingCache(this.ssrOptions);
  this.logOptions();
}
```

5.2 Il cuore di `renderResponse`

core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L250-L300

TS

```
if (this.returnCachedRender(request, callback)) {
  this.logger.log(`Render from cache (${request?.originalUrl})`, {
    request,
  });
  return;
}
if (!this.shouldRender(request)) {
  this.fallbackToCsr(response, filePath, callback);
  return;
}

let requestTimeout: ReturnType<typeof setTimeout> | undefined;
if (this.shouldTimeout(request)) {
  // establish timeout for rendering
  const timeout = this.getTimeout(request);
  requestTimeout = setTimeout(() => {
    requestTimeout = undefined;
    this.fallbackToCsr(response, filePath, callback);
    // ...
  }, timeout);
} else {
  // Here we respond with the fallback to CSR, but we don't `return`.
  // We let the actual rendering task to happen in the background

  // to eventually store the rendered result in the cache.
  this.fallbackToCsr(response, filePath, callback);
}

const renderingKey = this.getRenderingKey(request);
const renderCallback: SsrCallbackFn = (err, html): void => {
  if (requestTimeout) {
    // if request is still waiting for render, return it
    clearTimeout(requestTimeout);
    callback(err, html);

    this.logForRenderResult(err, html, request);

    // store the render only if caching is enabled
    if (this.ssrOptions?.cache) {
      this.renderingCache.store(renderingKey, err, html);
    } else {
      this.renderingCache.clear(renderingKey);
    }
  } else {
    // store the render for future use
```

```
        this.renderingCache.store(renderingKey, err, html);
      }
    };
```

5.3 shouldRender: ALWAYS_SSR scavalca tutto

core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L153-L158

```ts
    return (
      (!fallBack &&
        !concurrencyLimitExceeded &&
        this.getRenderingStrategy(request) !== RenderingStrategy.ALWAYS_CSR) ||
      this.getRenderingStrategy(request) === RenderingStrategy.ALWAYS_SSR
    );
```

5.4 Fallback CSR

core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L98-L110

```ts
  /**
   * When SSR page can not be returned in time, we're returning index.html of
   * the CSR application.
   * The CSR application is returned with the "Cache-Control: no-store" response-header. This notifies
external cache systems to not use the CSR application for the subsequent request.
   */
  protected fallbackToCsr(
    response: Response,
    filePath: string,
    callback: SsrCallbackFn
  ): void {
    response.set('Cache-Control', 'no-store');
    callback(undefined, this.getDocument(filePath));
  }
```

5.5 Condivisione del render

core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L350-L372

```ts
    const renderingKey = this.getRenderingKey(request);
    if (!this.renderCallbacks.has(renderingKey)) {
      this.renderCallbacks.set(renderingKey, []);
    }
    this.renderCallbacks.get(renderingKey)?.push(renderCallback);

    if (!this.renderingCache.isRendering(renderingKey)) {
      this.startRender({
        filePath,
```

```
        options,
        request,
        renderCallback: (err, html) => {
          // Share the result of the render with all awaiting requests for the same key:

          // Note: we access the Map at the moment of the render finished (don't store value in a local variable),
          //       because in the meantime something might have deleted the value (i.e. when `maxRenderTime` passed).
          this.renderCallbacks
            .get(renderingKey)
            ?.forEach((cb) => cb(err, html)); // pass the shared result to all waiting rendering callbacks
          this.renderCallbacks.delete(renderingKey);
        },
      });
    }
```

5.6 `maxRenderTime` e iniezione del logger

core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L405-L432

TS

```
    let maxRenderTimeout: ReturnType<typeof setTimeout> | undefined =
      setTimeout(() => {
        this.renderingCache.clear(renderingKey);
        maxRenderTimeout = undefined;
        this.currentConcurrency--;
        if (this.ssrOptions?.reuseCurrentRendering) {
          this.renderCallbacks.delete(renderingKey);
        }
        this.logger.log(
          `Rendering of ${request?.originalUrl} was not able to complete. This might cause memory leaks!`,
          { request }
        );
      }, this.ssrOptions?.maxRenderTime ?? 300000); // 300000ms == 5 minutes

    this.logger.log(`Rendering started (${request?.originalUrl})`, { request });
    this.renderingCache.setAsRendering(renderingKey);
    this.currentConcurrency++;

    options = {
      ...options,
      providers: [
        {
          provide: EXPRESS_SERVER_LOGGER,
          useValue: this.logger,
        },
```

```
        ...(options?.providers ?? []),
      ],
    };
```

5.7 Default delle opzioni

core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L220-L237

```ts
export const defaultSsrOptimizationOptions: DefaultSsrOptimizationOptions = {
  cache: false,
  cacheSizeMemory: 800_000_000,
  cacheEntrySizeCalculator: new DefaultCacheEntrySizeCalculator(),
  ttl: undefined,
  concurrency: 10,
  timeout: 3_000,
  forcedSsrTimeout: 60_000,
  maxRenderTime: 300_000,
  reuseCurrentRendering: true,
  renderingStrategyResolver: defaultRenderingStrategyResolver(
    defaultRenderingStrategyResolverOptions
  ),
  logger: new DefaultExpressServerLogger(),
  shouldCacheRenderingResult: ({ entry: { err } }) => !err,
  renderKeyResolver: getDefaultRenderKey,
  ssrFeatureToggles: {},
};
```

5.8 Resolver di strategia

core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L32-L39 e core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L62-L72

```ts
const hasExcludedUrl = (
  request: Request,
  excludedUrls: string[] | undefined
) => {
  return request.url && excludedUrls
    ? excludedUrls.some((url) => request.url.search(url) > -1)
    : false;
};
// ...
export const defaultRenderingStrategyResolver =
  (options: RenderingStrategyResolverOptions) =>
  (request: Request): RenderingStrategy => {
    if (hasExcludedUrl(request, defaultAlwaysCsrOptions.excludedUrls)) {
      return RenderingStrategy.ALWAYS_CSR;
    }
```

```
    return shouldFallbackToCsr(request, options)
      ? RenderingStrategy.ALWAYS_CSR
      : RenderingStrategy.DEFAULT;
  };
```

5.9 Eviction FIFO

core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L88-L103

```ts
  private clearOldestEntriesForSize(requiredSize: number): void {
    while (
      !this.sizeManager.hasSpaceForEntrySize(requiredSize) &&
      this.renders.size > 0
    ) {
      const oldestKey = this.renders.keys().next().value;
      if (oldestKey === undefined) {
        // Still not enough space, but no more entries to clear.
        // In practice this case should never happen, because initially we checked
`isEntryLargerThanCacheLimit` -
        // we validated whether the new entry's size is not bigger than the total cache size limit.
        break;
      }

      this.clear(oldestKey);
    }
  }
```

5.10 Log JSON in prod

core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L63-L86

```ts
  protected stringifyWithContext(
    message: string,
    context: ExpressServerLoggerContext
  ): string {
    const logObject = { message, context: this.mapContext(context) };

    if (isDevMode()) {
      // In dev mode, we want a *human-readable* string representation of the log object
      // that can be printed in the console.
      return formatWithOptions(getLoggerInspectOptions(), logObject);
    }

    // In prod mode, we want a *single-line* JSON which is machine-readable - easy to parse for
monitoring tools.
    // The monitoring tools can then decide how to display in their UI the JSON's properties.
```

```
    return JSON.stringify(
      logObject,
      /**
       * Replaces Error instances with their stringified representation.
       */
      (_key: string, value: any) => {
        return value instanceof Error ? this.stringifyError(value) : value;
      }
    );
  }
```

5.11 Scelta del `LoggerService` lato server

core-libs/setup/ssr/logger/services/server-logger-service-factory.ts:L12-L17

```ts
export const serverLoggerServiceFactory = () => {
  const isExpress = inject(REQUEST, { optional: true }) !== null;
  return isExpress
    ? inject(ExpressLoggerService)
    : inject(PrerenderingLoggerService);
};
```

5.12 Middleware d'errore Express

core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32

```ts
export const defaultExpressErrorHandlers =
  (documentContent: string): ErrorRequestHandler =>
  (err, _req, res, _next) => {
    if (!res.headersSent) {
      res.set('Cache-Control', 'no-store');
      const statusCode =
        err instanceof CmsPageNotFoundOutboundHttpError
          ? HttpResponseStatus.NOT_FOUND
          : HttpResponseStatus.INTERNAL_SERVER_ERROR;
      res.status(statusCode).send(documentContent);
    }
  };
```

5.13 Validazione dell'origin

core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L51-L67

```ts
export function getOriginValidationMiddleware(
  options: OriginValidationOptions
): RequestHandler {
  const allowedOrigins = parseAllowedOrigins(options.allowedOrigins);
```

```
  if (!allowedOrigins.length) {
    return (_req, _res, next) => next();
  }
  return (req, res, next) => {
    const origin = getRequestOrigin(req);
    if (!isAllowedOrigin(origin, allowedOrigins)) {
      res.setHeader('Cache-Control', 'no-store');
      res.status(400).send('Bad Request');
      return;
    }
    next();
  };
}
```

5.14 Configurazione della demo

projects/storefrontapp/src/server.ts:L22-L29

```ts
const ssrOptions: SsrOptimizationOptions = {
  timeout: Number(
    process.env['SSR_TIMEOUT'] ?? defaultSsrOptimizationOptions.timeout
  ),
  cache: process.env['SSR_CACHE'] === 'true',
};

const ngExpressEngine = NgExpressEngineDecorator.get(engine, ssrOptions);
```

6. Configurazione

Chiavi di `SsrOptimizationOptions` (definite in `core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts`, default in core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L220-L237):

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
timeout	number (ms)	3_000	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L19-L22	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L187-L204
cache	boolean	false	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L24-L31	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L223-L226, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L291-L295
cacheSizeMemory	number (byte)	800_000_000	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L33-L49	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L14-L29

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cacheEntrySizeCalculator	CacheEntrySizeCalculator	new DefaultCacheEntrySizeCalculator()	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L51-L57	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L59-L61
concurrency	number	10	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L59-L62	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L177-L179
ttl	number | undefined (ms)	undefined	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L64-L68	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L43-L45, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L122-L127
renderKeyResolver	(req) => string	getDefaultRenderKey (= getRequestUrl)	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L70-L76	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L112-L116
renderingStrategyResolver	(req) => RenderingStrategy	defaultRenderingStrategyResolver(defaultRenderingStrategyResolverOptions)	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L78-L86	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L118-L122
forcedSsrTimeout	number (ms)	60_000 (fallback interno 60000)	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L88-L92	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L200-L203
maxRenderTime	number (ms)	300_000 (fallback interno 300000)	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L94-L108	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L417-L417
reuseCurrentRendering	boolean	true	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L110-L132	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L139-L139, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L171-L171, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L340-L340, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L410-L410
debug	boolean (@deprecated)	— (escluso dal tipo dei default)	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L134-L143	nessuna lettura
logger	ExpressServerLogger	new DefaultExpressServerLogger()	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L145-L156	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L76-L79, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L35-L54
shouldCacheRenderingResult	({options, entry}) => boolean	({ entry: { err } }) => !err	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L158-L170	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L54-L61
ssrFeatureToggles	{}	{}	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L172-L180	merge in core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L69-L72

Altre chiavi e variabili coinvolte:

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`RenderingStrategyResolverOptions.excludedUrls`	`string[]` (RegExp)	`['checkout', 'my-account', 'punchout', 'opf']` (+ fisso `cx-preview`)	`core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver-options.ts:L7-L16`, `core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L11-L13`	`core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L32-L39`
`RenderingStrategyResolverOptions.excludedParams`	`string[]`	`['asm']`	`core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver-options.ts:L7-L16`	`core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L15-L30`
`OriginValidationOptions.allowedOrigins`	`string[] | string`	assente → no-op	`core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L10-L12`	`core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L54-L57`
`optimizationOptions` di `NgExpressEngineDecorator.get`	`SsrOptimizationOptions | null`	`defaultSsrOptimizationOptions`; `null` = nessuna ottimizzazione	`core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L43-L57`	`core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L69-L72`
env `SSR_TIMEOUT` / `SSR_CACHE` (demo)	stringa	`3000` / `false`	`projects/storefrontapp/src/server.ts:L22-L27`	idem
env `SSR_ALLOWED_ORIGINS` (demo)	CSV	non impostata → no-op	`projects/storefrontapp/src/server.ts:L45-L49`	idem
env **`NG_ALLOWED_HOSTS`** (`@angular/ssr`)	CSV	non impostata → opzione `allowedHosts` o insieme vuoto	`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L10-L29`	`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L112`; script in `package.json:L152-L155`
env `SERVER_REQUEST_ORIGIN` (demo)	stringa	—	`projects/storefrontapp/src/app/app.module.server.ts:L12-L14`	`core-libs/setup/ssr/providers/server-request-origin.ts:L22-L24` (vedi cap. 07)

Nella pipeline npm: gli script `serve:ssr*` impostano `NG_ALLOWED_HOSTS=localhost`; `serve:ssr:ci` usa `SSR_TIMEOUT=0`, i `:dev`/`:watch` `SSR_TIMEOUT=30000` (package.json:L152-L155). I test SSR E2E avviano il server con `SSR_CACHE` e `SSR_TIMEOUT` (default 20000) tramite `npm run serve:ssr` (projects/ssr-tests/src/utils/ssr.utils.ts:L17-L17, projects/ssr-tests/src/utils/ssr.utils.ts:L47-L56).

---

7. Commenti e note notevoli dal sorgente

- "If the `cache` is set to `false`, the response is evicted as soon as the first successful response is successfully returned." — core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L33-L36. Descrive il consumo singolo dell'entry (§3.2 passo 2).
- Il CSR è restituito con "Cache-Control: no-store" per impedire alle cache esterne di riusarlo — core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L98-L102.
- "Here we respond with the fallback to CSR, but we don't `return`." — core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L274-L277.
- Accesso alla Map al momento del completamento "because in the meantime something might have deleted the value (i.e. when `maxRenderTime` passed)" — core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L364-L365.
- "There is no way to abort the running render of Angular Universal ... (note: it's a potential memory leak!)" — core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L383-L387.
- **IMPORTANT**: il server deve avere molta più memoria di `cacheSizeMemory`, perché serve anche al rendering — core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L41-L45.
- `maxRenderTime` "should always be higher than `timeout` and `forcedSsrTimeout`" — core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L104-L104.
- `debug` è `@deprecated` dalla 2211.27: ora tutto viene loggato incondizionatamente — core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L134-L143.
- `ssrFeatureToggles` "are temporary and will be removed in the future" e sono distinti dai feature toggle dell'app — core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L173-L178.
- **CAUTION** sulla stima della dimensione degli errori: "It's not recommended to cache error objects." — core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L20-L22.
- 2 byte per carattere "verified through heap snapshots and memory measurements" — core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L66-L70.
- "This should never happen. MAX_SAFE_INTEGER is 7.9 Petabytes" — core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L84-L85.
- Il ramo `oldestKey === undefined` "should never happen" — core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L94-L98.
- Commento lungo in `setAsRendering`/`store` sul perché passare da `clear()` e non da `renders.delete()` (per non "orfanare" `_size`) — core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L26-L29, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L47-L51.
- `PROPAGATE_ERROR_TO_SERVER`: "We need it until Angular implements a proper propagation of async errors" — core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L16-L18.
- `breakLength: Infinity` per evitare che Kibana interpreti righe separate come log diversi — core-libs/setup/ssr/logger/get-logger-inspect-options.ts:L34-L40.

- "X-Forwarded-Host is normally only ever a single value, but this is to be safe." — core-libs/setup/ssr/express-utils/express-request-origin.ts:L16-L17.
- `TestConfigServerModule`: "CAUTION: DON'T USE IT IN PRODUCTION! IT HASN'T BEEN REVIEWED FOR SECURITY ISSUES." — core-libs/setup/ssr/testing/test-config-server.module.ts:L34-L34; nella demo "DO NOT USE IN CUSTOMERS APPS" — projects/storefrontapp/src/app/app.config.server.ts:L23-L26. Il modulo legge un cookie dalla `REQUEST` e lo parsa come JSON per fornire `TEST_CONFIG` (core-libs/setup/ssr/testing/test-config-server.module.ts:L11-L58), e `testing/index.ts` esporta solo il modulo (core-libs/setup/ssr/testing/index.ts:L7-L10).

8. Extension point

EXTENSION POINT	COME	DEFINITO	EXPORT
Attivare/disattivare l'ottimizzazione	`NgExpressEngineDecorator.get(engine, options);` `null` la disattiva	core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L36-L57	core-libs/setup/ssr/public_api.ts:L7-L7
Tarare timeout, cache, concorrenza, TTL, limiti	chiavi di `SsrOptimizationOptions` (§6)	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L18-L181	core-libs/setup/ssr/optimized-engine/index.ts:L12-L12
Chiave di cache custom (es. ignorare parametri di tracking)	`renderKeyResolver: (req) => string`	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L70-L76	idem
Strategia per richiesta (forzare SSR/CSR)	`renderingStrategyResolver`, eventualmente componendo `defaultRenderingStrategyResolver` con opzioni proprie	core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L62-L72, core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver-options.ts:L12-L16	core-libs/setup/ssr/optimized-engine/index.ts:L9-L10
Cosa mettere in cache	`shouldCacheRenderingResult({ options, entry })`	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L158-L170	core-libs/setup/ssr/optimized-engine/index.ts:L12-L12
Stima della dimensione	`cacheEntrySizeCalculator` (implementare `CacheEntrySizeCalculator` o estendere `DefaultCacheEntrySizeCalculator`, metodi `protected` `calculateHtmlSize`, `calculateErrorSize`, `getStringSize`)	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.model.ts:L26-L28, core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L24-L75	core-libs/setup/ssr/optimized-engine/rendering-cache/index.ts:L7-L10
Logger del server	`logger`: implementare `ExpressServerLogger` o estendere `DefaultExpressServerLogger` (metodi `protected` `stringifyWithContext`, `mapContext`, `mapRequest`, `stringifyError`)	core-libs/setup/ssr/logger/loggers/express-server-logger.ts:L14-L48, core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L28-L133	core-libs/setup/ssr/logger/loggers/index.ts:L7-L8

EXTENSION POINT	COME	DEFINITO	EXPORT
Formattazione dei messaggi dell'app	estendere `ExpressLoggerService` (`protected formatLogMessage`)	core-libs/setup/ssr/logger/services/express-logger.service.ts:L55-L61	core-libs/setup/ssr/logger/services/index.ts:L7-L9
Leggere il contesto di richiesta (uuid, trace)	`getRequestContext(request)`	core-libs/setup/ssr/optimized-engine/request-context.ts:L28-L30	core-libs/setup/ssr/optimized-engine/index.ts:L11-L11
Comportamento dell'engine	sottoclasse di `OptimizedSsrEngine` sovrascrivendo i metodi `protected` (`fallbackToCsr`, `getRenderingKey`, `getRenderingStrategy`, `shouldRender`, `shouldTimeout`, `getTimeout`, `returnCachedRender`, `renderResponse`, `getDocument`, `logOptions`); `handleRender`, `startRender`, `isConcurrencyLimitExceeded` e `logForRenderResult` (più i campi `templateCache` e `renderCallbacks`) sono `private`	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L84-L320, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L41-L41, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L53-L53, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L457-L457, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L166-L166, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L329-L329, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L389-L389	core-libs/setup/ssr/optimized-engine/index.ts:L7-L7
Status/documento per gli errori	sostituire o affiancare `defaultExpressErrorHandlers(document)` con un proprio `ErrorRequestHandler`	core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32	core-libs/setup/ssr/error-handling/index.ts:L8-L8
Propagare errori custom al server	`inject(PROPAGATE_ERROR_TO_SERVER, { optional: true })` o un `MULTI_ERROR_HANDLER` proprio	core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L20-L22	core-libs/setup/ssr/error-handling/index.ts:L7-L9
Allowlist di origin	`getOriginValidationMiddleware({ allowedOrigins })`	core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L10-L67	core-libs/setup/ssr/public_api.ts:L8-L8

Esempio (illustrativo, non presente nel repo; usa solo simboli esportati da `@spartacus/setup/ssr` secondo core-libs/setup/ssr/optimized-engine/index.ts:L7-L12 e core-libs/setup/ssr/public_api.ts:L7-L16): forzare l'SSR per una sezione, escludere `/cart` dall'SSR mantenendo i default, e ignorare un parametro di tracking nella chiave.

```ts
// server.ts — esempio illustrativo
import {
  NgExpressEngineDecorator,
```

```
  RenderingStrategy,
  SsrOptimizationOptions,
  defaultRenderingStrategyResolver,
  defaultRenderingStrategyResolverOptions,
  getDefaultRenderKey,
  ngExpressEngine as engine,
} from '@spartacus/setup/ssr';

const baseResolver = defaultRenderingStrategyResolver({
  ...defaultRenderingStrategyResolverOptions,
  excludedUrls: [
    ...(defaultRenderingStrategyResolverOptions.excludedUrls ?? []),
    '/cart', // NB: RegExp/sottostringa su request.url (§9)
  ],
});

const ssrOptions: SsrOptimizationOptions = {
  timeout: 3000,
  cache: true,
  ttl: 10 * 60_000,
  renderingStrategyResolver: (req) =>
    req.url.startsWith('/campaign/')
      ? RenderingStrategy.ALWAYS_SSR // ignora concurrency (§9)
      : baseResolver(req),
  renderKeyResolver: (req) => {
    // chiave di default = origin + originalUrl; rimuove i parametri utm_*
    const url = new URL(getDefaultRenderKey(req));
    [...url.searchParams.keys()]
      .filter((k) => k.startsWith('utm_'))
      .forEach((k) => url.searchParams.delete(k));
    return url.toString();
  },
};

const ngExpressEngine = NgExpressEngineDecorator.get(engine, ssrOptions);
```

Ricette pertinenti in Parte III:

- override di configurazione e provider via DI (es. registrare un `LoggerService`/`MULTI_ERROR_HANDLER` custom solo lato server): 50 — Ricette: configurazione, DI, interceptor, eventi;
- cosa è API pubblica in `@spartacus/setup/ssr` e cosa no (`decorateExpressEngine`, `getRequestOrigin`, `getRequestUrl`, `parseTraceparent`, `preprocessRequestForLogger` non sono esportati; i metodi `private` dell'engine non sono un contratto): 54 — API pubblica vs interna, upgrade, anti-pattern, checklist.

Simboli non esportati dal public_api: `decorateExpressEngine` e i tipi `NgExpressEngine`/`NgExpressEngineInstance` (core-libs/setup/ssr/engine-decorator/index.ts:L7-L7), `getLoggableSsrOptimizationOptions` e `preprocessRequestForLogger` (core-libs/setup/ssr/optimized-engine/index.ts:L7-L12), `getRequestOrigin`/`getRequestUrl` (core-libs/setup/ssr/public_api.ts:L7-L16; `getDefaultRenderKey` invece sì), `parseTraceparent`, `W3cTraceContext` e gli errori traceparent (nessun barrel riesporta `w3c-trace-context/*`: core-libs/setup/ssr/logger/loggers/index.ts:L7-L8).

9. Trappole, casi limite, comportamenti controintuitivi

1. `timeout: 0` **= CSR immediato + render in background.** Con strategia `DEFAULT` e `timeout` falsy `shouldTimeout` è falso: ogni prima richiesta riceve CSR e l'SSR serve la successiva (anche con `cache: false`, una volta). È la modalità di `serve:ssr:ci` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L187-L192, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L273-L279, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L296-L299, package.json:L154-L154).
2. `cache: false` **non significa "nessuna memoria".** I render terminati dopo il fallback sono salvati comunque (soggetti a `cacheSizeMemory` e `shouldCacheRenderingResult`) finché una richiesta non li consuma; se nessuno richiede più quell'URL restano in memoria (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L296-L299, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L223-L226, core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L24-L31).
3. `ALWAYS_SSR` **ignora concorrenza e render in corso.** Può superare `concurrency` e, senza `reuseCurrentRendering`, avviare un secondo render per la stessa chiave: lo spec lo verifica (`currentConcurrency: 2`) (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L153-L158, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.spec.ts:L681-L713). Il resolver di default non restituisce mai `ALWAYS_SSR`: serve un resolver custom (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L62-L72).
4. `excludedUrls` **sono RegExp/sottostringhe su** `request.url`**.** `'checkout'` esclude qualunque URL che contenga "checkout" (anche un prodotto con quello slug); caratteri speciali regex cambiano il significato (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L36-L38).
5. `cx-preview` **è sempre CSR** (SmartEdit), non disattivabile tramite le opzioni del resolver di default (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L11-L13, core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L65-L67).
6. **Chiave = origin +** `originalUrl`**.** Varianti di query string o host diversi creano entry distinte; Host header manipolati possono avvelenare la cache, motivo di `getOriginValidationMiddleware` (core-libs/setup/ssr/express-utils/express-request-url.ts:L10-L12, core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L14-L17).
7. **Merge shallow delle opzioni.** Passare `logger: undefined` esplicito fa lanciare il costruttore; `renderingStrategyResolver: undefined` riporta la strategia a `DEFAULT`, quindi checkout/my-account vengono renderizzati in SSR (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L63-L78, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L118-L122).
8. **Opzione** `null` **e log di avvio.** `typeof null === 'object'`, e `isClassInstance` accede a `value.constructor`: un'opzione impostata a `null` farebbe fallire il costruttore in `logOptions` con un `TypeError`. Analisi statica (core-libs/setup/ssr/optimized-engine/get-loggable-ssr-optimization-options.ts:L16-L29, core-libs/setup/ssr/optimized-engine/get-loggable-ssr-optimization-options.ts:L37-L39).
9. `get(engine)` ≠ `get(engine, null)`. Senza secondo argomento l'ottimizzazione è attiva con i default; solo `null` la disattiva (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L57, core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L69-L72).
10. **Senza** `OptimizedSsrEngine` **manca** `EXPRESS_SERVER_LOGGER`**.** L'unico provider è in `startRender`; con `null`, `ExpressLoggerService` (scelto perché `REQUEST` esiste) fa un `inject(EXPRESS_SERVER_LOGGER)` non opzionale → probabile `NullInjectorError` alla prima risoluzione di `LoggerService`. DA VERIFICARE a runtime (manca `node_modules`) (core-libs/setup/ssr/logger/services/express-logger.service.ts:L26-L27, core-libs/setup/ssr/logger/services/server-logger-service-factory.ts:L12-L17, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L426-L429).
11. `allowedHosts` **non configurato → finto SSR, potenzialmente in cache.** Senza **NG_ALLOWED_HOSTS** né `allowedHosts`, `CommonEngine.render` restituisce il documento grezzo come successo; per

`OptimizedSsrEngine` è HTML valido e, con `cache: true`, viene salvato e servito con 200 senza `no-store`. La demo non passa `allowedHosts` e conta sulla variabile negli script npm (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L117-L135`, `projects/storefrontapp/src/server.ts:L51-L56`, `package.json:L152-L155`). `DA VERIFICARE` il caso di URL non parsabile, per cui la validazione viene saltata (`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L117-L117`).

12. **Origin incoerente.** `ngExpressEngine` costruisce l'URL con `req.get('host')`, mentre la chiave di cache e `SERVER_REQUEST_ORIGIN` usano `getRequestOrigin`, che può preferire `X-Forwarded-Host` (`core-libs/setup/ssr/engine/ng-express-engine.ts:L104-L106`, `core-libs/setup/ssr/express-utils/express-request-origin.ts:L12-L25`).
13. `reuseCurrentRendering` **+** `cache: false`**:** `store`**/**`clear` **multipli sulla stessa chiave.** Le callback condivise vengono eseguite in sequenza: una richiesta già servita in CSR fa `store`, una ancora in attesa fa `clear`; l'esito finale in cache dipende dall'ordine d'arrivo. Analisi statica (`core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L282-L300`, `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L366-L368`).
14. **L'eviction FIFO può rimuovere un placeholder** `rendering`**.** `clearOldestEntriesForSize` non distingue i placeholder `{ rendering: true }` (dimensione 0): rimuoverne uno non libera spazio, ma rende falso `isRendering` per un render ancora in corso (possibile render duplicato o fallback diverso). Analisi statica; `DA VERIFICARE` con un test, non coperto dagli spec letti (`core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L25-L32`, `core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L88-L103`).
15. **Nessun LRU, TTL pigro.** `get` non riordina; le entry scadute non vengono rimosse finché non sono sovrascritte o sfrattate e continuano a occupare `usedSize` (`core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L105-L107`, `core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L116-L127`).
16. **Errori non cachati, neppure "one-shot".** Con il default `!err` un render tardivo in errore non viene conservato: la richiesta successiva rifà il render (`core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L234-L234`, `core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L54-L61`).
17. **Opzioni di cache invalide solo loggate.** `cacheSizeMemory` ≤ 0 o calculator mancante producono un log d'errore; con limite 0 nessuna entry con dimensione > 0 viene mai salvata, neppure il one-shot di §3.2 (`core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L35-L54`, `core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L66-L76`).
18. **Errori sincroni non-**`Error` **inghiottiti.** In `ngExpressEngine` un `throw` di un valore che non è `Error` non chiama il callback: il render resta "in corso" fino a `maxRenderTime` (`core-libs/setup/ssr/engine/ng-express-engine.ts:L124-L126`).
19. **Concorrenza per-processo e senza abort.** `currentConcurrency` è un contatore in memoria per istanza; dopo `maxRenderTime` lo slot è liberato ma il render continua a consumare CPU/memoria (`core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L37-L41`, `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L405-L417`).
20. `maxRenderTime` **<** `timeout`**.** Vietato dal JSDoc: il render verrebbe dichiarato "hanging" prima del timeout e il suo esito ignorato, così il client riceve CSR solo allo scadere del proprio timeout (`core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L104-L104`, `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L435-L442`).
21. **Log "waiting" fuorviante.** Con reuse attivo anche la richiesta che avvia il render logga "Request is waiting...", e i log "Rendering started/completed" portano il contesto della sola prima richiesta (`core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L374-L377`, `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L419-L419`).

22. `traceparent` **malformato → un log** `error` **per richiesta**, senza bloccare il render (core-libs/setup/ssr/optimized-engine/request-context.ts:L94-L102).
23. **Template letto una sola volta.** `getDocument` usa `readFileSync` al primo accesso e poi la `templateCache`: nessuna invalidazione se il file cambia a caldo (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L311-L320).
24. `defaultExpressErrorHandlers` **non chiama** `next` **e non logga**: se gli header sono già inviati l'errore sparisce (core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L23-L31).
25. **Wildcard di origin.** `*` non copre l'apex né più livelli: `https://*.my.domain.com` non ammette `a.b.my.domain.com` (core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L32-L36, core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L89-L95).
26. **Formato dei log deciso da** `ngDevMode` **del processo.** Nel bundle server ottimizzato i log sono JSON a riga singola. DA VERIFICARE che il build server definisca `ngDevMode` falso: dipende dalla configurazione del builder, non letta (core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L69-L73, angular:packages/core/src/util/is_dev_mode.ts:L20-L22).
27. **Istanze di default condivise.** `DefaultCacheEntrySizeCalculator` e `DefaultExpressServerLogger` dei default sono create una volta al caricamento del modulo e condivise da tutti gli engine che usano i default (core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L223-L233).

---

10. Come fare debug

- **Leggere il log di avvio.** "[spartacus] SSR optimization engine initialized" elenca tutte le opzioni effettive dopo il merge, con resolver e funzioni stampati come sorgente (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L84-L96, core-libs/setup/ssr/optimized-engine/get-loggable-ssr-optimization-options.ts:L16-L29). È il primo posto dove verificare un override "perso" (§9 trappola 7).
- **Seguire una richiesta per UUID.** Ogni log del server emesso con una `request` nel contesto (tutti quelli legati a una richiesta) porta `context.request.uuid`, `timeReceived`, `url` ed eventuale `traceContext` (core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L98-L107, core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L117-L122); il log di avvio e gli errori di `RenderingCacheSizeManager` non hanno contesto di richiesta (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L93-L95, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L40-L43, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L49-L52, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L87-L90, core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L103-L106). Messaggi tipici: "Render from cache", "CSR fallback: rendering in progress", "CSR fallback: Concurrency limit exceeded", "SSR rendering exceeded timeout", "Rendering started/completed", "Request is resolved with the SSR rendering result/error", "... This might cause memory leaks!", "... completed after the specified maxRenderTime, therefore it was ignored." (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L142-L150, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L251-L253, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L268-L271, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L413-L416, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L419-L419, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L437-L440, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L445-L447, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L462-L473).
- **Correlare con il tracing distribuito.** Inviare un header `traceparent` valido (55 caratteri) e cercare `traceId` nei log (core-libs/setup/ssr/logger/loggers/w3c-trace-context/parse-traceparent.ts:L11-L47).

- **Riconoscere un CSR.** La risposta CSR (fallback o errore) ha `Cache-Control: no-store` (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L108-L108, core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L25-L25). Una pagina "SSR" senza contenuto, con 200 e senza `no-store`, fa pensare a §9 trappola 11 (`allowedHosts`): cercare nell'output "Falling back to client side rendering" (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L117-L135).
- **Isolare l'ottimizzazione.** Per escludere effetti di cache/timeout: `SSR_TIMEOUT` alto (gli script `:dev` usano 30000) e `SSR_CACHE` non impostato (package.json:L152-L155, projects/storefrontapp/src/server.ts:L22-L27); per vedere sempre il render "fresco", `timeout: 0` mostra CSR + SSR alla seconda richiesta (§3.3).
- **Log leggibili in locale.** In dev mode `DefaultExpressServerLogger` usa `formatWithOptions` multi-linea (core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L69-L73).
- **Test di riferimento.** Gli scenari di `OptimizedSsrEngine` sono coperti da `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.spec.ts` (timeout, cache, concurrency, ttl, strategie, `maxRenderTime`, reuse: core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.spec.ts:L267-L1413); gli scenari end-to-end di status HTTP da projects/ssr-tests/src/ssr-testing.spec.ts:L19-L130. Per lanciare questi ultimi usare la skill `ssr` del progetto.

---

11. Domande di autoverifica

1. **Con le opzioni di default, cosa riceve il client se il render dura 5 secondi?** Il fallback CSR allo scadere di `timeout` (3000 ms), con `Cache-Control: no-store`; il render continua e il suo HTML viene salvato per la richiesta successiva alla stessa chiave, che lo consuma una sola volta perché `cache` è `false` (core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L220-L229, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L261-L272, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L296-L299, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L223-L226).
2. **Che differenza c'è tra** `NgExpressEngineDecorator.get(engine)` **e** `NgExpressEngineDecorator.get(engine, null)`**?** Nel primo caso `optimizationOptions` è `undefined`, scatta il default `defaultSsrOptimizationOptions` e si usa `OptimizedSsrEngine`; nel secondo l'istanza di `ngExpressEngine` è restituita così com'è (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L57, core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L69-L72).
3. **Perché una seconda richiesta per lo stesso URL, durante un render in corso, non ottiene un CSR immediato con i default?** Perché `reuseCurrentRendering` è `true`: `fallBack` è falso, `isConcurrencyLimitExceeded` restituisce `false` senza occupare uno slot e la callback viene accodata al render esistente, con il proprio timeout (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L137-L139, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L170-L175, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L350-L372).
4. **Come si ottiene che una pagina venga sempre servita in SSR, anche oltre il limite di concorrenza?** Con un `renderingStrategyResolver` custom che restituisce `RenderingStrategy.ALWAYS_SSR`: `shouldRender` è sempre vero e il timeout usato è `forcedSsrTimeout` (default 60 s). Il resolver di default non lo fa mai (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L153-L158, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L200-L204, core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L62-L72).
5. **Un URL di prodotto contiene la parola** `checkout` **nello slug: che succede?** Con il resolver di default viene servito in CSR, perché `excludedUrls` è valutato con `request.url.search(url) > -1` (core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver.ts:L32-L39, core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver-options.ts:L12-L16).

6. **Come arriva un 404 CMS al browser come status HTTP 404?** L'interceptor crea `CmsPageNotFoundOutboundHttpError`, `CxErrorHandler` lo passa a `PropagatingToServerErrorHandler`, che chiama `PROPAGATE_ERROR_TO_SERVER`; `CxCommonEngine` rigetta a fine render, l'errore passa da `OptimizedSsrEngine` a Express e `defaultExpressErrorHandlers` risponde 404 con il documento CSR (core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L48-L98, core-libs/setup/ssr/providers/ssr-providers.ts:L41-L45, core-libs/setup/ssr/engine/cx-common-engine.ts:L41-L66, core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32).
7. **Perché la cache può sfrattare una pagina molto visitata prima di una mai riletta?** Perché l'eviction rimuove la prima chiave in ordine di inserimento e `get` non riordina la Map: è FIFO, non LRU (core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L88-L107).
8. **Quanta memoria "costa" alla cache una pagina HTML di 500.000 caratteri?** 1.000.000 di byte secondo `DefaultCacheEntrySizeCalculator`, che stima 2 byte per carattere (core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L62-L74).
9. **Da dove viene il logger che `ExpressLoggerService` usa dentro l'app Angular?** Dal provider `EXPRESS_SERVER_LOGGER` che `startRender` antepone ai provider di ogni render, con il `logger` delle opzioni (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L423-L432, core-libs/setup/ssr/logger/services/express-logger.service.ts:L24-L27).
10. **Cosa succede a un render che non termina mai?** Dopo `maxRenderTime` (default 5 minuti) la chiave viene liberata, lo slot di concorrenza restituito, le callback in attesa dimenticate e viene loggato un possibile memory leak; se il render finisce più tardi, il suo esito è ignorato. Il render non viene abortito (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L380-L388, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L405-L417, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L434-L442).

TAVOLA 39.1 **4.1 Decisioni di renderResponse**

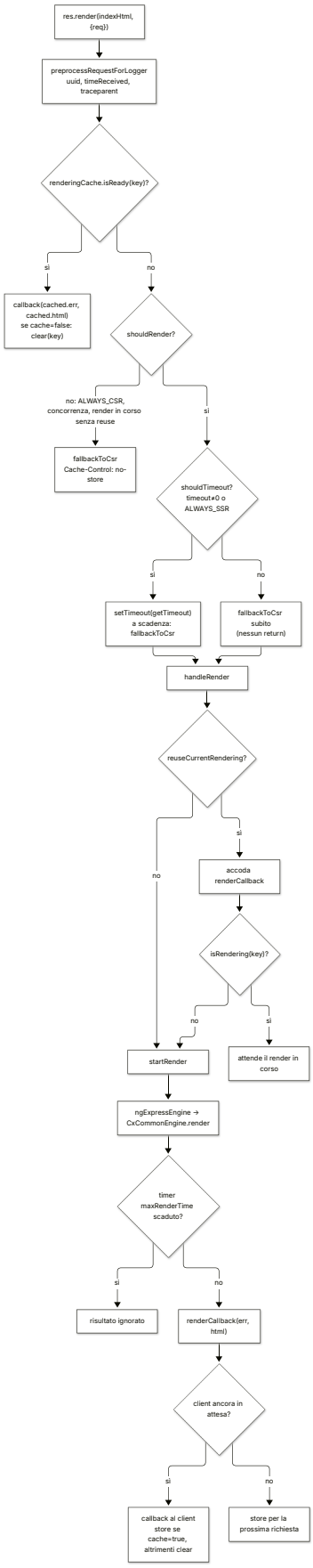


TAVOLA 39.2 **4.3 Errore applicativo → risposta HTTP**

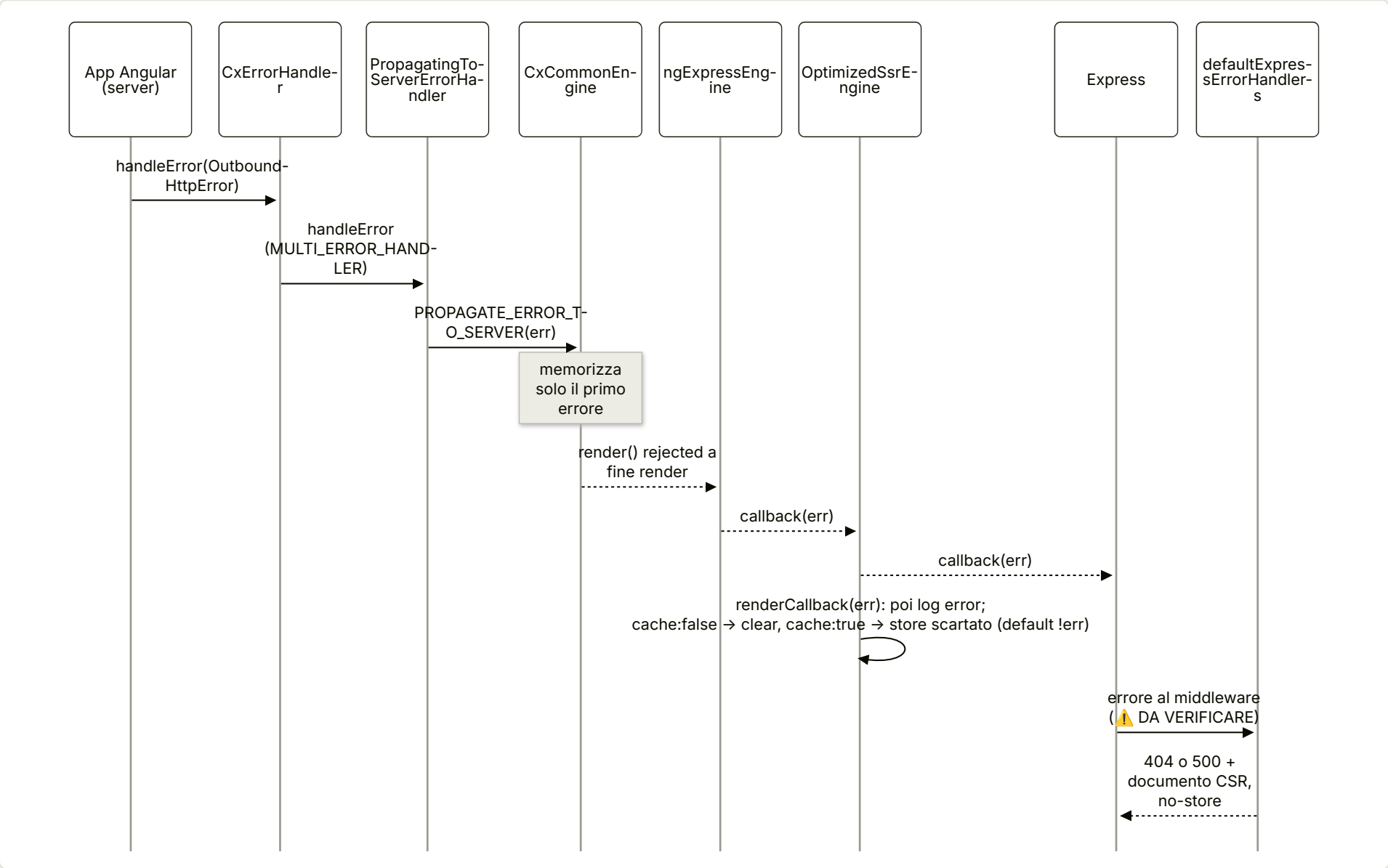
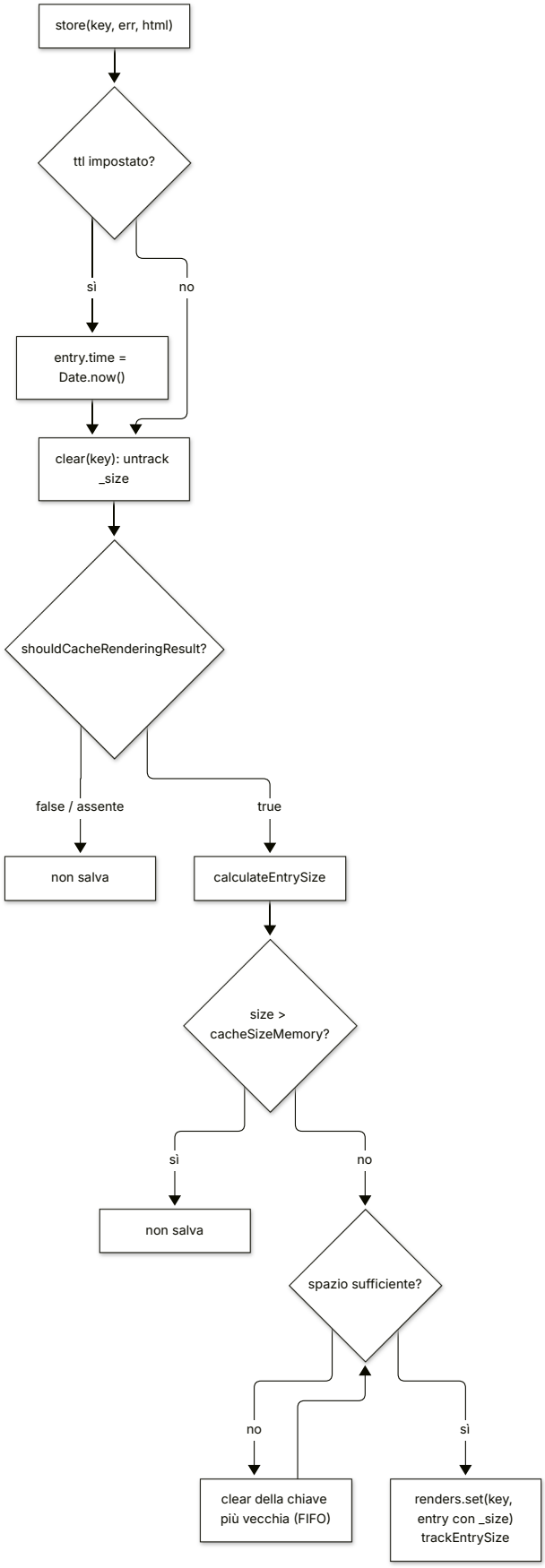


TAVOLA 39.3 **4.4 Inserimento in RenderingCache**



40

SEO e PWA

Title, meta, canonical e JSON-LD anche in SSR. Supporto PWA.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Una storefront deve farsi trovare dai motori di ricerca e condividere bene sui social. Per farlo ogni pagina deve avere `<title>`, `<meta name="description">`, `<meta name="robots">`, un `<link rel="canonical">` e possibilmente dati strutturati JSON-LD. In una Single Page Application questi tag cambiano a ogni navigazione e, perché un crawler li veda, vanno scritti anche nell'HTML prodotto in SSR. Spartacus divide il problema in tre livelli, in due librerie diverse.

LIVELLO	LIBRERIA	CLASSI CHIAVE	DOMANDA A CUI RISPONDE
Raccolta dei meta (dati)	`@spartacus/core`	`PageMetaService`, `PageMetaResolver`, `BasePageMetaResolver`, `PageLinkService`	"Qual è il title, la description, il canonical… della pagina corrente?"
Scrittura nel DOM (rendering)	`@spartacus/storefront`	`SeoModule`, `SeoMetaService`, `PageMetaLinkService`	"Come trasformo un `PageMeta` in tag dentro `<head>`?"
Dati strutturati (JSON-LD)	`@spartacus/storefront`	`StructuredDataFactory`, `SCHEMA_BUILDER`, `JsonLdScriptFactory`	"Quale schema.org scrivo nello `<script type="application/ld+json">`?"

Il cuore è il modello `PageMeta`: un oggetto con le proprietà opzionali `title`, `heading`, `description`, `robots`, `image`, `breadcrumbs` e `canonicalUrl` (core-libs/core/src/cms/model/page.model.ts:L40-L78). `PageMetaService` lo calcola per la pagina CMS corrente, scegliendo tra più `PageMetaResolver` quello con lo score migliore per tipo, template e uid della pagina (core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L45). Il resolver scelto espone metodi `resolve*` che restituiscono `Observable`, e una configurazione decide quale metodo alimenta quale proprietà (core-libs/core/src/cms/page/config/default-page-meta.config.ts:L9-L40).

`PageMeta` non serve solo alla SEO. Lo usano anche la UI, per l'heading `<h1>` (core-libs/storefront/cms-components/navigation/page-header/page-title.utils.ts:L11-L18) e per il breadcrumb (core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.component.ts:L59-L63), e il builder JSON-LD del breadcrumb (core-libs/storefront/cms-structure/seo/structured-data/builders/breadcrumb/breadcrumb-schema.builder.ts:L19-L23). Per questo l'interfaccia distingue `heading` (UI) da `title` (browser e crawler) (core-libs/core/src/cms/page/page.resolvers.ts:L10-L25).

La seconda metà del capitolo riguarda la **PWA**. `PwaModule` registra il service worker di Angular (`ngsw-worker.js`), ma lo abilita solo fuori da dev mode e con `pwa.enabled` (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L19-L23). Inoltre gestisce il prompt "Add to Home Screen" tramite l'evento browser `beforeinstallprompt` (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L36-L57). Entrambe le funzioni sono disattivate per default (core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L21-L26).

`SeoModule` e `PwaModule` vengono importati da `BaseStorefrontModule` (core-libs/storefront/base-storefront.module.ts:L26-L49), mentre `PageMetaModule.forRoot()` arriva tramite `CmsModule` (core-libs/core/src/cms/cms.module.ts:L14-L16), che è importato da `BaseCoreModule` (core-libs/core/src/base-core.module.ts:L34-L34). I resolver di prodotto, categoria e ricerca si aggiungono solo con `ProductModule.forRoot()` (core-libs/core/src/product/product.module.ts:L16-L49).

Capitoli collegati: il caricamento della pagina CMS che fa partire tutto è in 15 — CMS core; il rendering SSR, dove questi tag hanno davvero valore, è in 39 — SSR ottimizzato e 07 — SSR, hydration e TransferState; il meccanismo `UnifiedInjector` per i provider multi nei moduli lazy è in 12 — Feature module e lazy loading; i resolver di prodotto sono descritti anche in 24 — Prodotto e ricerca.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/cms/model/page.model.ts	PageMeta, BreadcrumbMeta, enum PageRobotsMeta
core-libs/core/src/cms/page/page.resolvers.ts	interfacce PageHeadingResolver, PageTitleResolver, PageDescriptionResolver, PageBreadcrumbResolver, PageImageResolver, PageRobotsResolver, CanonicalPageResolver
core-libs/core/src/cms/page/page-meta.resolver.ts	classe astratta PageMetaResolver (Applicable, scoring)
core-libs/core/src/cms/page/base-page-meta.resolver.ts	BasePageMetaResolver: implementazione riusabile via delega
core-libs/core/src/cms/page/content-page-meta.resolver.ts	ContentPageMetaResolver (CONTENT_PAGE)
core-libs/core/src/cms/page/page-meta.module.ts	PageMetaModule: multi-provide di ContentPageMetaResolver + default config
core-libs/core/src/cms/page/config/page-meta.config.ts	PageMetaConfig, PageMetaResolversConfig, CanonicalUrlOptions
core-libs/core/src/cms/page/config/default-page-meta.config.ts	defaultPageMetaConfig (7 resolver + opzioni canonical)
core-libs/core/src/cms/page/routing/page-link.service.ts	PageLinkService: calcolo del canonical URL
core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts	RoutingPageMetaResolver: breadcrumb dalle route Angular
core-libs/core/src/cms/page/routing/default-route-page-meta.resolver.ts	DefaultRoutePageMetaResolver: breadcrumb da cxPageMeta
core-libs/core/src/cms/page/routing/route-page-meta.model.ts	RoutePageMetaConfig, RouteBreadcrumbConfig
core-libs/core/src/cms/facade/page-meta.service.ts	PageMetaService: scelta del resolver e composizione del PageMeta
core-libs/core/src/util/applicable.ts	Applicable, resolveApplicable (selezione per priorità)
core-libs/core/src/util/rxjs/unite-latest.ts	uniteLatest: combineLatest "smussato"
core-libs/core/src/lazy-loading/unified-injector.ts	UnifiedInjector.getMulti: resolver da root + moduli lazy
core-libs/core/src/product/product.module.ts	registra i resolver PDP, PLP, ricerca, coupon
core-libs/core/src/product/services/product-page-meta.resolver.ts	ProductPageMetaResolver (PDP)
core-libs/core/src/product/services/category-page-meta.resolver.ts	CategoryPageMetaResolver (PLP)
core-libs/core/src/product/services/search-page-meta.resolver.ts	SearchPageMetaResolver
core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts	CouponSearchPageResolver (non esportato)

PATH	RUOLO
core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts	normalizeRobots: robotTag OCC → PageRobotsMeta[]
core-libs/storefront/cms-structure/seo/seo.module.ts	SeoModule, initSeoService
core-libs/storefront/cms-structure/seo/seo-meta.service.ts	SeoMetaService: scrive title, meta e canonical
core-libs/storefront/cms-structure/seo/page-meta-link.service.ts	PageMetaLinkService: <link rel="canonical"> e <link rel="preconnect">
core-libs/storefront/cms-structure/seo/media-preconnect.service.ts	MediaPreconnectService: preconnect verso il dominio dei media
core-libs/storefront/cms-structure/seo/html-lang-provider.ts	htmlLangProvider: <html lang> (non esportato)
core-libs/storefront/cms-structure/seo/config/seo.config.ts	SeoConfig
core-libs/storefront/cms-structure/seo/config/default-seo.config.ts	defaultSeoConfig
core-libs/storefront/cms-structure/seo/structured-data/structured-data.module.ts	StructuredDataModule, getStructuredDataFactory
core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts	StructuredDataFactory: orchestrazione degli SCHEMA_BUILDER
core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts	JsonLdScriptFactory: <script id="json-ld"> ed escape
core-libs/storefront/cms-structure/seo/structured-data/json-ld.directive.ts	JsonLdDirective ([cxJsonLd])
core-libs/storefront/cms-structure/seo/structured-data/builders/tokens.ts	SCHEMA_BUILDER, JSONLD_PRODUCT_BUILDER
core-libs/storefront/cms-structure/seo/structured-data/builders/schema.interface.ts	SchemaBuilder, JsonLdBuilder<T>
core-libs/storefront/cms-structure/seo/structured-data/builders/json-ld-builder.module.ts	JsonLdBuilderModule (opt-in)
core-libs/storefront/cms-structure/seo/structured-data/builders/product/product-schema.builder.ts	ProductSchemaBuilder
core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-base-product.builder.ts	sku, name, description, brand, image
core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-offer.builder.ts	offers
core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-review.builder.ts	aggregateRating, review

PATH	RUOLO
core-libs/storefront/cms-structure/seo/structured-data/builders/breadcrumb/breadcrumb-schema.builder.ts	BreadcrumbList
core-libs/storefront/cms-structure/pwa/pwa.module.ts	PwaModule, pwaConfigurationFactory, pwaFactory
core-libs/storefront/cms-structure/pwa/pwa.module-config.ts	PWAModuleConfig, defaultPWAModuleConfig
core-libs/storefront/cms-structure/pwa/ngsw-config.json	configurazione ngsw di riferimento (non usata dalla build della demo)
core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts	AddToHomeScreenService
core-libs/storefront/cms-structure/pwa/components/add-to-home-screen.component.ts	base astratta AddToHomeScreenComponent
core-libs/storefront/cms-structure/pwa/components/add-to-home-screen-banner/add-to-home-screen-banner.component.ts	cx-add-to-home-screen-banner
core-libs/storefront/cms-structure/pwa/components/add-to-home-screen-btn/add-to-home-screen-btn.component.ts	cx-add-to-home-screen-btn

3. Come funziona passo per passo

3.1 Dal backend: robots nella pagina CMS

1. La pagina CMS OCC porta un singolo `robotTag` (es. `NOINDEX_FOLLOW`). Il normalizer lo spezza in un array di due `PageRobotsMeta` e, se manca, assegna un array vuoto (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L198-L222).
2. Title e description arrivano direttamente come `page.title` e `page.description`, e `BasePageMetaResolver` li legge dalla pagina corrente (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L47-L61).

3.2 Registrazione dei resolver

1. `PageMetaModule` fa multi-provide del token `PageMetaResolver` con `useExisting: ContentPageMetaResolver`; `forRoot()` aggiunge `provideDefaultConfig(defaultPageMetaConfig)` (core-libs/core/src/cms/page/page-meta.module.ts:L13-L29).
2. `ProductModule.forRoot()` aggiunge, sullo stesso token, `ProductPageMetaResolver`, `CategoryPageMetaResolver`, `SearchPageMetaResolver` e `CouponSearchPageResolver` (core-libs/core/src/product/product.module.ts:L16-L49). Nella demo `ProductModule.forRoot()` è importato in `spartacus-features.module.ts` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L220-L220), e la schematic `add-spartacus` genera la stessa riga (core-libs/schematics/src/add-spartacus/spartacus-features.ts:L186-L186).
3. Le feature-libs aggiungono i loro resolver con lo stesso schema. Per esempio `CheckoutPageMetaResolver` usa `CONTENT_PAGE + MultiStepCheckoutSummaryPageTemplate` (feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts:L46-L47), mentre `CustomerTicketingPageMetaResolver` usa

`pageUid = 'support-ticket-details'` (feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts:L53-L54).

3.3 `PageMetaService`: raccolta dei resolver

1. `PageMetaService` è `providedIn: 'root'` e riceve `CmsService`, `UnifiedInjector`, `PageMetaConfig` e `PLATFORM_ID`; quest'ultimo è commentato come deprecato (core-libs/core/src/cms/facade/page-meta.service.ts:L29-L41).
2. `resolvers$ = unifiedInjector.getMulti(PageMetaResolver)` con `shareReplay({bufferSize:1, refCount:true})` (core-libs/core/src/cms/facade/page-meta.service.ts:L53-L57).
3. `getMulti` parte dall'injector root (`startWith(this.rootInjector)`) e aggiunge l'injector di ogni modulo lazy caricato (core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35). Per i moduli lazy legge solo le istanze proprie (`{ self: true }`, core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68) e accumula le liste con `scan`. Emette solo quando trova istanze nuove e lancia un errore se il token mescola provider multi e singoli (core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93). In pratica, quando un modulo lazy che fornisce un resolver (come quello del checkout) viene caricato, `resolvers$` riemette la lista allargata.

3.4 `PageMetaService`: scelta del resolver

1. `meta$` parte da `cms.getCurrentPage()`, scarta le pagine falsy, e con `switchMap` passa a `getMetaResolver(page)` (core-libs/core/src/cms/facade/page-meta.service.ts:L59-L68).
2. `getMetaResolver` applica `resolveApplicable(resolvers, [page], [page])` a ogni emissione di `resolvers$` (core-libs/core/src/cms/facade/page-meta.service.ts:L133-L139).
3. `resolveApplicable` tiene solo le istanze con `hasMatch(page)` vero e, tra queste, sceglie quella con `getPriority` più alta. Il confronto è `lastPriority > currPriority` (stretto), quindi **a parità vince il provider registrato per ultimo** (core-libs/core/src/util/applicable.ts:L63-L87), come dichiara anche il JSDoc (core-libs/core/src/util/applicable.ts:L47-L62).
4. In `PageMetaResolver`, `hasMatch` è `getScore(page) > 0` e `getPriority` è lo score stesso (core-libs/core/src/cms/page/page-meta.resolver.ts:L47-L53). Lo score somma ±1 per `pageType`, ±1 per `pageTemplate` e ±10 per `pageUid`; ogni criterio conta solo se la proprietà è valorizzata sul resolver (core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L45).

Esempi di scoring con i resolver standard (calcolati da core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L45 e core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L89-L91):

PAGINA	CONTENTPAGE… (CONTENT)	SEARCH… (CONTENT + TEMPLATE RICERCA)	COUPONSEARCH… (COME SEARCH ±1)	PRODUCT… (PRODUCT)	CATEGORY… (CATEGORY)	VINCE
contenuto generico, template qualsiasi	1	0	−1	−1	−1	Content
`SearchResultsListPageTemplate` senza couponcode	1	2	1	−1	−1	Search
`SearchResultsListPageTemplate` con couponcode	1	2	3	−1	−1	CouponSearch
PDP	−1	−2	−3	1	−1	Product
PLP / categoria	−1	−2	−3	−1	1	Category

Un resolver con `pageUid` corretto prende +10 e vince quasi sempre: per esempio `CustomerTicketingPageMetaResolver` arriva a 11 sulla pagina `support-ticket-details` (feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts:L51-L55). Se nessun resolver ha score > 0, `meta$` emette `null` (core-libs/core/src/cms/facade/page-meta.service.ts:L64-L66).

3.5 `PageMetaService`: composizione del `PageMeta`

1. `getResolverMethods()` trasforma `pageMeta.resolvers` (array di `{property, method}`) in una mappa `{ property: method }`; senza config restituisce `{}` (core-libs/core/src/cms/facade/page-meta.service.ts:L115-L125).
2. La config di default mappa 7 proprietà: `title→resolveTitle`, `heading→resolveHeading`, `breadcrumbs→resolveBreadcrumbs`, `description→resolveDescription`, `image→resolveImage`, `robots→resolveRobots`, `canonicalUrl→resolveCanonicalUrl` (core-libs/core/src/cms/page/config/default-page-meta.config.ts:L9-L40).
3. `resolve(metaResolver)` tiene solo i metodi che esistono **sull'istanza scelta**, li invoca senza argomenti, mappa ogni risultato in `{ [property]: data }` e unisce tutto con `uniteLatest` + `Object.assign`. Se non resta alcun metodo restituisce `of({})` (core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103).
4. `uniteLatest` emette la prima combinazione in modo sincrono; le successive sono raggruppate con `audit` sull'`asapScheduler`, così più sorgenti che cambiano insieme producono un'unica emissione (core-libs/core/src/util/rxjs/unite-latest.ts:L15-L47). Sotto c'è un `combineLatest` (core-libs/core/src/util/rxjs/unite-latest.ts:L45-L45), che emette solo dopo che **tutte** le sorgenti hanno emesso almeno una volta (deps:rxjs-7.8.2/package/src/internal/observable/combineLatest.ts:L242-L269).
5. `getMeta()` restituisce `meta$`, condiviso con `shareReplay({bufferSize:1, refCount:true})` (core-libs/core/src/cms/facade/page-meta.service.ts:L67-L77).
6. Il costruttore si sottoscrive a `languageService.getActive()` con `skip(1)` e `distinctUntilChanged()`, e a ogni cambio lingua chiama `cms.refreshLatestPage()` (core-libs/core/src/cms/facade/page-meta.service.ts:L42-L50). Questo metodo prende il page context corrente e fa dispatch di `CmsActions.LoadCmsPageData` (core-libs/core/src/cms/facade/cms.service.ts:L197-L204): la pagina viene ricaricata e i meta si ricalcolano nella nuova lingua.

3.6 `BasePageMetaResolver` e `ContentPageMetaResolver`

1. `BasePageMetaResolver` (`providedIn: 'root'`) implementa title, description, breadcrumb, robots e canonical (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L25-L42). **Non** estende `PageMetaResolver`: non viene mai scelto da `resolveApplicable`, serve solo come delegato.
2. `title$`, `description$` e `robots$` derivano da `page.title`, `page.description` e `page.robots || []` (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L47-L61).
3. Il breadcrumb home è `[{ label: translate('common.home'), link: '/' }]` (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L66-L68). `breadcrumb$` lo unisce con i breadcrumb delle route Angular figlie, prodotti da `RoutingPageMetaResolver.resolveBreadcrumbs()` (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L73-L79).
4. `resolveCanonicalUrl(options?)` ricalcola `pageLinkService.getCanonicalUrl(options)` a ogni `NavigationEnd`, con `startWith(null)` per emettere subito (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L101-L107).
5. `ContentPageMetaResolver` estende `PageMetaResolver`, imposta `pageType = PageType.CONTENT_PAGE` e delega i cinque metodi a `BasePageMetaResolver` (core-libs/core/src/cms/page/content-page-meta.resolver.ts:L29-L65). Il suo JSDoc suggerisce di creare resolver più specifici aggiungendo il template (core-libs/core/src/cms/page/content-page-meta.resolver.ts:L21-L28).

3.7 Breadcrumb dalle route Angular

1. `RoutingPageMetaResolver.routes$` scarta la prima route, cioè la root speciale di Angular (core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L47-L50).

2. Ogni route usa il resolver dichiarato in `cxPageMeta.resolver` oppure eredita quello del parent; per la prima route il default è `DefaultRoutePageMetaResolver`. L'URL assoluto si costruisce concatenando i segmenti (core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L64-L85).
3. `resolveBreadcrumbs(options)` per default rimuove la route corrente, a meno che `includeCurrentRoute` sia true (core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L91-L111). `trimCurrentRoute` rimuove anche le route finali con path `''` e poi il primo antenato con path non vuoto (core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L159-L173).
4. La config si legge da `route.routeConfig.data.cxPageMeta`, non da `route.data`, per evitare che le route figlie con path vuoto ereditino la config del parent (core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L186-L198).
5. `RoutePageMetaConfig` ha `breadcrumb?: string | RouteBreadcrumbConfig` (`{raw?, i18n?}`) e `resolver?: Type<any>` (core-libs/core/src/cms/page/routing/route-page-meta.model.ts:L30-L55). `DefaultRoutePageMetaResolver` restituisce `[]` senza config, la label letterale con `raw`, altrimenti traduce la chiave (core-libs/core/src/cms/page/routing/default-route-page-meta.resolver.ts:L36-L71).

3.8 Canonical URL (`PageLinkService`)

1. `getCanonicalUrl(options?, url?)` unisce `pageMeta.canonicalUrl` della config con le `options` passate, e prende come URL `url ?? winRef.location.href ?? ''` (core-libs/core/src/cms/page/routing/page-link.service.ts:L33-L43). In SSR `WindowRef.location` restituisce `{href: serverUrl, origin: serverOrigin}` e lancia un errore se `SERVER_REQUEST_URL` o `SERVER_REQUEST_ORIGIN` non sono forniti (core-libs/core/src/window/window-ref.ts:L70-L89).
2. Con il feature toggle `pageLinkSanitizeCanonicalUrl` l'URL viene parsato con `new URL`: i protocolli diversi da http/https e gli URL malformati diventano `''` (core-libs/core/src/cms/page/routing/page-link.service.ts:L39-L55). Il toggle vale `false` per default (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1079-L1079) e la demo lo attiva (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L367-L367).
3. `buildCanonicalUrl` applica nell'ordine `forceHttps`, `forceWww`, `removeQueryParams` e `forceTrailingSlash` (core-libs/core/src/cms/page/routing/page-link.service.ts:L57-L79). `removeQueryParams` toglie tutta la query string se l'opzione è `true`; se è un array toglie solo le chiavi elencate e conserva le altre (core-libs/core/src/cms/page/routing/page-link.service.ts:L81-L102).
4. I default sono `forceHttps: true`, `forceWww: false`, `removeQueryParams: true`, `forceTrailingSlash: true` (core-libs/core/src/cms/page/config/default-page-meta.config.ts:L41-L46).
5. Sulla PDP il canonical non parte da `location.href`. `ProductPageMetaResolver` risale ricorsivamente al `baseProduct` (scope LIST), costruisce `routingService.getFullUrl({cxRoute:'product', params: product})` e lo passa a `pageLinkService.getCanonicalUrl({}, url)` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L180-L209). `getFullUrl` è `winRef.location.origin + getUrl(commands)` (core-libs/core/src/routing/facade/routing.service.ts:L119-L121). Così tutte le varianti di un prodotto puntano allo stesso URL canonico.

3.9 Resolver di prodotto, categoria e ricerca

- **PDP** (`pageType = PRODUCT_PAGE`). `product$` legge `params['productCode']` dallo stato del router e carica il prodotto con `ProductScope.DETAILS` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L50-L69). Il resolver produce:
  - heading: `pageMetaResolver.product.heading` con `{heading: p.name}` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L76-L84);

  - title: `name | prima categoria | manufacturer` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L91-L102, core-libs/core/src/product/services/product-page-meta.resolver.ts:L151-L161);
  - description: traduzione di `pageMetaResolver.product.description` con `{description: product.summary}`, testo EN `{{description}}` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L108-L116, core-libs/assets/src/translations/en/common.json:L59-L59). Come per title e heading, quindi, un override delle traduzioni cambia il risultato;
  - breadcrumb: home più un crumb per categoria (core-libs/core/src/product/services/product-page-meta.resolver.ts:L122-L139);
  - image: `images.PRIMARY.zoom.url ?? null` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L145-L149);
  - robots: delegati a `BasePageMetaResolver` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L163-L165).
- **Categoria** (`pageType = CATEGORY_PAGE`). `searchPage$` usa i risultati di `ProductSearchService` solo se la pagina contiene un componente `CMSProductListComponent` o `ProductGridComponent`; altrimenti usa la pagina CMS (core-libs/core/src/product/services/category-page-meta.resolver.ts:L41-L64, core-libs/core/src/product/services/category-page-meta.resolver.ts:L119-L128). Il resolver produce:
  - title: `pageMetaResolver.category.title` con `count` e `query` (core-libs/core/src/product/services/category-page-meta.resolver.ts:L66-L78);
  - breadcrumb: se la search page ha `breadcrumbs`, home e poi `/c/<code>` per le facet `category/allCategories` e `/Brands/<name>/c/<code>` per `brand`; altrimenti `[]`, cioè nessun crumb, nemmeno la home (core-libs/core/src/product/services/category-page-meta.resolver.ts:L80-L117, in particolare core-libs/core/src/product/services/category-page-meta.resolver.ts:L85-L89);
  - robots e canonical: delegati a `BasePageMetaResolver` (core-libs/core/src/product/services/category-page-meta.resolver.ts:L130-L142).
- **Ricerca** (`CONTENT_PAGE` + `SearchResultsListPageTemplate`, core-libs/core/src/product/services/search-page-meta.resolver.ts:L46-L55). Il title usa `count` dai risultati e `query` da `params['query']`, con `default_title` come fallback (core-libs/core/src/product/services/search-page-meta.resolver.ts:L35-L73). Robots e canonical sono delegati (core-libs/core/src/product/services/search-page-meta.resolver.ts:L75-L87). Non risolve breadcrumb né description.
- **Coupon**. Stesso type e template della ricerca, con +1/−1 allo score in base alla presenza del query param `couponcode` (core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L44-L54, core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L89-L95). Il title usa `findProductTitle` e il breadcrumb aggiunge "My coupons" solo per utenti loggati (core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L56-L87).
- I testi EN sono in `common.json`: `category.title` "{{count}} result for {{query}}", `search.default_title` "All products", `product.title` "{{title}}" (core-libs/assets/src/translations/en/common.json:L43-L63).

3.10 Scrittura nel DOM: `SeoModule` e `SeoMetaService`

1. `SeoModule` importa `StructuredDataModule` e fornisce `provideDefaultConfig(defaultSeoConfig)`, l'`APP_INITIALIZER` `initSeoService` e `htmlLangProvider` (core-libs/storefront/cms-structure/seo/seo.module.ts:L22-L35).
2. `initSeoService` ottiene `SeoMetaService` dall'`Injector` e chiama `init()` (core-libs/storefront/cms-structure/seo/seo.module.ts:L14-L20). `init()` sottoscrive `pageMetaService.getMeta()`, filtra i `null` con `isNotNullable` e assegna il risultato al setter `meta` (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L32-L37).

3. Il setter `meta` imposta, nell'ordine, title, description, image, robots e canonicalUrl (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L39-L45). `heading` e `breadcrumbs` non finiscono in `<head>`.
   - **title**: `ngTitle.setTitle(title || '')` (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L47-L49), che in Angular assegna `document.title` (angular:packages/platform-browser/src/browser/title.ts:L36-L38);
   - **description**: `updateTag({name:'description'})` se c'è un valore, altrimenti `removeTag` (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L51-L57);
   - **image**: `og:image` scritto con l'attributo `name` e rimosso se manca (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L59-L65);
   - **robots**: se l'array non è vuoto, `content` = elementi uniti da `', '`, es. `INDEX, FOLLOW` (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L67-L71);
   - **canonical**: delegato a `pageMetaLinkService?.setCanonicalLink(url)`. `pageMetaLinkService` è un parametro opzionale del costruttore (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L23-L28, core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L79-L81).
4. `addTag` chiama `ngMeta.updateTag(meta)` solo se `meta.content` è truthy (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L83-L87). `Meta.updateTag` di Angular cerca un tag con lo stesso `name=` (o `property=`): se esiste lo aggiorna, altrimenti ne crea uno e lo appende a `<head>` (angular:packages/platform-browser/src/browser/meta.ts:L137-L146, angular:packages/platform-browser/src/browser/meta.ts:L167-L197).
5. `setCanonicalLink(url)` cerca `link[rel="canonical"]`. Senza url rimuove il link; se il link non esiste lo crea con `Renderer2` e lo appende a `head`; altrimenti ne aggiorna `href` (core-libs/storefront/cms-structure/seo/page-meta-link.service.ts:L33-L53).
6. `htmlLangProvider` è un altro `APP_INITIALIZER`: sottoscrive `languageService.getActive()` e scrive `documentElement.lang`, sostituendo `_` con `-` (core-libs/storefront/cms-structure/seo/html-lang-provider.ts:L10-L30).
7. Preconnect dei media: `MediaModule.forRoot()` registra un `APP_INITIALIZER` che chiama `MediaPreconnectService.addPreconnectLink()` (core-libs/storefront/shared/components/media/media.module.ts:L14-L18, core-libs/storefront/shared/components/media/media.module.ts:L29-L43). Il servizio aggiunge il link solo se l'origin di `mediaService.getBaseUrl()` è valido e diverso da `windowRef.location.origin` (core-libs/storefront/cms-structure/seo/media-preconnect.service.ts:L20-L34). `getBaseUrl()` usa `backend.media.baseUrl ?? backend.occ.baseUrl ?? ''` (core-libs/storefront/shared/components/media/media.service.ts:L399-L403). `addPreconnectLink` evita i duplicati e inserisce il link come primo figlio di `<head>` (core-libs/storefront/cms-structure/seo/page-meta-link.service.ts:L55-L67).

3.11 Structured data (JSON-LD)

1. `StructuredDataModule` importa ed esporta `JsonLdDirective` e registra un `APP_INITIALIZER` che chiama `StructuredDataFactory.build()` (core-libs/storefront/cms-structure/seo/structured-data/structured-data.module.ts:L16-L36).
2. `StructuredDataFactory` inietta `SCHEMA_BUILDER` con `@Optional()`. `build()` procede solo se `isJsonLdRequired()` è vero e se ci sono builder (core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts:L20-L42).
3. `isJsonLdRequired()` è vero sul server, oppure in dev mode quando `disableInDevMode` non è true (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L46-L51). Il default è `disableInDevMode: true` (core-libs/storefront/cms-structure/seo/config/default-seo.config.ts:L9-L15), quindi **per default il JSON-LD si genera solo in SSR**.

4. `collectSchemas()` fa `combineLatest` dei `build()` di tutti gli schema builder, e ogni emissione (l'array degli schemi) va a `scriptBuilder.build(schema)` (core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts:L36-L49).
5. `JsonLdScriptFactory.build` scrive nel `textContent` di un unico `<script id="json-ld" type="application/ld+json">`, appeso al `body` alla prima chiamata (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L35-L39, core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L62-L76). Il contenuto passa da `escapeHtml`, che serializza con `JSON.stringify`, assegna il risultato a `div.textContent` e restituisce `innerHTML`, per evitare l'XSS (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L78-L88).
6. I builder standard sono registrati da `JsonLdBuilderModule`: `SCHEMA_BUILDER` → `ProductSchemaBuilder` e `BreadcrumbSchemaBuilder`; `JSONLD_PRODUCT_BUILDER` → base product, offer e review (core-libs/storefront/cms-structure/seo/structured-data/builders/json-ld-builder.module.ts:L21-L52). **Questo modulo è opt-in**: né `SeoModule` né `StructuredDataModule` lo importano, mentre la demo lo importa esplicitamente (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L246-L247).
7. `ProductSchemaBuilder` parte da `CurrentProductService.getProduct()`. Se c'è un prodotto, combina `{@context, @type:'Product'}` con i risultati dei `JsonLdBuilder` iniettati e li unisce con `Object.assign`; altrimenti emette `{}` (core-libs/storefront/cms-structure/seo/structured-data/builders/product/product-schema.builder.ts:L31-L58). I tre sotto-builder producono:
   - `JsonLdBaseProductBuilder`: `sku`, `name`, `description` (summary), `brand` (manufacturer) e `image` (core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-base-product.builder.ts:L20-L60);
   - `JsonLdProductOfferBuilder`: `offers` con price, currency e availability (core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-offer.builder.ts:L20-L36);
   - `JsonLdProductReviewBuilder`: `aggregateRating` e `review[]` se c'è almeno una recensione (core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-review.builder.ts:L27-L82).
8. `BreadcrumbSchemaBuilder` trasforma `pageMeta.breadcrumbs` in `ListItem` con `position` e `item:{@id: link, name: label}`, aggiunge il title come ultimo elemento e restituisce un `BreadcrumbList` (core-libs/storefront/cms-structure/seo/structured-data/builders/breadcrumb/breadcrumb-schema.builder.ts:L19-L56).
9. `JsonLdDirective` (`[cxJsonLd]`) appende uno script escapato all'elemento host, se `isJsonLdRequired()`. Secondo il JSDoc Spartacus non la usa: è pensata per i merchant (core-libs/storefront/cms-structure/seo/structured-data/json-ld.directive.ts:L10-L47).

3.12 PWA: service worker

1. `PwaModule` importa `ServiceWorkerModule.register('ngsw-worker.js')`, `I18nModule` e i due componenti standalone, che esporta (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L32-L56).
2. `ServiceWorkerModule.register` chiama `provideServiceWorker`, che fornisce `SwRegistrationOptions` con `useValue: options` e registra un app initializer (angular:packages/service-worker/src/module.ts:L30-L38, angular:packages/service-worker/src/provider.ts:L240-L254).
3. `PwaModule` sovrascrive `SwRegistrationOptions` con `pwaConfigurationFactory`, che restituisce `enabled: true` solo fuori da dev mode e con `pwa.enabled` (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L19-L23, core-libs/storefront/cms-structure/pwa/pwa.module.ts:L42-L46).
4. `ngswAppInitializer` termina subito in SSR (`ngServerMode`), quando `options.enabled === false` o quando `serviceWorker` non è disponibile in `navigator` (angular:packages/service-worker/src/provider.ts:L32-

L41). Senza `registrationStrategy` la strategia è `'registerWhenStable:30000'` (angular:packages/service-worker/src/provider.ts:L69-L73).

5. Nella demo: `pwa: { enabled: false, addToHomeScreen: true }` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L49-L54), build prod con `"serviceWorker": false` (projects/storefrontapp/project.json:L169-L169), e `index.html` che linka `manifest.json` (projects/storefrontapp/src/index.html:L6-L6), incluso negli assets (projects/storefrontapp/project.json:L26-L26).

3.13 PWA: Add to Home Screen

1. `PwaModule` registra l'`APP_INITIALIZER` `pwaFactory`, che ha come dipendenza `AddToHomeScreenService`. Lo scopo è istanziare il servizio all'avvio: la funzione restituita si limita a ritornarlo (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L25-L30, core-libs/storefront/cms-structure/pwa/pwa.module.ts:L47-L52).
2. Il costruttore di `AddToHomeScreenService` chiama `init()` solo se `pwa.addToHomeScreen` è true (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L26-L34).
3. `init()` registra i listener solo se esiste `nativeWindow`, che in SSR è `undefined` (core-libs/core/src/window/window-ref.ts:L41-L44). Su `beforeinstallprompt` chiama `preventDefault`, salva l'evento e abilita il prompt. Su `appinstalled` mostra il global message `pwa.addedToHomeScreen` di tipo CONFIRMATION, disabilita il prompt e azzera l'evento (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L36-L57).
4. `canPrompt$` deriva da un `BehaviorSubject<boolean>(false)`; `firePrompt()` chiama `deferredEvent.prompt()` se l'evento è presente (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L20-L24, core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L67-L71).
5. La base astratta `AddToHomeScreenComponent` (`@Directive()`) espone `canPrompt$` in `ngOnInit`, e `prompt()` chiama `firePrompt()` (core-libs/storefront/cms-structure/pwa/components/add-to-home-screen.component.ts:L11-L23).
6. `cx-add-to-home-screen-banner` mostra descrizione, due vantaggi e il bottone solo quando `canPrompt$` è true (core-libs/storefront/cms-structure/pwa/components/add-to-home-screen-banner/add-to-home-screen-banner.component.html:L1-L16). `cx-add-to-home-screen-btn` proietta il contenuto solo se `canPrompt$` è true, dentro uno `<span (click)="prompt()">` (core-libs/storefront/cms-structure/pwa/components/add-to-home-screen-btn/add-to-home-screen-btn.component.html:L1-L3).
7. Il banner compare nel messaggio di ringraziamento della conferma d'ordine (feature-libs/order/components/order-confirmation/order-confirmation-thank-you-message/order-confirmation-thank-you-message.component.html:L26-L26), e `OrderConfirmationModule` importa `PwaModule` (feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L56-L56).

---

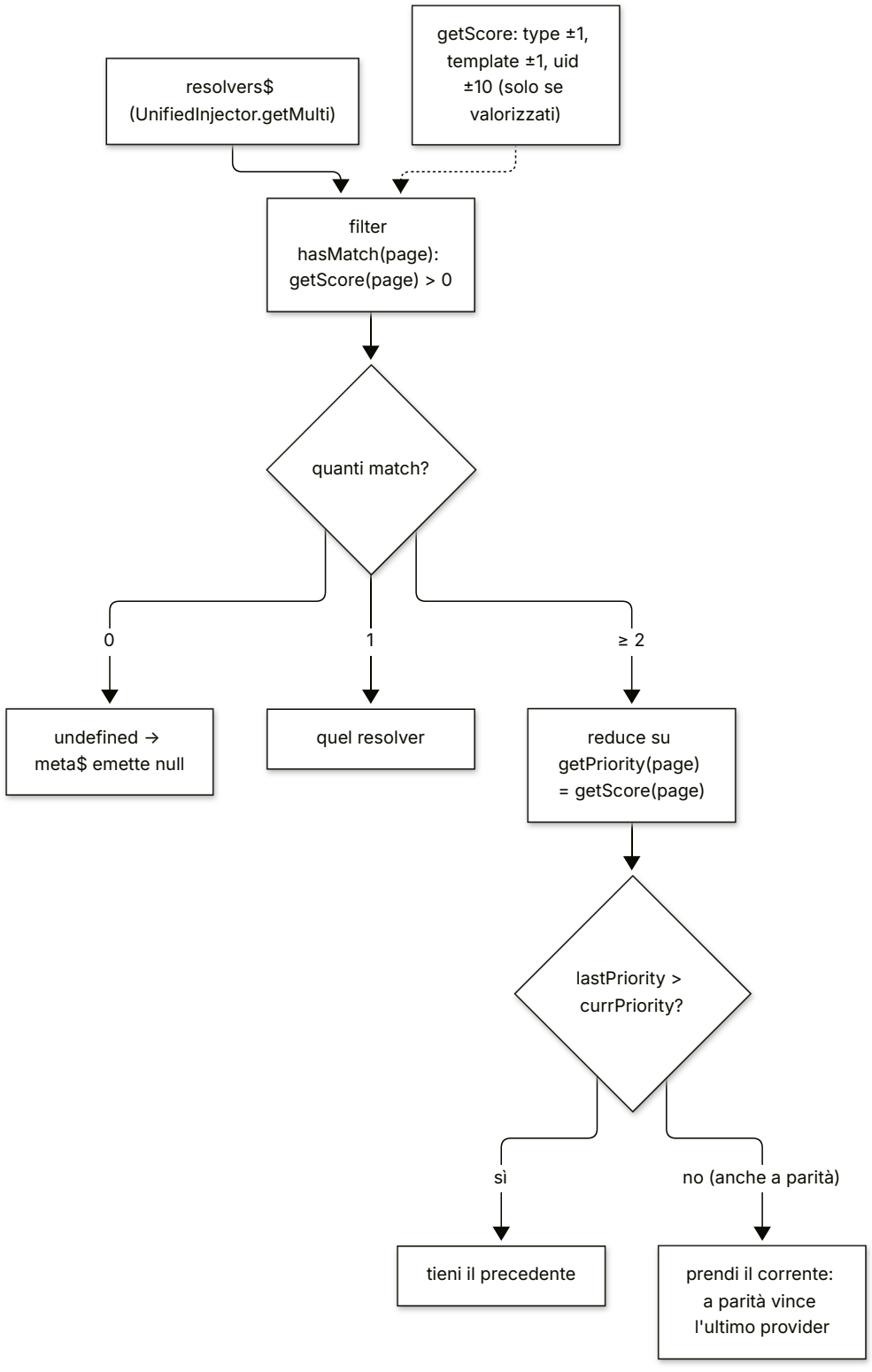
4. Diagrammi

4.1 Dal cambio pagina CMS ai tag in <head>

Ogni messaggio corrisponde ai passi 3.3–3.10.

■ **TAVOLA 40.1** diagramma a tutta pagina, in fondo al capitolo

4.2 Selezione del resolver



4.3 Calcolo del canonical URL

■ TAVOLA 40.2 diagramma a tutta pagina, in fondo al capitolo

4.4 JSON-LD in SSR

■ TAVOLA 40.3 diagramma a tutta pagina, in fondo al capitolo

4.5 PWA: service worker e Add to Home Screen

■ TAVOLA 40.4 diagramma a tutta pagina, in fondo al capitolo

---

5. Estratti di codice reale

5.1 Lo scoring dei resolver

core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L53

```ts
getScore(page: Page): number {
  let score = 0;
  if (this.pageType) {
    score += page.type === this.pageType ? 1 : -1;
  }
  if (this.pageTemplate) {
    score += page.template === this.pageTemplate ? 1 : -1;
  }

  if (this.pageUid) {
    score += page.pageId === this.pageUid ? 10 : -10;
  }

  return score;
}

hasMatch(page: Page): boolean {
  return this.getScore(page) > 0;
}

getPriority(page: Page): number {
  return this.getScore(page);
}
```

5.2 Pipeline `meta$` e composizione

core-libs/core/src/cms/facade/page-meta.service.ts:L59-L68

```ts
protected meta$: Observable<PageMeta | null> = defer(() =>
  this.cms.getCurrentPage()
).pipe(
```

```
    filter((page) => Boolean(page)),
    switchMap((page: Page) => this.getMetaResolver(page)),
    switchMap((metaResolver: PageMetaResolver | undefined) =>
      metaResolver ? this.resolve(metaResolver) : of(null)
    ),
    shareReplay({ bufferSize: 1, refCount: true })
  );
```

core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103

```ts
  protected resolve(metaResolver: PageMetaResolver): Observable<PageMeta> {
    const resolverMethods = this.getResolverMethods();
    const resolvedData: Observable<PageMeta>[] = Object.keys(resolverMethods)
      // TODO: Revisit if typing is possible here with Template Literal Types when we update to TS >=4.1
      .filter((key) => (metaResolver as any)[resolverMethods[key]])
      .map((key) => {
        return (metaResolver as any)
          [resolverMethods[key]]()
          .pipe(map((data) => ({ [key]: data })));
      });

    if (resolvedData.length === 0) {
      // uniteLatest will fail otherwise
      return of({});
    } else {
      return uniteLatest(resolvedData).pipe(
        map((data) => Object.assign({}, ...data))
      );
    }
  }
```

5.3 Selezione con parità a favore dell'ultimo provider

core-libs/core/src/util/applicable.ts:L76-L87

```ts
  let lastPriority = -Infinity;
  return matchedApplicables.reduce((acc: T | undefined, curr: T) => {
    const currPriority = curr.getPriority
      ? curr.getPriority(...priorityParams)
      : Priority.NORMAL;
    if (lastPriority > currPriority) {
      return acc;
    }
    lastPriority = currPriority;
    return curr;
  }, undefined);
}
```

5.4 Normalizzazione del canonical

core-libs/core/src/cms/page/routing/page-link.service.ts:L57-L79

```ts
protected buildCanonicalUrl(
  url: string,
  options: CanonicalUrlOptions
): string {
  if (options.forceHttps) {
    url = url.replace(/^http(?!s):/i, 'https:');
  }

  if (options.forceWww) {
    // this will not allow for not adding wwww. in case of a subdomain
    url = url.replace(/^(https?:\/\/)(?!www\.)(.*)/i, '$1www.$2');
  }

  if (options.removeQueryParams) {
    url = this.removeQueryParams(url, options);
  }

  if (options.forceTrailingSlash) {
    url = url.replace(/^([^\?]+[^\/\?]$)$/i, '$1/');
  }

  return url;
}
```

5.5 I setter di `SeoMetaService`

core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L39-L71

```ts
protected set meta(meta: PageMeta) {
  this.title = meta.title;
  this.description = meta.description;
  this.image = meta.image;
  this.robots = meta.robots;
  this.canonicalUrl = meta.canonicalUrl;
}

protected set title(title: string | undefined) {
  this.ngTitle.setTitle(title || '');
}

protected set description(value: string | undefined) {
  if (value) {
    this.addTag({ name: 'description', content: value });
  } else {
    this.ngMeta.removeTag('name="description"');
```

```
    }
  }

  protected set image(imageUrl: string | undefined) {
    if (imageUrl) {
      this.addTag({ name: 'og:image', content: imageUrl });
    } else {
      this.ngMeta.removeTag('name="og:image"');
    }
  }

  protected set robots(value: PageRobotsMeta[] | undefined) {
    if (value?.length) {
      this.addTag({ name: 'robots', content: value.join(', ') });
    }
  }
```

5.6 Quando si genera il JSON-LD, e come viene protetto

core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L46-L51

```ts
  isJsonLdRequired(): boolean {
    return (
      !isPlatformBrowser(this.platformId) ||
      (isDevMode() && !this.config.seo?.structuredData?.disableInDevMode)
    );
  }
```

core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L84-L88

```ts
  escapeHtml(schema: {}): string {
    const div: HTMLScriptElement = this.renderer.createElement('div');
    div.textContent = JSON.stringify(schema);
    return div.innerHTML;
  }
```

5.7 Abilitazione del service worker

core-libs/storefront/cms-structure/pwa/pwa.module.ts:L19-L30

```ts
export function pwaConfigurationFactory(
  pwaConfig: PWAModuleConfig
): SwRegistrationOptions {
  return { enabled: (!isDevMode() && pwaConfig.pwa?.enabled) || false };
}

export function pwaFactory(
```

```
  addToHomeScreenService: AddToHomeScreenService
): any {
  const result = () => addToHomeScreenService;
  return result;
}
```

5.8 Listener Add to Home Screen

core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L36-L57

```ts
  init() {
    if (this.winRef.nativeWindow) {
      this.winRef.nativeWindow.addEventListener(
        'beforeinstallprompt',
        (event) => {
          event.preventDefault();
          this.deferredEvent = event;
          this.enableAddToHomeScreen();
        }
      );

      this.winRef.nativeWindow.addEventListener('appinstalled', () => {
        this.globalMessageService.add(
          { key: 'pwa.addedToHomeScreen' },
          GlobalMessageType.MSG_TYPE_CONFIRMATION
        );

        this.disableAddToHomeScreen();
        this.deferredEvent = null;
      });
    }
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
pageMeta.resolvers	PageMetaResolverConfig[] ({property, method})	7 voci (title, heading, breadcrumbs, description, image, robots, canonicalUrl)	core-libs/core/src/cms/page/config/page-meta.config.ts:L22-L41; default core-libs/core/src/cms/page/config/default-page-meta.config.ts:L11-L40	core-libs/core/src/cms/facade/page-meta.service.ts:L115-L125
pageMeta.canonicalUrl.forceHttps	boolean	true	core-libs/core/src/cms/page/config/page-meta.config.ts:L49-L52; default core-libs/core/src/cms/page/config/default-page-meta.config.ts:L42-L42	core-libs/core/src/cms/page/routing/page-link.service.ts:L61-L63

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
pageMeta.canonicalUrl.forceWww	boolean	false	core-libs/core/src/cms/page/config/page-meta.config.ts:L54-L59; default core-libs/core/src/cms/page/config/default-page-meta.config.ts:L43-L43	core-libs/core/src/cms/page/routing/page-link.service.ts:L65-L68
pageMeta.canonicalUrl.removeQueryParams	boolean | string[]	true	core-libs/core/src/cms/page/config/page-meta.config.ts:L61-L70; default core-libs/core/src/cms/page/config/default-page-meta.config.ts:L44-L44	core-libs/core/src/cms/page/routing/page-link.service.ts:L70-L72, core-libs/core/src/cms/page/routing/page-link.service.ts:L81-L102
pageMeta.canonicalUrl.forceTrailingSlash	boolean	true	core-libs/core/src/cms/page/config/page-meta.config.ts:L44-L47; default core-libs/core/src/cms/page/config/default-page-meta.config.ts:L45-L45	core-libs/core/src/cms/page/routing/page-link.service.ts:L74-L76
seo.structuredData.disableInDevMode	boolean	true	core-libs/storefront/cms-structure/seo/config/seo.config.ts:L18-L24; default core-libs/storefront/cms-structure/seo/config/default-seo.config.ts:L9-L15	core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L46-L51
pwa.enabled	boolean	false	core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L14-L26	core-libs/storefront/cms-structure/pwa/pwa.module.ts:L19-L23
pwa.addToHomeScreen	boolean	false	core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L14-L26	core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L31-L33
feature toggle pageLinkSanitizeCanonicalUrl	boolean	false	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L511-L518, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1079-L1079	core-libs/core/src/cms/page/routing/page-link.service.ts:L39-L41
route data.cxPageMeta	RoutePageMetaConfig	—	core-libs/core/src/cms/page/routing/route-page-meta.model.ts:L16-L40	core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L186-L198

Le tre classi di config sono abstract `useExisting: Config` e aumentano `Config` tramite module augmentation: `PageMetaConfig` (core-libs/core/src/cms/page/config/page-meta.config.ts:L10-L20), `SeoConfig` (core-libs/storefront/cms-structure/seo/config/seo.config.ts:L10-L28) e `PWAModuleConfig` (core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L10-L30). Si configurano quindi con `provideConfig({...})` come qualsiasi altra chiave (vedi 11 — Configurazione).

Esempio di config (illustrativo, basato sui tipi `CanonicalUrlOptions` in core-libs/core/src/cms/page/config/page-meta.config.ts:L43-L71 e `PWAModuleConfig` in core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L14-L19):

```ts
provideConfig({
  pageMeta: {
    canonicalUrl: {
      forceWww: true,
      removeQueryParams: ['utm_source', 'utm_medium'],
    },
  },
  pwa: { enabled: true, addToHomeScreen: true },
});
```

Nota: `pageMeta.resolvers` è un array, e il deep merge della config sostituisce gli array invece di concatenarli. Una config custom deve quindi elencare anche le voci di default che vuole mantenere. `Config` è costruita da `deepMerge({}, inject(DefaultConfig), inject(RootConfig))` (core-libs/core/src/config/config-tokens.ts:L10-L12). `deepMerge` scende ricorsivamente solo nei valori per cui `isObject` è vero, e `isObject` esclude gli array: un array viene assegnato così com'è (core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/config/utils/deep-merge.ts:L38-L45). Per i dettagli sul merge vedi 11 — Configurazione.

---

7. Commenti e note notevoli dal sorgente

- `// @deprecated (will be removed in 12 months)` sul parametro `platformId` di `PageMetaService` (core-libs/core/src/cms/facade/page-meta.service.ts:L39-L40).
- `// NOTE: Solution to the issue: https://jira.tools.sap/browse/CXSPA-10923`: il cambio lingua forza il ricaricamento della pagina CMS per aggiornare il title, saltando la prima emissione per non duplicare il refresh all'avvio (core-libs/core/src/cms/facade/page-meta.service.ts:L42-L44).
- `// TODO: Revisit if typing is possible here with Template Literal Types when we update to TS >=4.1`: il codice è su TypeScript 5.9 ma il TODO è ancora lì (core-libs/core/src/cms/facade/page-meta.service.ts:L87-L87).
- `// uniteLatest will fail otherwise`: giustifica il ramo `of({})` (core-libs/core/src/cms/facade/page-meta.service.ts:L96-L96).
- `// this will not allow for not adding wwww. in case of a subdomain`: `forceWww` non riconosce i sottodomini (core-libs/core/src/cms/page/routing/page-link.service.ts:L66-L66), come dice anche il JSDoc della config (core-libs/core/src/cms/page/config/page-meta.config.ts:L54-L59).
- Il JSDoc di `ProductPageMetaResolver.resolveCanonicalUrl` ammette due limiti: con varianti multi-livello OCC espone solo il `baseProduct` diretto, e l'URL della variante contiene solo il codice prodotto, non dati per URL "pretty" (core-libs/core/src/product/services/product-page-meta.resolver.ts:L167-L179).
- JSDoc di `JsonLdDirective`: "This helper directive is actually not used in Spartacus" (core-libs/storefront/cms-structure/seo/structured-data/json-ld.directive.ts:L10-L20).
- JSDoc di `escapeHtml`: "The given schema is not trusted, as malicious code could be injected (XSS)" (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L78-L83).
- Canonical: la rimozione del link è un "edge case, but useful if the canonical url is created in CSR while developing/testing" (core-libs/storefront/cms-structure/seo/page-meta-link.service.ts:L38-L42), e il JSDoc del setter dice che i canonical "are (typically) only added in SSR" (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L73-L78).
- Preconnect: "we want the preconnect-link to be at the top of the <head> section" (core-libs/storefront/cms-structure/seo/page-meta-link.service.ts:L66-L66), e non serve se il dominio dei media

coincide con quello della storefront (core-libs/storefront/cms-structure/seo/media-preconnect.service.ts:L29-L30).

- La nota lunga sull'uso di `routeConfig.data` invece di `data` spiega l'ereditarietà di `data` per le route figlie con path `''` (core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L189-L196).
- Il JSDoc di `PageHeadingResolver` invita ad aggiungere resolver custom "to the list of `PageMetaService.resolverMethods`", ma quella proprietà non esiste: la mappa è una variabile locale costruita da `getResolverMethods()` sulla config `pageMeta.resolvers` (core-libs/core/src/cms/page/page.resolvers.ts:L17-L18, core-libs/core/src/cms/facade/page-meta.service.ts:L84-L86).

8. Extension point

EXTENSION POINT	COME	SORGENTE	RICETTA
Nuovo resolver di pagina	estendere `PageMetaResolver`, impostare `pageType`/`pageTemplate`/`pageUid`, implementare i `resolve*` e fare multi-provide sul token `PageMetaResolver`	core-libs/core/src/cms/page/page-meta.resolver.ts:L17-L54; registrazione d'esempio core-libs/core/src/cms/page/page-meta.module.ts:L13-L21	52 — Ricette OCC, meta resolver
Riuso dei comportamenti standard	iniettare `BasePageMetaResolver` e delegare, come fa `ContentPageMetaResolver`	core-libs/core/src/cms/page/base-page-meta.resolver.ts:L25-L108, core-libs/core/src/cms/page/content-page-meta.resolver.ts:L41-L64	52
Specializzare un resolver di prodotto	estendere `ProductPageMetaResolver` / `CategoryPageMetaResolver` / `SearchPageMetaResolver` (esportati) e registrarlo dopo l'originale, o con score più alto	export core-libs/core/src/product/services/index.ts:L7-L9; registrazione core-libs/core/src/product/product.module.ts:L16-L37	52
Nuova proprietà di meta	aggiungere `{property, method}` a `pageMeta.resolvers`, ripetendo anche le 7 voci di default perché l'array custom sostituisce quello di default (sezione 6), e implementare il metodo sul resolver	core-libs/core/src/cms/page/config/page-meta.config.ts:L22-L41, core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103	50 — Ricette config e DI
Regole del canonical	config `pageMeta.canonicalUrl`, oppure override di `PageLinkService` (metodi `protected` `buildCanonicalUrl`, `removeQueryParams`, `sanitizeUrl`)	core-libs/core/src/cms/page/routing/page-link.service.ts:L45-L102	50
Breadcrumb di route Angular	`data: { cxPageMeta: { breadcrumb, resolver } }` sulla route; resolver custom che estende `DefaultRoutePageMetaResolver`	core-libs/core/src/cms/page/routing/route-page-meta.model.ts:L30-L55, core-libs/core/src/cms/page/routing/default-route-page-meta.resolver.ts:L21-L79	51 — Ricette CMS, outlet, route
Scrittura dei tag	sostituire `SeoMetaService` via DI e ridefinire i setter `protected` (es. `og:image` con `property`, robots rimovibile)	core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L39-L87	50
Canonical/preconnect nel DOM	override di `PageMetaLinkService` o `MediaPreconnectService`	core-libs/storefront/cms-structure/seo/page-meta-link.service.ts:L33-L71, core-libs/storefront/cms-structure/seo/media-preconnect.service.ts:L20-L34	50

EXTENSION POINT	COME	SORGENTE	RICETTA
Nuovo schema JSON-LD	classe che implementa `SchemaBuilder` e multi-provide su `SCHEMA_BUILDER`	core-libs/storefront/cms-structure/seo/structured-data/builders/schema.interface.ts:L9-L11, core-libs/storefront/cms-structure/seo/structured-data/builders/tokens.ts:L15-L15	50
Campi aggiuntivi nello schema Product	`JsonLdBuilder<Product>` in multi-provide su `JSONLD_PRODUCT_BUILDER`	core-libs/storefront/cms-structure/seo/structured-data/builders/tokens.ts:L21-L23, core-libs/storefront/cms-structure/seo/structured-data/builders/product/product-schema.builder.ts:L44-L58	50
Schema statico in un componente	direttiva `[cxJsonLd]`	core-libs/storefront/cms-structure/seo/structured-data/json-ld.directive.ts:L21-L47	51
UI Add to Home Screen custom	estendere `AddToHomeScreenComponent` o usare `AddToHomeScreenService` (`canPrompt$`, `firePrompt`, `enableAddToHomeScreen`, `disableAddToHomeScreen`)	core-libs/storefront/cms-structure/pwa/components/add-to-home-screen.component.ts:L11-L23, core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L59-L71	51
Opzioni del service worker	sovrascrivere `SwRegistrationOptions` (es. `registrationStrategy`) dopo `PwaModule`	core-libs/storefront/cms-structure/pwa/pwa.module.ts:L42-L46, angular:packages/service-worker/src/provider.ts:L69-L73	50

Esempio di resolver custom per una pagina di contenuto con template specifico. È illustrativo e usa solo simboli esportati da `@spartacus/core`: `PageMetaResolver` (core-libs/core/src/cms/page/index.ts:L11-L11), `BasePageMetaResolver` (core-libs/core/src/cms/page/index.ts:L7-L7), interfacce `Page*Resolver` (core-libs/core/src/cms/page/index.ts:L12-L12).

```ts
@Injectable({ providedIn: 'root' })
export class StorePageMetaResolver
  extends PageMetaResolver
  implements PageTitleResolver, PageRobotsResolver, CanonicalPageResolver
{
  constructor(protected base: BasePageMetaResolver) {
    super();
    this.pageType = PageType.CONTENT_PAGE;     // +1
    this.pageTemplate = 'StoreFinderPageTemplate'; // +1 → score 2 > ContentPageMetaResolver (1)
  }
  resolveTitle() { return this.base.resolveTitle(); }
  resolveRobots() { return this.base.resolveRobots(); }
  resolveCanonicalUrl() { return this.base.resolveCanonicalUrl({ forceWww: true }); }
}
// provider: { provide: PageMetaResolver, useExisting: StorePageMetaResolver, multi: true }
```

DA VERIFICARE: il nome di template `'StoreFinderPageTemplate'` è solo un esempio e non è stato verificato nei dati CMS. I template reali dipendono dal backend.

Per l'API pubblica e la stabilità negli upgrade (per esempio `CouponSearchPageResolver` non esportato, toggle `pageLinkSanitizeCanonicalUrl`) vedi 54 — API pubblica e upgrade e 42 — Feature toggle e deprecazioni.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Un resolver che non emette blocca tutti i meta.** `resolve()` combina i metodi con `uniteLatest`, che è un `combineLatest` (core-libs/core/src/util/rxjs/unite-latest.ts:L45-L45), e `combineLatest` emette solo dopo che tutte le sorgenti hanno emesso (deps:rxjs-7.8.2/package/src/internal/observable/combineLatest.ts:L242-L269). Un esempio concreto: `CategoryPageMetaResolver.resolveTitle` filtra le pagine senza `pagination` (core-libs/core/src/product/services/category-page-meta.resolver.ts:L66-L69), e se la pagina categoria non contiene un componente PLP `searchPage$` emette la `Page` CMS, che non ha `pagination` (core-libs/core/src/product/services/category-page-meta.resolver.ts:L45-L52). In quel caso `meta$` non emette per quella pagina. DA VERIFICARE a runtime: è una deduzione dal codice, che non tiene conto di eventuali proprietà `pagination` aggiunte alla `Page` da normalizer custom.
2. **Il robots meta non viene mai rimosso.** Il setter non ha un ramo `else` (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L67-L71). Se si naviga da una pagina `NOINDEX` a una con `robots` vuoto (array vuoto quando il backend non manda `robotTag`, core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L198-L222), in CSR il vecchio tag resta nel DOM.
3. `og:image` **con** `name` **invece di** `property`**.** Open Graph usa `property`, ma Spartacus scrive `{ name: 'og:image' }` (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L59-L65). Angular usa `name` come selettore quando è presente (angular:packages/platform-browser/src/browser/meta.ts:L194-L197).
4. **Niente JSON-LD senza** `JsonLdBuilderModule`**.** Il modulo non è importato da `SeoModule` né da `StructuredDataModule`. Senza di esso `SCHEMA_BUILDER` è vuoto e `build()` non fa nulla (core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts:L35-L35). La demo lo importa a mano (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L246-L247), mentre un grep su `core-libs/schematics` non trova alcuna occorrenza di `JsonLdBuilderModule`: la schematic non lo genera.
5. **JSON-LD solo in SSR.** In browser in produzione `isJsonLdRequired()` è false, e in dev è spento da `disableInDevMode: true` (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L46-L51, core-libs/storefront/cms-structure/seo/config/default-seo.config.ts:L11-L13). Chi testa con `ng serve` non vede lo script finché non imposta `disableInDevMode: false`.
6. **Schema del breadcrumb e** `null` **nell'array.** Senza breadcrumb, `BreadcrumbSchemaBuilder.collect` restituisce `undefined` (core-libs/storefront/cms-structure/seo/structured-data/builders/breadcrumb/breadcrumb-schema.builder.ts:L25-L28), che entra nell'array passato a `escapeHtml` (core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts:L47-L49). `JSON.stringify` serializza un `undefined` dentro un array come `null`. Inoltre l'ultimo `ListItem` usa il title come `@id`, non un URL (core-libs/storefront/cms-structure/seo/structured-data/builders/breadcrumb/breadcrumb-schema.builder.ts:L40-L49).
7. **Dettagli schema.org discutibili.** `@context` è `http://schema.org` (core-libs/storefront/cms-structure/seo/structured-data/builders/product/product-schema.builder.ts:L52-L55). Il `@type` delle recensioni è `'review'` minuscolo (core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-review.builder.ts:L55-L57). `datePublished` non ha zero iniziali e usa l'ora locale (core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-review.builder.ts:L62-L67). `availability` vale `'InStock'`/`'OutOfStock'` senza prefisso URL, e `lowStock` finisce

in `OutOfStock` (core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-offer.builder.ts:L30-L33).

8. **Parità di score.** A parità vince il provider registrato per ultimo (core-libs/core/src/util/applicable.ts:L81-L85). Un resolver custom con lo stesso score di uno standard vince solo se è registrato dopo; per non dipendere dall'ordine conviene alzare lo score con `pageTemplate` o `pageUid`.
9. `CouponSearchPageResolver` **e** `ActivatedRoute`**.** Il getter `couponCode` legge `this.route.snapshot?.queryParams?.couponcode` da un `ActivatedRoute` iniettato in un servizio root (core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L48-L48, core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L93-L95). Nell'injector root Angular fornisce `routerState.root` (angular:packages/router/src/provide_router.ts:L112-L120), e lo snapshot root riceve i query params dell'URL (angular:packages/router/src/recognize.ts:L122-L126), quindi il parametro è visibile. DA VERIFICARE il timing: lo score è calcolato quando cambia la pagina CMS, e non è stato verificato che in quel momento lo snapshot della root sia già quello della nuova navigazione.
10. `CouponSearchPageResolver` **non è esportato** dal barrel `core-libs/core/src/product/services/index.ts` (core-libs/core/src/product/services/index.ts:L7-L11), quindi non si può estendere importandolo da `@spartacus/core`.
11. `forceTrailingSlash` **e query residue.** La regex esclude `?`: se `removeQueryParams` è un array e restano parametri, la slash finale non viene aggiunta (core-libs/core/src/cms/page/routing/page-link.service.ts:L74-L76).
12. `forceWww` **sui sottodomini.** Produce `www.shop.example.com` (core-libs/core/src/cms/page/routing/page-link.service.ts:L65-L68).
13. **Sanitize del canonical spento per default.** Senza il toggle, un `location.href` con protocollo non http finisce nel canonical (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1079-L1079, core-libs/core/src/cms/page/routing/page-link.service.ts:L39-L42).
14. **Le** `CanonicalUrlOptions` **per-chiamata non arrivano dal flusso standard.** `PageMetaService.resolve` invoca i metodi senza argomenti (core-libs/core/src/cms/facade/page-meta.service.ts:L89-L92). Il parametro `options` di `BasePageMetaResolver.resolveCanonicalUrl` è utile solo se un resolver custom lo passa esplicitamente, come nell'esempio della sezione 8.
15. `resolveImage` **della PDP restituisce** `null` pur avendo tipo `Observable<string>` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L145-L149). `SeoMetaService` lo tratta come falsy e rimuove il tag (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L59-L65).
16. **Preconnect con URL completo.** `MediaPreconnectService` controlla l'origin ma passa a `addPreconnectLink` l'URL intero restituito da `getBaseUrl()` (core-libs/storefront/cms-structure/seo/media-preconnect.service.ts:L21-L32).
17. `htmlLangProvider` **non esportato e mai chiuso.** Non compare in `seo/index.ts` (core-libs/storefront/cms-structure/seo/index.ts:L7-L12), e la subscription a `getActive()` non viene mai annullata (core-libs/storefront/cms-structure/seo/html-lang-provider.ts:L24-L27).
18. **Service worker spento in dev, e la demo non lo genera.** `!isDevMode()` blocca la registrazione anche con `pwa.enabled: true` (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L22-L22). La build prod della demo ha `serviceWorker: false` (projects/storefrontapp/project.json:L169-L169), quindi `ngsw-worker.js` non viene prodotto. Per attivare davvero la PWA servono `pwa.enabled`, una build con service worker e una configurazione ngsw. In `projects/storefrontapp` non c'è un `ngsw-config.json`. Spartacus ne fornisce uno di riferimento in `core-libs/storefront/cms-structure/pwa/ngsw-config.json` (core-libs/storefront/cms-structure/pwa/ngsw-config.json:L1-L6), che la demo non usa: la sua build prod ha `"serviceWorker": false` (projects/storefrontapp/project.json:L169-L169).
19. `PwaModule` **importato due volte.** Lo importano sia `BaseStorefrontModule` (core-libs/storefront/base-storefront.module.ts:L33-L33) sia `OrderConfirmationModule` (feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L56-L56). DA VERIFICARE l'effetto runtime quando

`OrderConfirmationModule` è caricato in lazy: `ServiceWorkerModule.register` e l'`APP_INITIALIZER` verrebbero ri-dichiarati nell'injector della feature.

20. **Listener mai rimossi e bottone poco accessibile.** I listener `beforeinstallprompt`/`appinstalled` non vengono mai rimossi (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L36-L57). Nel bottone, lo `<span (click)>` è sempre renderizzato e non ha `role` né `tabindex`, quindi non è raggiungibile da tastiera (core-libs/storefront/cms-structure/pwa/components/add-to-home-screen-btn/add-to-home-screen-btn.component.html:L1-L3).
21. **Errore con provider non multi.** `UnifiedInjector.getMulti` lancia `Multi-providers mixed with single providers` se `PageMetaResolver` è fornito senza `multi: true` in un injector (core-libs/core/src/lazy-loading/unified-injector.ts:L83-L88).
22. **JSDoc fuorvianti.** `SearchPageMetaResolver` cita `PageType.CATEGORY_PAGE` ma imposta `CONTENT_PAGE` (core-libs/core/src/product/services/search-page-meta.resolver.ts:L22-L27, core-libs/core/src/product/services/search-page-meta.resolver.ts:L53-L53). `CategoryPageMetaResolver` dice "title, and breadcrumbs only" ma risolve anche robots e canonical (core-libs/core/src/product/services/category-page-meta.resolver.ts:L28-L32, core-libs/core/src/product/services/category-page-meta.resolver.ts:L130-L142). `setCanonicalLink` parla di un "id" che il metodo non accetta (core-libs/storefront/cms-structure/seo/page-meta-link.service.ts:L27-L33).
23. **Field initializer che dipende dal costruttore.** In `JsonLdScriptFactory`, `renderer` è inizializzato con `this.rendererFactory`, una parameter property (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L23-L33). Funziona perché il `tsconfig.json` di root imposta `"useDefineForClassFields": false` (tsconfig.json:L799-L799) e la libreria lo eredita (core-libs/storefront/tsconfig.lib.json:L1-L5). Attivare quell'opzione romperebbe la classe.
24. **Cambio lingua = nuovo load CMS.** Ogni cambio di lingua attiva dopo il primo dispatcha `LoadCmsPageData` (core-libs/core/src/cms/facade/page-meta.service.ts:L42-L50, core-libs/core/src/cms/facade/cms.service.ts:L197-L204), anche se gli unici dati da aggiornare fossero i meta.

---

10. Come fare debug

- **Il title non cambia.** Verificare prima che arrivi un `PageMeta`: sottoscrivere `PageMetaService.getMeta()` in un componente di test e loggare. `null` significa che nessun resolver ha score > 0 (core-libs/core/src/cms/facade/page-meta.service.ts:L64-L66). Se invece non arriva nulla, uno dei `resolve*` del resolver scelto non emette (trappola 1). Per capirlo, sottoscrivere singolarmente i metodi elencati in `pageMeta.resolvers` (core-libs/core/src/cms/page/config/default-page-meta.config.ts:L11-L40).
- **Quale resolver viene scelto?** Leggere `page.type`, `page.template` e `page.pageId` dalla pagina CMS (`CmsService.getCurrentPage()`) e calcolare a mano lo score con la formula di core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L45. In alternativa, mettere un breakpoint in `PageMetaService.getMetaResolver` (core-libs/core/src/cms/facade/page-meta.service.ts:L133-L139) e ispezionare l'array `resolvers`.
- **Un resolver registrato in un modulo lazy non viene usato.** `getMulti` legge gli injector dei moduli lazy solo dopo il caricamento (core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35). Se la pagina viene visualizzata prima che la feature sia caricata, il resolver non c'è ancora.
- **Canonical inatteso.** Chiamare `PageLinkService.getCanonicalUrl()` dalla console tramite un componente di debug e confrontare l'URL grezzo (`location.href`) con le opzioni effettive (core-libs/core/src/cms/page/routing/page-link.service.ts:L33-L43). In SSR, se compare l'errore `Cannot resolve the href as the SERVER_REQUEST_URL is undefined`, il server non ha fornito il token (core-libs/core/src/window/window-ref.ts:L74-L78).

- **Il JSON-LD non compare.** Nell'ordine: (a) è importato `JsonLdBuilderModule`? (b) si sta guardando l'HTML SSR (view-source) e non il DOM CSR? (c) in dev, `seo.structuredData.disableInDevMode` è `false`? Il punto di controllo è `StructuredDataFactory.build` (core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts:L34-L42). Lo script cercato è `#json-ld` in fondo al `body` (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L62-L76).
- **Hydration e JSON-LD.** DA VERIFICARE se lo `<script id="json-ld">` generato in SSR, che è appeso al `body` fuori dall'app root, resta nel DOM dopo l'idratazione client: non è stato confermato a runtime.
- **Service worker non registrato.** Controllare `isDevMode()` e `pwa.enabled` (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L19-L23), la presenza di `ngsw-worker.js` nell'output della build (projects/storefrontapp/project.json:L169-L169) e il pannello Application → Service Workers del browser. Ricordare che la registrazione di default avviene quando l'app è stabile o dopo 30 secondi (angular:packages/service-worker/src/provider.ts:L69-L73).
- **Il banner A2HS non appare.** Il banner dipende da `beforeinstallprompt`, che emette il browser. Servono `pwa.addToHomeScreen: true` (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L31-L33) e i criteri di installabilità del browser (manifest, HTTPS, service worker). Per testare la UI si può chiamare `AddToHomeScreenService.enableAddToHomeScreen()` (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L59-L61). Il prompt reale però non parte senza `deferredEvent` (core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L67-L71).
- `<html lang>` **sbagliato.** Il valore è la lingua attiva con `_` → `-` (core-libs/storefront/cms-structure/seo/html-lang-provider.ts:L24-L27). Verificare la lingua attiva del site context (vedi 13 — Site context e theme).

---

11. Domande di autoverifica

1. **Su una pagina** `CONTENT_PAGE` **con template** `SearchResultsListPageTemplate` **e query param** `couponcode=ABC`**, quale resolver viene scelto e con che score?** `CouponSearchPageResolver` con score 3: +1 type, +1 template, +1 coupon. La ricerca ha 2 e il content resolver 1 (core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L45, core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L89-L91).
2. **Due resolver hanno lo stesso score: quale vince?** Quello fornito per ultimo, perché `resolveApplicable` sostituisce l'accumulatore anche a parità (`lastPriority > currPriority` è stretto) (core-libs/core/src/util/applicable.ts:L81-L85).
3. **Perché** `BasePageMetaResolver` **non viene mai selezionato da** `PageMetaService`**?** Perché non estende `PageMetaResolver` e non è multi-provided su quel token: è un servizio root usato per delega (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L25-L42, core-libs/core/src/cms/page/page-meta.module.ts:L13-L21).
4. **Come si aggiunge una nuova proprietà al** `PageMeta` **calcolato?** Si aggiunge `{property, method}` a `pageMeta.resolvers` e si implementa `method` sul resolver. La config custom deve ripetere anche le voci di default, perché `deepMerge` sostituisce gli array (core-libs/core/src/config/utils/deep-merge.ts:L38-L45). `resolve()` invoca solo i metodi esistenti sull'istanza scelta (core-libs/core/src/cms/page/config/page-meta.config.ts:L22-L41, core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103). Per scriverla nel DOM serve anche un `SeoMetaService` custom, perché il setter `meta` gestisce solo cinque proprietà (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L39-L45).
5. **Con la config di default, cosa diventa** `http://shop.com/p/123?utm=x`**?** `https://shop.com/p/123/`: forceHttps, nessun www, query rimossa, slash finale aggiunta (core-libs/core/src/cms/page/config/

default-page-meta.config.ts:L41-L46, core-libs/core/src/cms/page/routing/page-link.service.ts:L57-L79).

6. **Perché il canonical di una variante di prodotto punta al prodotto base?** `resolveCanonicalUrl` della PDP risale ricorsivamente `baseProduct` e costruisce l'URL della route `product` con quel prodotto (core-libs/core/src/product/services/product-page-meta.resolver.ts:L180-L209).
7. **In quali condizioni viene generato lo script JSON-LD?** Quando `isJsonLdRequired()` è vero (server, oppure dev mode con `disableInDevMode` non true) **e** sono registrati degli `SCHEMA_BUILDER`, cioè c'è `JsonLdBuilderModule` o un builder custom (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L46-L51, core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts:L34-L42).
8. **Come si evita che contenuti del prodotto iniettino HTML nello script JSON-LD?** `escapeHtml` serializza con `JSON.stringify`, assegna la stringa a `textContent` di un `div` e legge `innerHTML`, così i caratteri HTML vengono escapati (core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L78-L88).
9. **Con** `pwa.enabled: true` **e** `ng serve`**, il service worker si registra?** No. `pwaConfigurationFactory` richiede `!isDevMode()` (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L19-L23), e `ngswAppInitializer` esce quando `enabled === false` (angular:packages/service-worker/src/provider.ts:L32-L41).
10. **Perché** `AddToHomeScreenService` **viene istanziato all'avvio anche se nessun componente lo usa?** Perché `PwaModule` registra l'`APP_INITIALIZER` `pwaFactory` con `deps: [AddToHomeScreenService]`, e il costruttore del servizio chiama `init()` se `pwa.addToHomeScreen` è true (core-libs/storefront/cms-structure/pwa/pwa.module.ts:L47-L52, core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L26-L34).

TAVOLA 40.1 **4.1 Dal cambio pagina CMS ai tag in <head>**

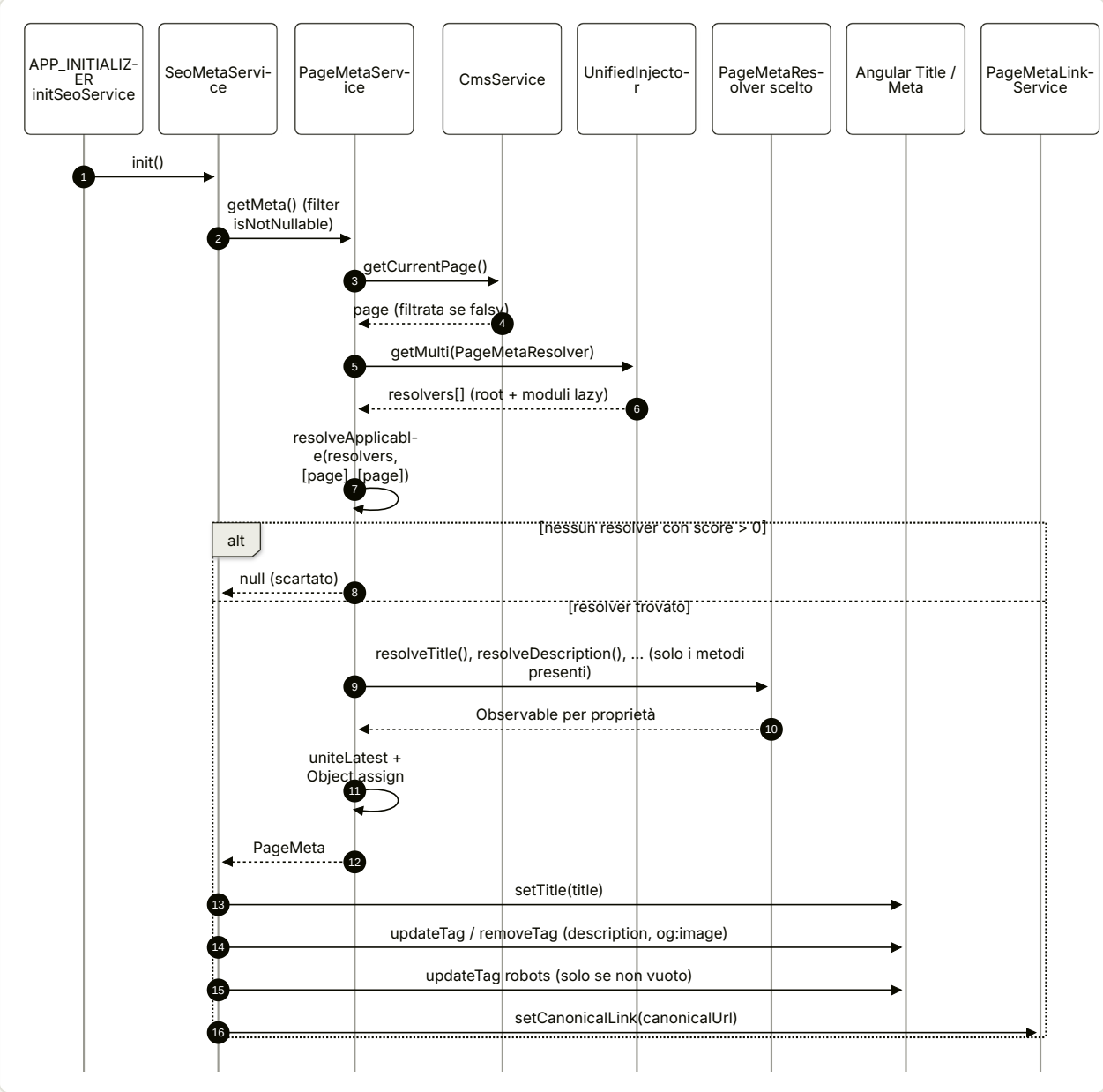


TAVOLA 40.2 **4.3 Calcolo del canonical URL**

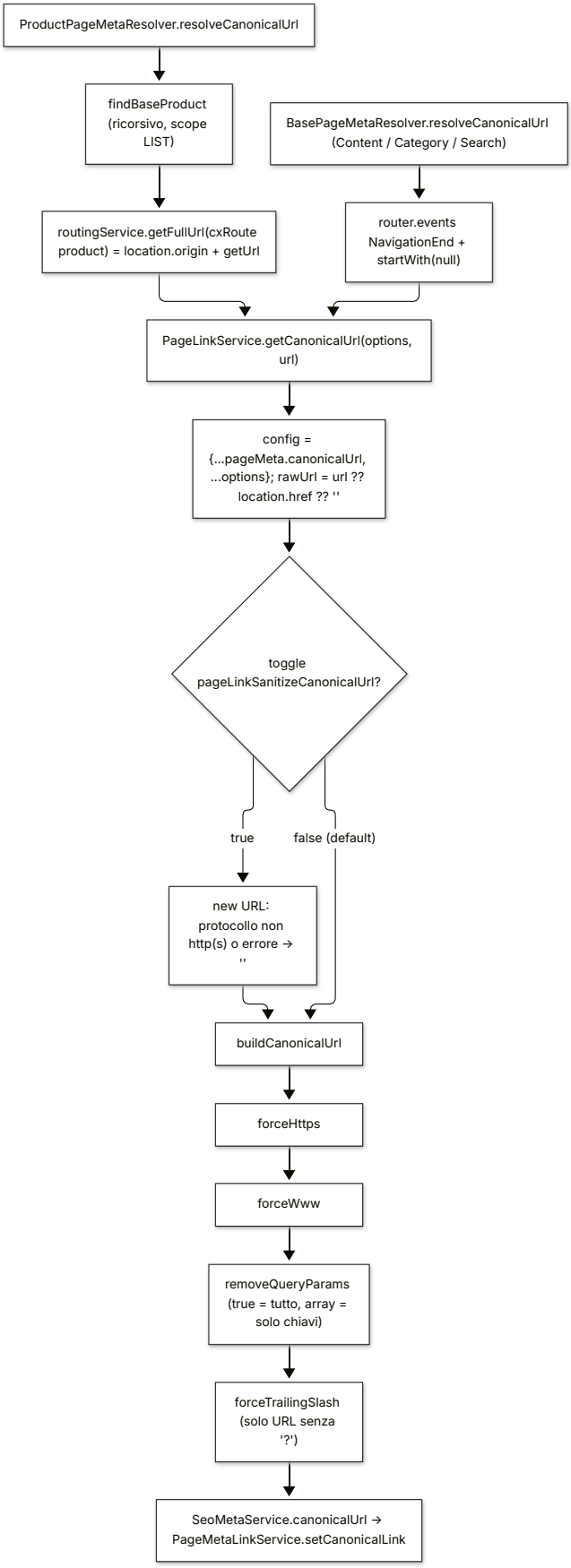


TAVOLA 40.3 **4.4 JSON-LD in SSR**

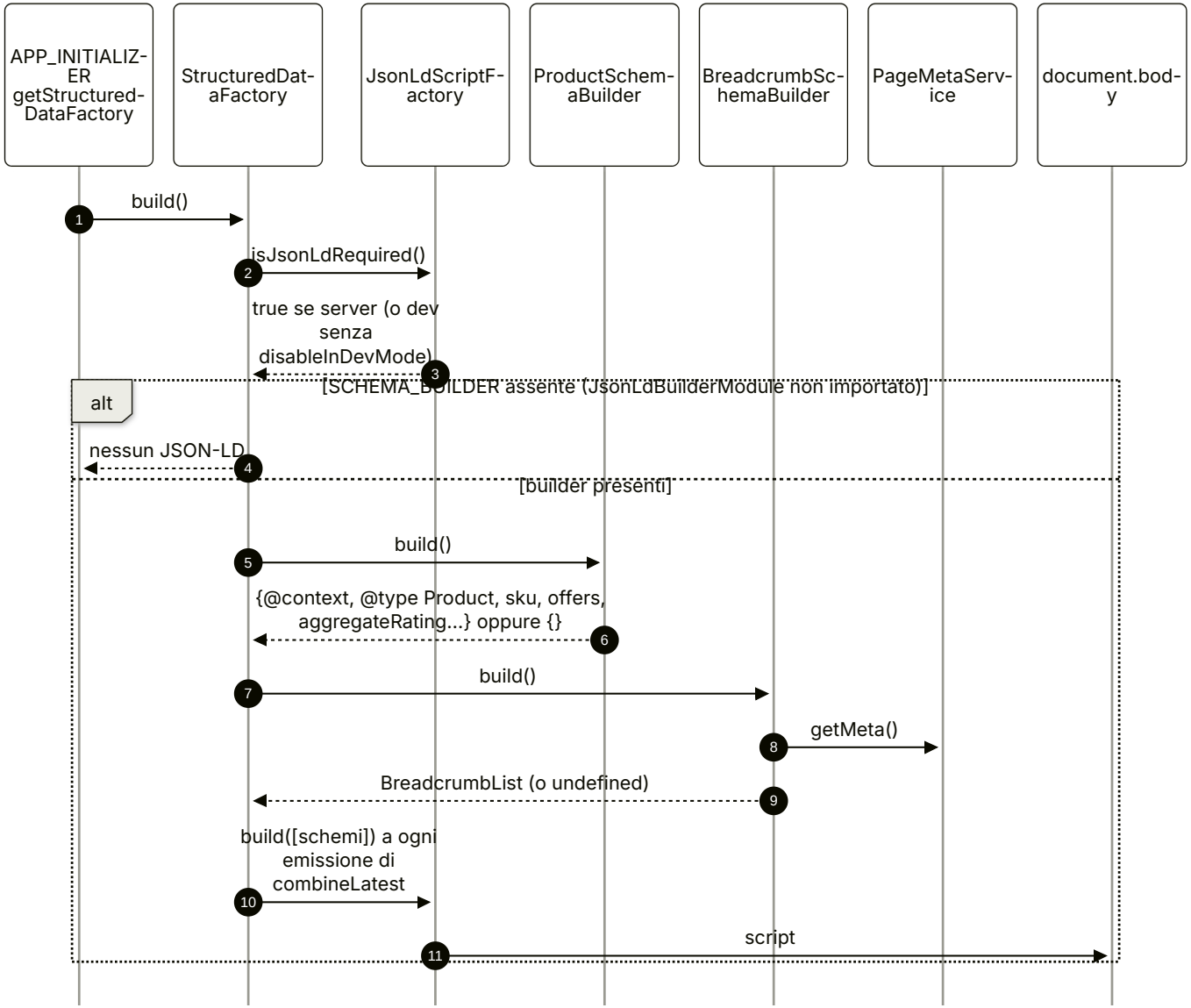
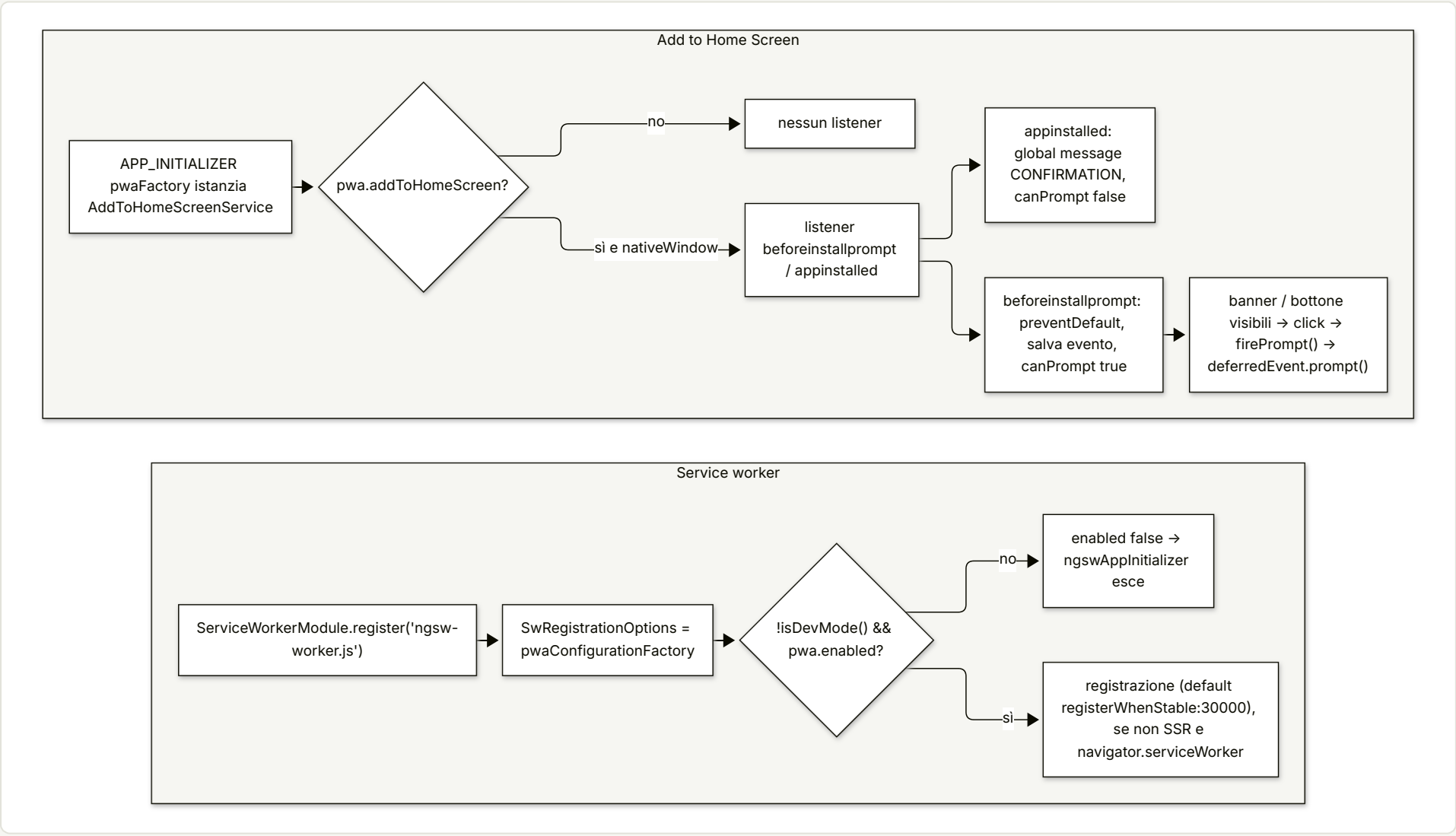


TAVOLA 40.4 **4.5 PWA: service worker e Add to Home Screen**



41

Errori HTTP, error handling, logger e global message

Due sottosistemi distinti: global message (UX) e error handling (osservabilità/SSR).

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Quando una chiamata OCC fallisce, in Spartacus succedono **due cose indipendenti**, gestite da due sottosistemi diversi che è facile confondere perché hanno nomi quasi identici:

SOTTOSISTEMA	CLASSE CHIAVE	DOMANDA A CUI RISPONDE	DOVE AGISCE
Global message (UX)	`HttpErrorInterceptor` + `HttpErrorHandler` (uno per status)	"Che messaggio mostro all'utente?"	browser e server (ma i messaggi hanno senso nel browser)
Error handling (osservabilità / SSR)	`HttpErrorHandlerInterceptor` + `CxErrorHandler` + `MULTI_ERROR_HANDLER`	"Chi deve sapere che c'è stato un errore? Devo far fallire il rendering SSR?"	inoltro di errori HTTP e fail action: solo in SSR (core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L72-L74). `CxErrorHandler` è comunque attivo anche nel browser (errori di componenti ed effect — deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L176-L178 —, listener globali — projects/storefrontapp/src/app/app.config.ts:L31-L31)

Il primo sottosistema vive in `core-libs/core/src/global-message/`: un interceptor HTTP sceglie **un solo** handler applicabile per lo status della risposta (core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L45-L65) e l'handler, di norma, aggiunge una chiave di traduzione allo store NgRx `global-message` tramite `GlobalMessageService.add` (core-libs/core/src/global-message/facade/global-message.service.ts:L43-L55). Il componente `cx-global-message` legge lo store e mostra gli alert (core-libs/storefront/cms-components/misc/global-message/global-message.component.ts:L44-L50); un effect li rimuove dopo un timeout configurabile (core-libs/core/src/global-message/store/effects/global-message.effect.ts:L70-L106).

Il secondo sottosistema vive in `core-libs/core/src/error-handling/`: sostituisce l'`ErrorHandler` di Angular con `CxErrorHandler` (core-libs/core/src/error-handling/error-handling.module.ts:L20-L31), che fa da *fan-out* verso un array di `MultiErrorHandler` (core-libs/core/src/error-handling/cx-error-handler.ts:L26-L30). Due "sorgenti" alimentano l'`ErrorHandler`:

- gli errori HTTP, tramite `HttpErrorHandlerInterceptor`, che li incapsula in `OutboundHttpError` / `CmsPageNotFoundOutboundHttpError` (core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L76-L81);
- le *fail action* NgRx che portano una proprietà `error`, tramite `CxErrorHandlerEffect` ed `ErrorActionService` (core-libs/core/src/error-handling/effects-error-handler/cx-error-handler.effect.ts:L23-L32, core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L24-L37).

Entrambe le sorgenti, però, inoltrano solo quando **non** sono nel browser (`!windowRef.isBrowser()`): il motivo dichiarato è non esporre nella console del browser dati potenzialmente confidenziali, mentre in SSR si vuole catturare tutto per poter rispondere con un 404/500 (core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L63-L74). In SSR `provideServer()` aggiunge infatti un secondo multi handler che propaga l'errore al server Express (core-libs/setup/ssr/providers/ssr-providers.ts:L37-L45), dove `defaultExpressErrorHandlers` sceglie 404 o 500 (core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32).

Trasversale a tutto c'è `LoggerService`: un servizio `providedIn: 'root'` che delega al `console` ed è pensato come extension point per redirigere i log (core-libs/core/src/logger/logger.service.ts:L9-L22). L'unico multi error handler di default, `LoggingErrorHandler`, scrive proprio lì (core-libs/core/src/error-handling/multi-error-handler/logging-error-handler.ts:L17-L26), e in SSR il logger è sostituito da una versione server (core-libs/setup/ssr/logger/services/server-logger-service-factory.ts:L12-L17).

Questo capitolo si concentra su `@spartacus/core`. La parte Express/SSR (dove viene fornito `PROPAGATE_ERROR_TO_SERVER`, formato dei log server) è trattata in 39 — SSR ottimizzato; la catena di interceptor di Angular in 06 — HttpClient e interceptor.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/error-handling/error-handling.module.ts	`ErrorHandlingModule`: importa effects/http error handler; `forRoot()` sostituisce `ErrorHandler` con `CxErrorHandler` e registra `LoggingErrorHandler`
core-libs/core/src/error-handling/cx-error-handler.ts	`CxErrorHandler`: fan-out su `MULTI_ERROR_HANDLER`
core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts	interfaccia `MultiErrorHandler`, token `MULTI_ERROR_HANDLER`
core-libs/core/src/error-handling/multi-error-handler/logging-error-handler.ts	`LoggingErrorHandler` → `LoggerService.error`
core-libs/core/src/error-handling/multi-error-handler/provide-multi-error-handler.ts	`provideMultiErrorHandler()` (non esportato dal barrel)
core-libs/core/src/error-handling/multi-error-handler/index.ts	barrel
core-libs/core/src/error-handling/http-error-handler/http-error-handler.module.ts	`HttpErrorHandlerModule.forRoot()`: registra l'interceptor
core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts	`HttpErrorHandlerInterceptor`: errori HTTP → `ErrorHandler` (solo SSR)
core-libs/core/src/error-handling/http-error-handler/outbound-http-error.ts	`OutboundHttpError`, `CmsPageNotFoundOutboundHttpError`
core-libs/core/src/error-handling/http-error-handler/index.ts	barrel
core-libs/core/src/error-handling/effects-error-handler/effects-error-handler.module.ts	`EffectsErrorHandlerModule`: effect + `ErrorActionService`
core-libs/core/src/error-handling/effects-error-handler/cx-error-handler.effect.ts	`CxErrorHandlerEffect` (non esportato)
core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts	`ErrorActionService`: filtro action d'errore, esclusione errori HTTP
core-libs/core/src/error-handling/effects-error-handler/error-action.ts	interfaccia `ErrorAction`
core-libs/core/src/error-handling/effects-error-handler/index.ts	barrel

PATH	RUOLO
core-libs/core/src/error-handling/index.ts	barrel dell'area
core-libs/core/src/global-message/global-message.module.ts	`GlobalMessageModule`: store, effect, default config; `forRoot()` aggiunge handler + interceptor
core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts	`HttpErrorInterceptor`: sceglie l'`HttpErrorHandler` applicabile (non esportato)
core-libs/core/src/global-message/http-interceptors/index.ts	array `errorHandlers` e `httpErrorInterceptors`
core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts	classe astratta `HttpErrorHandler` (`Applicable`)
core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts	400: password, login, validation, duplicate uid, unknown identifier, illegal argument
core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts	403: messaggio + `logout()` se `users/current`
core-libs/core/src/global-message/http-interceptors/handlers/not-found/not-found.handler.ts	404: no-op intenzionale
core-libs/core/src/global-message/http-interceptors/handlers/conflict/conflict.handler.ts	409
core-libs/core/src/global-message/http-interceptors/handlers/internal-server/internal-server.handler.ts	500
core-libs/core/src/global-message/http-interceptors/handlers/bad-gateway/bad-gateway.handler.ts	502
core-libs/core/src/global-message/http-interceptors/handlers/gateway/gateway-timeout.handler.ts	504
core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts	fallback: `warn` solo in dev mode
core-libs/core/src/global-message/http-interceptors/handlers/index.ts	barrel degli handler
core-libs/core/src/global-message/models/global-message.model.ts	`GlobalMessageType`, `GlobalMessage`
core-libs/core/src/global-message/models/response-status.model.ts	`HttpResponseStatus`
core-libs/core/src/global-message/facade/global-message.service.ts	`GlobalMessageService` (`get`/`add`/`remove`)
core-libs/core/src/global-message/config/global-message-config.ts	`GlobalMessageConfig`, `GlobalMessageTypeConfig`, augmentation di `Config`
core-libs/core/src/global-message/config/default-global-message-config.ts	`defaultGlobalMessageConfig` (non pubblico)

PATH	RUOLO
core-libs/core/src/global-message/store/global-message-state.ts	GLOBAL_MESSAGE_FEATURE, interfacce di stato
core-libs/core/src/global-message/store/global-message-store.module.ts	GlobalMessageStoreModule
core-libs/core/src/global-message/store/actions/global-message.actions.ts	AddMessage, RemoveMessage, RemoveMessagesByType
core-libs/core/src/global-message/store/reducers/global-message.reducer.ts	reducer, initialState
core-libs/core/src/global-message/store/reducers/index.ts	reducerToken, reducerProvider, getReducers
core-libs/core/src/global-message/store/selectors/global-message.selectors.ts	entities / byType / countByType
core-libs/core/src/global-message/store/effects/global-message.effect.ts	GlobalMessageEffect: dedup + auto-hide
core-libs/core/src/global-message/index.ts	barrel pubblico dell'area
core-libs/core/src/logger/logger.service.ts	LoggerService
core-libs/core/src/logger/testing/mock-logger.service.ts	MockLoggerService silenzioso
core-libs/core/src/logger/testing/logger-testing.module.ts	LoggerTestingModule
core-libs/core/src/util/applicable.ts	Applicable, Priority, resolveApplicable (fuori area, ma decisivo)
core-libs/core/src/util/try-normalize-http-error.ts	tryNormalizeHttpError → HttpErrorModel
core-libs/storefront/cms-components/misc/global-message/global-message.component.ts	GlobalMessageComponent (cx-global-message)
core-libs/setup/ssr/providers/ssr-providers.ts	provideServer(): logger server + PropagatingToServerErrorHandler
core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts	defaultExpressErrorHandlers: 404/500 + fallback CSR

3. Come funziona passo per passo

3.1 Bootstrap: chi registra cosa, e in che ordine

1. BaseStorefrontModule importa BaseCoreModule.forRoot() e GlobalMessageComponentModule — core-libs/storefront/base-storefront.module.ts:L26-L30.
2. BaseCoreModule importa **per primo** ErrorHandlingModule.forRoot(), con il commento "Import this module before any other interceptor to handle HTTP errors efficiently" — core-libs/core/src/base-core.module.ts:L26-L29.
3. ErrorHandlingModule importa EffectsErrorHandlerModule.forRoot() e HttpErrorHandlerModule.forRoot() — core-libs/core/src/error-handling/error-handling.module.ts:L13-L18. Il secondo registra HttpErrorHandlerInterceptor in HTTP_INTERCEPTORS (useExisting, multi) — core-libs/core/src/

error-handling/http-error-handler/http-error-handler.module.ts:L11-L24. Essendo il primo interceptor registrato, è il più **esterno** della catena.

4. `ErrorHandlingModule.forRoot()` aggiunge `provideMultiErrorHandler()` (→ `MULTI_ERROR_HANDLER = LoggingErrorHandler`) e `{ provide: ErrorHandler, useClass: CxErrorHandler }` — core-libs/core/src/error-handling/error-handling.module.ts:L20-L31, core-libs/core/src/error-handling/multi-error-handler/provide-multi-error-handler.ts:L11-L19.
5. `EffectsErrorHandlerModule` registra `CxErrorHandlerEffect` via `EffectsModule.forFeature` e, nel `forRoot()`, fornisce `ErrorActionService` — core-libs/core/src/error-handling/effects-error-handler/effects-error-handler.module.ts:L12-L22.
6. Più avanti `BaseCoreModule` importa `GlobalMessageModule.forRoot()` (dopo `CmsModule.forRoot()`, prima di `HttpModule.forRoot()`) — core-libs/core/src/base-core.module.ts:L34-L42.
7. `GlobalMessageModule` importa lo store e l'effect e fornisce `provideDefaultConfig(defaultGlobalMessageConfig)` — core-libs/core/src/global-message/global-message.module.ts:L19-L25; `forRoot()` aggiunge `...errorHandlers, ...httpErrorInterceptors` — core-libs/core/src/global-message/global-message.module.ts:L26-L32.
8. `errorHandlers` registra 8 handler come multi-provider del token-classe `HttpErrorHandler` con `useExisting`: Unknown, BadGateway, BadRequest, Conflict, Forbidden, GatewayTimeout, InternalServerError, NotFound — core-libs/core/src/global-message/http-interceptors/index.ts:L22-L63. `httpErrorInterceptors` registra `HttpErrorInterceptor` — core-libs/core/src/global-message/http-interceptors/index.ts:L65-L71.
9. `HttpModule.forRoot()` porta `HttpTimeoutInterceptor`, registrato dopo e quindi più interno — core-libs/core/src/base-core.module.ts:L42-L42, core-libs/core/src/http/http-timeout/http-timeout.module.ts:L13-L23.
10. Nell'app demo, `BaseStorefrontModule` precede `SpartacusFeaturesModule` (che contiene `AuthModule.forRoot()`), quindi gli interceptor di autenticazione sono ancora più interni — projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L14-L15, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L179-L179.
11. Angular compone `HTTP_INTERCEPTORS` con `reduceRight`: dato `[a, b, c]` costruisce `c(b(a(end)))`, cioè il primo registrato vede la richiesta per primo e l'errore **per ultimo** — angular:packages/common/http/src/interceptor.ts:L232-L240.
12. In SSR, `provideServer()` (usato dall'`AppServerModule` della demo, projects/storefrontapp/src/app/app.module.server.ts:L10-L17) sostituisce `LoggerService` con `serverLoggerServiceFactory` e aggiunge `PropagatingToServerErrorHandler` come secondo `MULTI_ERROR_HANDLER` — core-libs/setup/ssr/providers/ssr-providers.ts:L27-L46.

Il risultato, per una richiesta OCC nella demo, è una catena (dall'esterno all'interno): `HttpErrorHandlerInterceptor` → … → `HttpErrorInterceptor` → … → `HttpTimeoutInterceptor` → … → interceptor auth → backend.

> DA VERIFICARE — L'ordine effettivo degli interceptor intermedi (site context, consensi, ecc.) dipende dall'ordine degli import dell'applicazione; qui è verificato solo l'ordine relativo dei quattro citati e solo per `projects/storefrontapp` (dossier Q4). Nella demo `AppModule` (e quindi tutta la catena) è caricato con `importProvidersFrom(AppModule)` e non ci sono altri `HTTP_INTERCEPTORS` registrati prima — projects/storefrontapp/src/app/app.config.ts:L26-L35.

3.2 Il ramo "global message": dall'`HttpErrorResponse` all'alert

1. La risposta d'errore risale la catena dall'interno verso l'esterno — angular:packages/common/http/src/interceptor.ts:L232-L240.

2. `HttpErrorInterceptor.intercept` aggancia un `tap({ error })`: l'errore **non** viene catturato né sostituito, continua a propagarsi al chiamante. Reagisce solo se l'errore è `instanceof HttpErrorResponse` — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L30-L43.
3. L'Observable degli handler è creato alla costruzione dell'interceptor con `UnifiedInjector.getMulti(HttpErrorHandler)` più `shareReplay(1)` — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L22-L28. Viene sottoscritto al primo errore e da allora resta attivo (`shareReplay` senza `refCount`), accumulando anche gli handler dei moduli lazy caricati in seguito; i moduli già caricati sono riemessi perché `modules$` è un `ReplaySubject` connesso — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L55. `getMulti` interroga il root injector e gli injector dei moduli lazy, accumulando le istanze con `scan` — core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93; `injectors$` parte con `startWith(this.rootInjector)` — core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35. Per il funzionamento dell'`UnifiedInjector` vedi 12 — Feature module e lazy loading.
4. `getResponseHandler` legge **sincronamente** l'ultimo array con `getLastValueSync` (subscribe + unsubscribe immediato) e lo passa a `resolveApplicable(handlers, [response])` — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L59-L65, core-libs/core/src/util/rxjs/get-last-value-sync.ts:L9-L23.
5. `resolveApplicable` tiene solo gli handler con `hasMatch(response) === true` (o senza `hasMatch`); se ne restano meno di due restituisce il primo — core-libs/core/src/util/applicable.ts:L63-L74. Altrimenti vince la priorità più alta; a parità vince quello registrato **più tardi**; senza `getPriority` vale `Priority.NORMAL` — core-libs/core/src/util/applicable.ts:L76-L86, regola documentata nel JSDoc core-libs/core/src/util/applicable.ts:L54-L57.
6. L'`hasMatch` di default di `HttpErrorHandler` confronta `errorResponse.status === this.responseStatus` — core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L42-L44. `UnknownErrorHandler` sovrascrive `hasMatch` restituendo sempre `true` e ha `Priority.FALLBACK` (−50) — core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L22-L50. Tutti gli altri handler di default hanno `Priority.LOW` (−10) — es. core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts:L39-L41. Valori di `Priority`: core-libs/core/src/util/applicable.ts:L7-L24.
7. Viene invocato **un solo** handler: `handler.handleError(request, response)` — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L45-L53.
8. L'handler, tipicamente, chiama `globalMessageService.add({ key: 'httpHandlers.xxx' }, GlobalMessageType.MSG_TYPE_ERROR)` — es. core-libs/core/src/global-message/http-interceptors/handlers/internal-server/internal-server.handler.ts:L16-L28. Le traduzioni inglesi stanno sotto `httpHandlers` in core-libs/assets/src/translations/en/common.json:L116-L172 (vedi 37 — i18n).
9. `GlobalMessageService.add` dispatcha `AddMessage`, avvolgendo una stringa in `{ raw: text }` — core-libs/core/src/global-message/facade/global-message.service.ts:L43-L55.
10. Il reducer accoda `message.text` all'array del tipo (creandolo se manca); `timeout` **non** viene salvato nello stato — core-libs/core/src/global-message/store/reducers/global-message.reducer.ts:L24-L45.
11. `GlobalMessageEffect.removeDuplicated$`: se dopo l'aggiunta esistono più testi deep-equal, dispatcha `RemoveMessage` sull'indice della **prima** occorrenza — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L31-L68, utilità di confronto in core-libs/core/src/util/object-comparison-utils.ts:L77-L92.
12. `GlobalMessageEffect.hideAfterDelay$` (solo browser) attende `message.timeout || config.timeout` e poi rimuove l'indice `0` del tipo — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L73-L101.
13. `GlobalMessageComponent` sottoscrive `globalMessageService.get()` e rende gli alert; la X chiama `remove(type, index)` — core-libs/storefront/cms-components/misc/global-message/global-message.

component.ts:L44-L50, template core-libs/storefront/cms-components/misc/global-message/global-message.component.html:L25-L94.

3.3 Il ramo "error handling": dall'errore HTTP all'`ErrorHandler`

1. `HttpErrorHandlerInterceptor` (il più esterno) usa anch'esso `tap({ error })` e osserva **qualsiasi** errore, non solo `HttpErrorResponse` (anche ad es. un `TimeoutError` RxJS lanciato da un interceptor successivo, come nota il JSDoc) — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L48-L61, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L34-L35.
2. `shouldHandleError` restituisce `!this.windowRef.isBrowser()`: **nel browser si ferma qui** — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L63-L74.
3. In SSR, `handleError` avvolge l'errore in `CmsPageNotFoundOutboundHttpError` se è un 404 sull'endpoint OCC `pages`, altrimenti in `OutboundHttpError`, e chiama `errorHandler.handleError` — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L76-L81.
4. Il test "CMS page not found" costruisce l'URL atteso con `occEndpointsService.buildUrl('pages', { urlParams: { userId } })` (userId letto da un signal ottenuto con `toSignal(UserIdService.getUserId())`) e verifica `HttpErrorResponse`, status 404 e `url.startsWith(expectedUrl)` — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L46-L46, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L89-L98.
5. `OutboundHttpError` estende `Error` con messaggio `'Outbound HTTP Error'` e conserva l'originale in `cause`; la sottoclasse sovrascrive `message` con `'CMS Page Not Found'` — core-libs/core/src/error-handling/http-error-handler/outbound-http-error.ts:L10-L25.
6. L'`ErrorHandler` iniettato è `CxErrorHandler`, che chiama `handleError(error)` su ogni elemento di `MULTI_ERROR_HANDLER`, nell'ordine di registrazione e senza `try/catch` — core-libs/core/src/error-handling/cx-error-handler.ts:L18-L30, core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L20-L27.
7. `LoggingErrorHandler` → `this.logger.error(error)` — core-libs/core/src/error-handling/multi-error-handler/logging-error-handler.ts:L17-L26. In SSR il `LoggerService` è `ExpressLoggerService` se esiste il token `REQUEST`, altrimenti `PrerenderingLoggerService` — core-libs/setup/ssr/logger/services/server-logger-service-factory.ts:L12-L17.
8. `PropagatingToServerErrorHandler` chiama `PROPAGATE_ERROR_TO_SERVER(error)` (opzionale, fallback noop) — core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts:L29-L37.
9. Lato Express, `defaultExpressErrorHandlers` (solo se `!res.headersSent`) imposta `Cache-Control: no-store` e risponde 404 per `CmsPageNotFoundOutboundHttpError`, 500 altrimenti, inviando il documento come fallback CSR — core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32.

3.4 Il ramo "error handling" per le fail action NgRx

1. Un effect cattura un errore e dispatcha una fail action con proprietà `error`: lo fanno di serie `LoaderFailAction` ed `EntityFailAction`, che implementano `ErrorAction` — core-libs/core/src/state/utils/loader/loader.action.ts:L74-L83, core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L80-L88; ed esempi "manuali" come `LoadDeliveryCountriesFail` — core-libs/core/src/user/store/actions/delivery-countries.action.ts:L25-L32. Per le action loader/entity vedi 19 — State management.

2. Gli errori HTTP sono di norma normalizzati con `tryNormalizeHttpError`, che converte `HttpErrorResponse` in `HttpErrorModel` (message, status, statusText, url, details) o restituisce l'errore così com'è — core-libs/core/src/util/try-normalize-http-error.ts:L21-L57.
3. `CxErrorHandlerEffect.error$` (`dispatch: false`) filtra con `errorActionService.isErrorAction` e chiama `errorActionService.handle` — core-libs/core/src/error-handling/effects-error-handler/cx-error-handler.effect.ts:L23-L32.
4. `isErrorAction` è semplicemente `'error' in action` — core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L39-L43.
5. `handle` scarta gli errori `HttpErrorModel` e `HttpErrorResponse` ("already handled in HttpErrorHandlerInterceptor") e inoltra gli altri all'`ErrorHandler` solo se `shouldHandleError`, che ancora una volta è `!windowRef.isBrowser()` — core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L24-L37, core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L54-L56.
6. Da qui il percorso è quello di 3.3 passi 6-9.

Nota: indipendentemente da Spartacus, NgRx stesso chiama `errorHandler.handleError(error)` quando un effect va in errore (non quando dispatcha una fail action) e lo risottoscrive fino a 10 volte — deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L174-L185. Quel percorso arriva a `CxErrorHandler` **anche nel browser**, perché non passa da `shouldHandleError`.

3.5 L'`ErrorHandler` di Angular: cosa sostituisce Spartacus

L'`ErrorHandler` di default di Angular si limita a `console.error('ERROR', error)` — angular:packages/core/src/error_handler.ts:L54-L63. Angular lo risolve pigramente dall'`EnvironmentInjector` tramite `INTERNAL_APPLICATION_ERROR_HANDLER` e lo esegue fuori dalla zona — angular:packages/core/src/error_handler.ts:L68-L91. I listener globali `error`/`unhandledrejection` del browser (via `provideBrowserGlobalErrorListeners`) non sono attivi in server mode — angular:packages/core/src/error_handler.ts:L108-L124. Spartacus non cambia questo meccanismo: cambia solo *quale* classe risponde al token `ErrorHandler` (core-libs/core/src/error-handling/error-handling.module.ts:L25-L28). Il risultato pratico: nel browser, un errore non gestito di un componente arriva a `CxErrorHandler` → `LoggingErrorHandler` → `console.error` via `LoggerService`.

3.6 Timeout e errori di rete

1. `HttpTimeoutInterceptor` (interno) converte un `TimeoutError` RxJS in `HttpErrorResponse({ url, error })` senza status — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L101-L110; un `HttpErrorResponse` senza status vale `0` ("Unknown Error") — angular:packages/common/http/src/response.ts:L401-L421.
2. In `HttpErrorInterceptor` nessun handler ha `responseStatus` 0, quindi vince `UnknownErrorHandler`, che non aggiunge messaggi e fa solo `logger.warn` in `isDevMode()` — core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L30-L43.
3. In SSR `HttpErrorHandlerInterceptor` lo inoltra come `OutboundHttpError` → 500 — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L76-L81.

3.7 Il `BadRequestHandler` in dettaglio

Il 400 è lo status più "carico" di semantica OCC, perciò `BadRequestHandler` (priorità LOW) esegue **in sequenza** sei sotto-handler, ognuno dei quali filtra gli `ErrorModel` della risposta — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L22-L33:

SOTTO-HANDLER	CONDIZIONE	MESSAGGIO	RIFERIMENTO
handleBadPassword	URL contiene / authorizationserver/oauth/token, error === 'invalid_grant', body grant_type === 'password'	chiave da getErrorTranslationKey(error_description), param errorMessage; poi remove(MSG_TYPE_CONFIRMATION)	core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L17-L17, core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L35-L67
handleBadLoginResponse	type === 'PasswordMismatchError'	path /password → httpHandlers.badRequestOldPasswordIncorrect, altrimenti httpHandlers.validationErrors.invalid.password	core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L73-L87
handleValidationError	type === 'ValidationError'	httpHandlers.validationErrors.<reason>.<subject>	core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L94-L103
handleGuestDuplicateEmail	type === 'DuplicateUidError'	httpHandlers.badRequestGuestDuplicateEmail con errorMessage	core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L124-L136
handleUnknownIdentifierError	type === 'UnknownIdentifierError'	error.message raw, altrimenti httpHandlers.unknownIdentifier	core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L143-L152
handleValidationErrorIleggalArgument	type === 'IllegalArgumentError'	error.message raw	core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L106-L118

getErrors legge response.error.errors (o []) escludendo JaloObjectNoLongerValidError — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L155-L159. getErrorTranslationKey (definito nella classe base) converte la descrizione in httpHandlers.badRequest.<snake_case> o, se vuota, in httpHandlers.badRequestPleaseLoginAgain — core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L51-L59. Con il toggle enablePasswordExpiredErrorTranslation, una chiave che inizia con ...password_expired_for_the_user è rimappata su httpHandlers.badRequest.password_expired — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L50-L57.

Le feature lib specializzano il 400 con handler a priorità più alta: ad esempio BadCartRequestHandler usa Priority.NORMAL e restringe hasMatch a super.hasMatch(...) && getErrors(...).some(isCartError) — feature-libs/cart/base/core/http-interceptors/handlers/bad-cart-request.handler.ts:L21-L33, registrato in CartBaseCoreModule — feature-libs/cart/base/core/cart-base-core.module.ts:L37-L46. Così un 400 "di carrello" va **solo** a BadCartRequestHandler, mentre un 400 generico continua ad andare a BadRequestHandler. Pattern analoghi in checkout B2B, quote, organization, pdf-invoices, configuratore, ticketing, requested delivery date (dossier F+2, es. feature-libs/quote/root/quote-root.module.ts:L110-L117).

3.8 Gli altri handler per status

STATUS	HANDLER	PRIORITÀ	EFFETTO	RIFERIMENTO
403	`ForbiddenHandler`	LOW	se `request.url` termina con l'URL OCC `user` per `current` → `authService.logout()`; sempre `httpHandlers.forbidden`	`core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts:L23-L37`
404	`NotFoundHandler`	LOW	nessuno (vuoto per non cadere nell'Unknown)	`core-libs/core/src/global-message/http-interceptors/handlers/not-found/not-found.handler.ts:L15-L24`
409	`ConflictHandler`	LOW	`httpHandlers.conflict`	`core-libs/core/src/global-message/http-interceptors/handlers/conflict/conflict.handler.ts:L16-L28`
500	`InternalServerErrorHandler`	LOW	`httpHandlers.internalServerError`	`core-libs/core/src/global-message/http-interceptors/handlers/internal-server/internal-server.handler.ts:L16-L28`
502	`BadGatewayHandler`	LOW	`httpHandlers.badGateway`	`core-libs/core/src/global-message/http-interceptors/handlers/bad-gateway/bad-gateway.handler.ts:L16-L28`
504	`GatewayTimeoutHandler`	LOW	`httpHandlers.gatewayTimeout`	`core-libs/core/src/global-message/http-interceptors/handlers/gateway/gateway-timeout.handler.ts:L16-L28`
qualsiasi	`UnknownErrorHandler`	FALLBACK	solo `warn` in dev mode	`core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L22-L50`

`HttpResponseStatus` include anche `UNAUTHORIZED = 401`, ma nessun handler di default lo usa — `core-libs/core/src/global-message/models/response-status.model.ts:L7-L17`, `core-libs/core/src/global-message/http-interceptors/index.ts:L22-L63`. Il 401 è normalmente assorbito dall'`AuthInterceptor` con refresh e retry (vedi 22 — Autenticazione).

4. Diagrammi

4.1 Registrazione e ordine della catena di interceptor

■ **TAVOLA 41.1** diagramma a tutta pagina, in fondo al capitolo

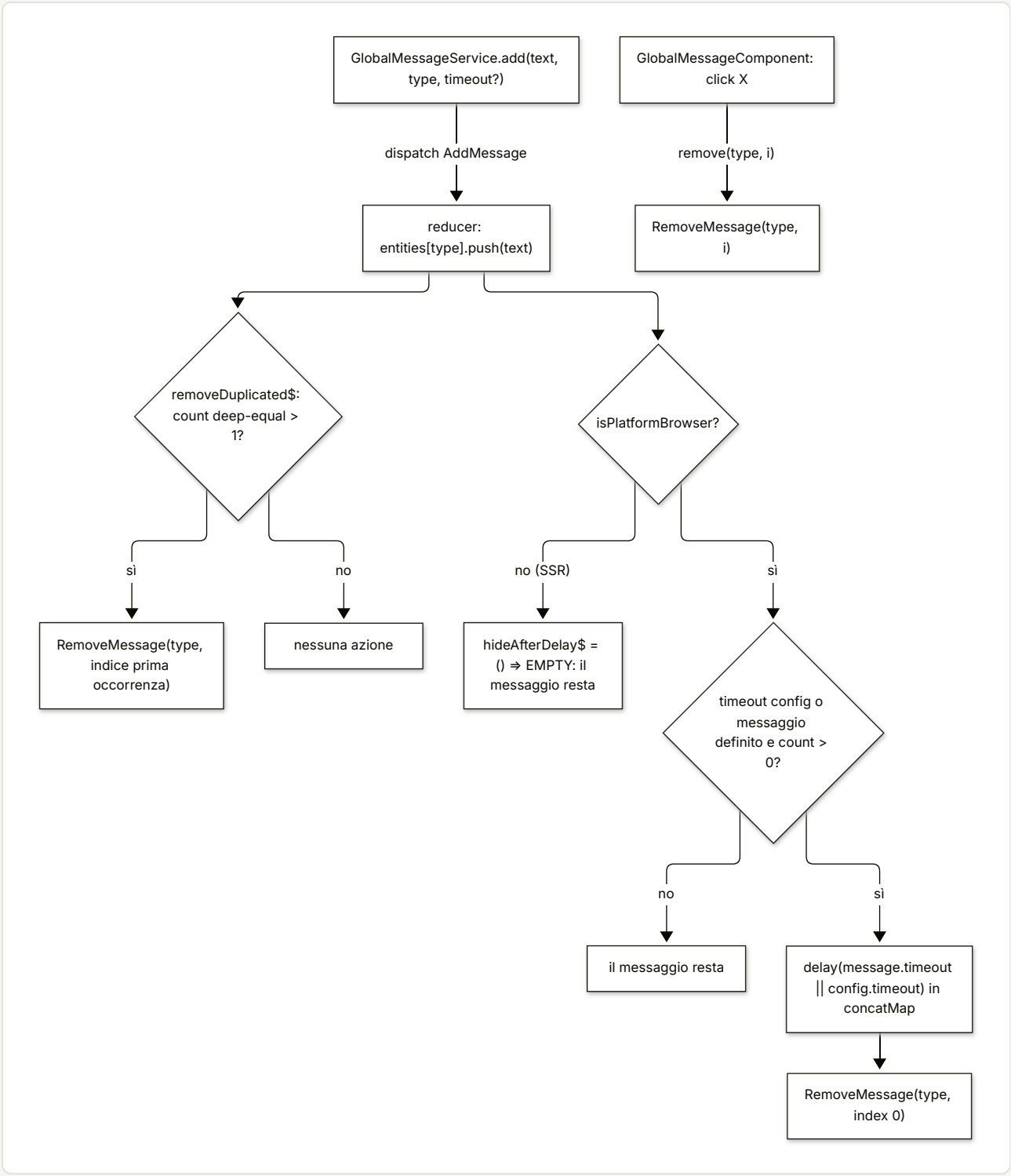
Ogni freccia corrisponde a un passo di §3.1 (passi 1-12). L'ordine di registrazione, per `reduceRight`, diventa ordine di annidamento: `HttpErrorHandlerInterceptor` è il più esterno, gli interceptor auth i più interni (`angular:packages/common/http/src/interceptor.ts:L232-L240`).

4.2 Un errore 500 che risale la catena: browser vs SSR

■ **TAVOLA 41.2** diagramma a tutta pagina, in fondo al capitolo

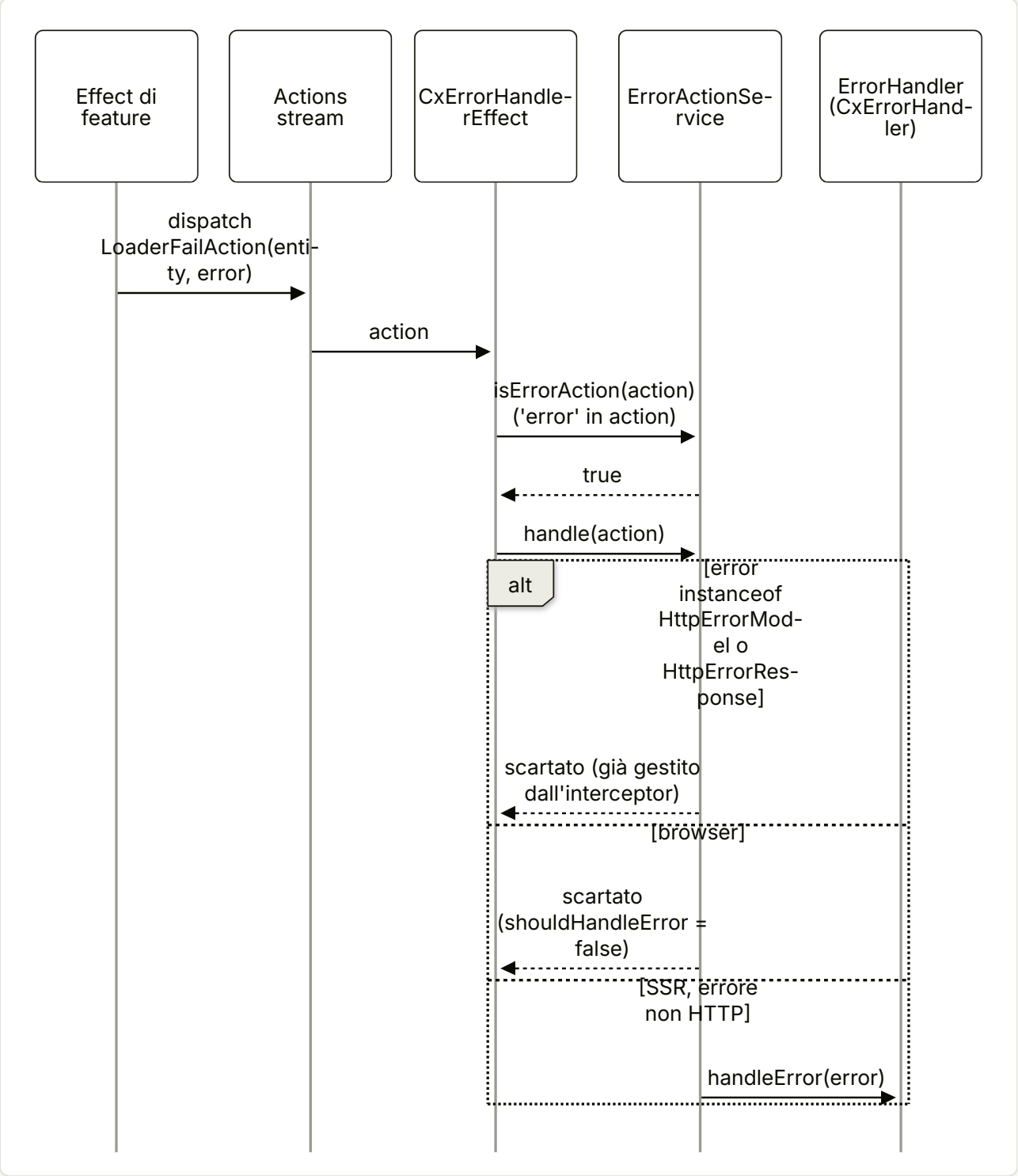
Riferimenti: §3.2 passi 2-9 e §3.3 passi 1-8.

4.3 Ciclo di vita di un global message nello store



Riferimenti: core-libs/core/src/global-message/facade/global-message.service.ts:L43-L72, core-libs/core/src/global-message/store/reducers/global-message.reducer.ts:L24-L67, core-libs/core/src/global-message/store/effects/global-message.effect.ts:L31-L106.

4.4 Fail action NgRx verso l'`ErrorHandler`



Riferimenti: §3.4.

5. Estratti di codice reale

5.1 ErrorHandlingModule

core-libs/core/src/error-handling/error-handling.module.ts:L13-L32

```ts
@NgModule({
  imports: [
    EffectsErrorHandlerModule.forRoot(),
    HttpErrorHandlerModule.forRoot(),
  ],
})
export class ErrorHandlingModule {
  static forRoot(): ModuleWithProviders<ErrorHandlingModule> {
    return {
      ngModule: ErrorHandlingModule,
      providers: [
        provideMultiErrorHandler(),
        {
          provide: ErrorHandler,
          useClass: CxErrorHandler,
        },
      ],
    };
  }
}
```

5.2 CxErrorHandler

core-libs/core/src/error-handling/cx-error-handler.ts:L18-L31

```ts
@Injectable()
export class CxErrorHandler implements ErrorHandler {
  protected errorHandlers = inject(MULTI_ERROR_HANDLER);

  /**
   * Error handler method. Handles the error by passing it to the registered multi error handlers.
   * @param error - The error to be handled.
   */
  handleError(error: unknown): void {
    this.errorHandlers.forEach((handler) => {
      handler.handleError(error);
    });
  }
}
```

5.3 `HttpErrorHandlerInterceptor` (cuore)

core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L40-L81

```ts
@Injectable({ providedIn: 'root' })
export class HttpErrorHandlerInterceptor implements HttpInterceptor {
  protected errorHandler = inject(ErrorHandler);
  protected occEndpointsService = inject(OccEndpointsService);
  protected windowRef = inject(WindowRef);

  protected userId = toSignal(inject(UserIdService).getUserId());

  intercept(
    request: HttpRequest<unknown>,
    next: HttpHandler
  ): Observable<HttpEvent<unknown>> {
    return next.handle(request).pipe(
      tap({
        error: (error: unknown) => {
          if (this.shouldHandleError(error)) {
            this.handleError(error);
          }
        },
      })
    );
  }
  // ...
  protected shouldHandleError(_error: unknown): boolean {
    return !this.windowRef.isBrowser();
  }

  protected handleError(error: unknown): void {
    error = this.isCmsPageNotFoundHttpError(error)
      ? new CmsPageNotFoundOutboundHttpError(error)
      : new OutboundHttpError(error);
    this.errorHandler.handleError(error);
  }
```

5.4 `HttpErrorInterceptor` (global message)

core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L22-L65

```ts
@Injectable({ providedIn: 'root' })
export class HttpErrorInterceptor implements HttpInterceptor {
  constructor(protected unifiedInjector: UnifiedInjector) {}

  protected handlers$: Observable<HttpErrorHandler[]> = this.unifiedInjector
    .getMulti(HttpErrorHandler)
    .pipe(shareReplay(1));
```

```
  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    return next.handle(request).pipe(
      tap({
        error: (response: any) => {
          if (response instanceof HttpErrorResponse) {
            this.handleErrorResponse(request, response);
          }
        },
      })
    );
  }

  protected handleErrorResponse(
    request: HttpRequest<any>,
    response: HttpErrorResponse
  ): void {
    const handler = this.getResponseHandler(response);
    if (handler) {
      handler.handleError(request, response);
    }
  }
  // ...
  protected getResponseHandler(
    response: HttpErrorResponse
  ): HttpErrorHandler | undefined {
    return resolveApplicable(getLastValueSync(this.handlers$) ?? [], [
      response,
    ]);
  }
```

5.5 `resolveApplicable`: la regola di scelta

core-libs/core/src/util/applicable.ts:L63-L87

```ts
export function resolveApplicable<T extends Applicable>(
  applicables: T[],
  matchParams: any[] = [],
  priorityParams: any[] = []
): T | undefined {
  const matchedApplicables = (applicables ?? []).filter(
    (applicable) => !applicable.hasMatch || applicable.hasMatch(...matchParams)
  );

  if (matchedApplicables.length < 2) {
```

```
    return matchedApplicables[0];
  }

  let lastPriority = -Infinity;
  return matchedApplicables.reduce((acc: T | undefined, curr: T) => {
    const currPriority = curr.getPriority
      ? curr.getPriority(...priorityParams)
      : Priority.NORMAL;
    if (lastPriority > currPriority) {
      return acc;
    }
    lastPriority = currPriority;
    return curr;
  }, undefined);
}
```

Nota la condizione `lastPriority > currPriority`: a parità (`===`) non si ritorna `acc`, si sostituisce con `curr`. È questo che realizza "provided later wins".

5.6 `HttpErrorHandler` (classe base)

core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L12-L44

TS

```
@Injectable({
  providedIn: 'root',
})
export abstract class HttpErrorHandler implements Applicable {
  constructor(
    protected globalMessageService: GlobalMessageService,
    @Inject(PLATFORM_ID) protected platformId?: Object
  ) {}

  /**
   * The http response status number which is handled by this handler.
   * Implementations can set the response status number, i.e. 404, so that
   * the handler can be found by the error interceptor.
   */
  responseStatus?: number;
  // ...
  abstract handleError(
    request: HttpRequest<any>,
    errorResponse: HttpErrorResponse
  ): void;
  // ...
  hasMatch(errorResponse: HttpErrorResponse): boolean {
    return errorResponse.status === this.responseStatus;
  }
```

5.7 UnknownErrorHandler

core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L22-L50

```ts
export class UnknownErrorHandler extends HttpErrorHandler {
  protected logger = inject(LoggerService);

  responseStatus = HttpResponseStatus.UNKNOWN;

  /**
   * hasMatch always returns true, to mach all errors
   */
  hasMatch(_errorResponse: HttpErrorResponse): boolean {
    return true;
  }

  handleError(_request: HttpRequest<any>, errorResponse: HttpErrorResponse) {
    // Error is already handled and logged by the `HttpErrorHandlerInterceptor`.
    // Log error here only in dev mode.
    if (isDevMode()) {
      this.logger.warn(
        `An unknown http error occurred\n`,
        errorResponse.message
      );
    }
  }

  /**
   * Fallback priority assures that the handler is used as a last resort
   */
  getPriority() {
    return Priority.FALLBACK;
  }
}
```

5.8 ForbiddenHandler

core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts:L20-L50

```ts
export class ForbiddenHandler extends HttpErrorHandler {
  responseStatus = HttpResponseStatus.FORBIDDEN;

  handleError(request: HttpRequest<any>) {
    if (
      request.url.endsWith(
        this.occEndpoints.buildUrl('user', {
          urlParams: { userId: 'current' },
        })
```

```
      )
    ) {
      this.authService.logout();
    }
    this.globalMessageService.add(
      { key: 'httpHandlers.forbidden' },
      GlobalMessageType.MSG_TYPE_ERROR
    );
  }

  getPriority(): Priority {
    return Priority.LOW;
  }

  constructor(
    protected globalMessageService: GlobalMessageService,
    protected authService: AuthService,
    protected occEndpoints: OccEndpointsService
  ) {
    super(globalMessageService);
  }
}
```

5.9 `BadCartRequestHandler`: come una feature specializza il 400

feature-libs/cart/base/core/http-interceptors/handlers/bad-cart-request.handler.ts:L21-L38

TS

```
export class BadCartRequestHandler extends HttpErrorHandler {
  responseStatus = HttpResponseStatus.BAD_REQUEST;

  getPriority(): Priority {
    return Priority.NORMAL;
  }

  hasMatch(errorResponse: HttpErrorResponse): boolean {
    return (
      super.hasMatch(errorResponse) &&
      this.getErrors(errorResponse).some(isCartError)
    );
  }

  handleError(request: HttpRequest<any>, response: HttpErrorResponse): void {
    this.handleCartNotFoundError(request, response);
    this.handleOtherCartErrors(request, response);
  }
```

5.10 `ErrorActionService`

core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L24-L43

```ts
handle(action: ErrorAction): void {
  const error: unknown = action.error;

  // Http errors are already handled in HttpErrorHandlerInterceptor.
  // To avoid duplicate errors we want to check if the error is not of type
  // HttpErrorModel or HttpErrorResponse.
  const isNotHttpError =
    !(error instanceof HttpErrorModel) &&
    !(error instanceof HttpErrorResponse);

  if (isNotHttpError && this.shouldHandleError(error)) {
    this.errorHandler.handleError(error);
  }
}

/** Here we want to filter which error actions should be handled.
 * By default, we check if action implements interface ErrorAction  */
isErrorAction(action: Action): action is ErrorAction {
  return 'error' in action;
}
```

5.11 `GlobalMessageService.add` / `remove`

core-libs/core/src/global-message/facade/global-message.service.ts:L43-L72

```ts
add(
  text: string | Translatable,
  type: GlobalMessageType,
  timeout?: number
): void {
  this.store.dispatch(
    new GlobalMessageActions.AddMessage({
      text: typeof text === 'string' ? { raw: text } : text,
      type,
      timeout,
    })
  );
}
// ...
remove(type: GlobalMessageType, index?: number): void {
  this.store.dispatch(
    index !== undefined
      ? new GlobalMessageActions.RemoveMessage({
          type: type,
          index: index,
```

```
        })
      : new GlobalMessageActions.RemoveMessagesByType(type)
  );
}
```

5.12 hideAfterDelay$

core-libs/core/src/global-message/store/effects/global-message.effect.ts:L70-L106

```ts
hideAfterDelay$:
  | Observable<GlobalMessageActions.RemoveMessage>
  | (() => Observable<never>) = createEffect(() =>
  isPlatformBrowser(this.platformId) // we don't want to run this logic when doing SSR
    ? this.actions$.pipe(
        ofType(GlobalMessageActions.ADD_MESSAGE),
        map((action) => (action as GlobalMessageActions.AddMessage).payload),
        concatMap((message) => {
          const config = this.config.globalMessages?.[message.type];
          return this.store.pipe(
            select(
              GlobalMessageSelectors.getGlobalMessageCountByType(message.type)
            ),
            take(1),
            filter(
              (count: number) =>
                ((config && config.timeout !== undefined) ||
                  message.timeout !== undefined) &&
                count > 0
            ),
            delay((message.timeout as number) || (config?.timeout as number)),
            switchMap(() =>
              of(
                new GlobalMessageActions.RemoveMessage({
                  type: message.type,
                  index: 0,
                })
              )
            )
          );
        })
      )
    : // workaround is required due to NGRX mutating a global static
      // observable EMPTY, causing to throw an error if we have
      // effect registered on the same observable twice
      () => EMPTY
);
```

5.13 LoggerService

core-libs/core/src/logger/logger.service.ts:L15-L30

```ts
@Injectable({
  providedIn: 'root',
})
export class LoggerService {
  log(...args: Parameters<typeof console.log>): void {
    /* eslint-disable-next-line no-console */
    console.log(...args);
  }
  warn(...args: Parameters<typeof console.warn>): void {
    /* eslint-disable-next-line no-console */
    console.warn(...args);
  }
  error(...args: Parameters<typeof console.error>): void {
    /* eslint-disable-next-line no-console */
    console.error(...args);
  }
  // ...
```

5.14 Il ponte verso Express

core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32

```ts
export const defaultExpressErrorHandlers =
  (documentContent: string): ErrorRequestHandler =>
  (err, _req, res, _next) => {
    if (!res.headersSent) {
      res.set('Cache-Control', 'no-store');
      const statusCode =
        err instanceof CmsPageNotFoundOutboundHttpError
          ? HttpResponseStatus.NOT_FOUND
          : HttpResponseStatus.INTERNAL_SERVER_ERROR;
      res.status(statusCode).send(documentContent);
    }
  };
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`globalMessages[MSG_TYPE_CONFIRMATION].timeout`	`number?` (ms)	3000	`core-libs/core/src/global-message/config/global-message-config.ts:L15-L27`; default `core-libs/core/src/global-message/config/default-global-message-config.ts:L12-L14`	`core-libs/core/src/global-message/store/effects/global-message.effect.ts:L78-L90`
`globalMessages[MSG_TYPE_INFO].timeout`	`number?`	3000	idem; default `core-libs/core/src/global-message/config/default-global-message-config.ts:L15-L17`	idem
`globalMessages[MSG_TYPE_ERROR].timeout`	`number?`	7000	idem; default `core-libs/core/src/global-message/config/default-global-message-config.ts:L18-L20`	idem
`globalMessages[MSG_TYPE_WARNING].timeout`	`number?`	7000	idem; default `core-libs/core/src/global-message/config/default-global-message-config.ts:L21-L23`	idem
`globalMessages[MSG_TYPE_ASSISTIVE].timeout`	`number?`	7000	idem; default `core-libs/core/src/global-message/config/default-global-message-config.ts:L24-L26`	idem
`timeout` del singolo messaggio (3° argomento di `GlobalMessageService.add`)	`number?`	`undefined`	`core-libs/core/src/global-message/facade/global-message.service.ts:L43-L54`	`core-libs/core/src/global-message/store/effects/global-message.effect.ts:L86-L90` (prevale se truthy)
feature toggle `enablePasswordExpiredErrorTranslation`	`boolean?`	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L174-L179`; default `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1031-L1031`	`core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L50-L57`
feature toggle `a11yFilteredFacetAnnouncement`	`boolean?`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L333-L333`; default `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1054-L1054`	`core-libs/storefront/cms-components/misc/global-message/global-message.component.html:L1-L24`: se attivo, i messaggi `MSG_TYPE_ASSISTIVE` vanno in un'unica live region `aria-live="polite"`; se disattivo, viene creata una live region per ogni messaggio

Note:

- `GlobalMessageTypeConfig` contiene **solo** `timeout` — core-libs/core/src/global-message/config/global-message-config.ts:L11-L13. `GlobalMessageConfig` è `providedIn: 'root'` con `useExisting: Config` e arricchisce `Config` via module augmentation — core-libs/core/src/global-message/config/global-message-config.ts:L15-L31. Si configura quindi con `provideConfig({ globalMessages: { ... } })` come qualsiasi altra chiave (vedi 11 — Configurazione).
- Il default è registrato con `provideDefaultConfig(defaultGlobalMessageConfig)` — core-libs/core/src/global-message/global-message.module.ts:L24-L24; `defaultGlobalMessageConfig` non è esportato dal barrel pubblico — core-libs/core/src/global-message/index.ts:L7-L16.
- Non esistono chiavi di configurazione per l'error handling (`MULTI_ERROR_HANDLER`, `shouldHandleError`, logger): si personalizza solo via DI (§8). I toggle in generale sono trattati in 42 — Feature toggle e deprecazioni.

---

7. Commenti e note notevoli dal sorgente

- "The CxErrorHandler is the default ErrorHandler for Spartacus. ... The prefix `Cx` was used to distinguish it from Angular's `ErrorHandler` class." — core-libs/core/src/error-handling/cx-error-handler.ts:L10-L17.
- "CAUTION: It MUST be provided as the first one in the application to be able to catch errors from all subsequent interceptors." — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L27-L39. Il contraltare in `BaseCoreModule`: "Import this module before any other interceptor to handle HTTP errors efficiently" — core-libs/core/src/base-core.module.ts:L28-L28.
- "Be default, we avoid sending unpredictable errors to the browser's console, to prevent possibly exposing there potentially confidential user's data. This isn't an issue in SSR, where pages are rendered anonymously." — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L63-L71 (testo identico, refuso "Be default" compreso, in core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L45-L53).
- "It ensures that HTTP errors, which are already handled by `HttpErrorHandlerInterceptor`, are not processed again to avoid duplicate error handling." — core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L14-L18.
- "Multi provided error handlers will be called in the order they are provided." — core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L20-L23.
- "Intentional empty error handler to avoid we fallabck to the unknown error handler" — core-libs/core/src/global-message/http-interceptors/handlers/not-found/not-found.handler.ts:L19-L19.
- "Error is already handled and logged by the `HttpErrorHandlerInterceptor`. Log error here only in dev mode." — core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L35-L36. Vero solo in SSR: vedi §9, trappola 1.
- "return the error handler that matches the `HttpResponseStatus` code. If no handler is available, the UNKNOWN handler is returned." — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L55-L58.
- "Updating email and changing password share same http error occurence." — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L76-L77.
- "Added in 3.2, depends on the injected `platformId`." (su `isSsr()`) — core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L63-L67.
- "workaround is required due to NGRX mutating a global static observable EMPTY, causing to throw an error if we have effect registered on the same observable twice" — core-libs/core/src/global-message/

store/effects/global-message.effect.ts:L102-L105.

- "It's a good extension point for customizing the destination of logs (e.g. to use a 3rd party logger library) or to enhance logs with more contextual data." — core-libs/core/src/logger/logger.service.ts:L9-L14.
- "Should be used with caution, as it's not a legitimate way for getting value from the observable." (su getLastValueSync) — core-libs/core/src/util/rxjs/get-last-value-sync.ts:L13-L15.
- "CXSPA-11104: Revise type to avoid redundant types - typescript:S6571" — core-libs/core/src/util/try-normalize-http-error.ts:L28-L28.
- "We need it until Angular implements a proper propagation of async errors ... https://github.com/angular/angular/issues/33642" — core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts:L18-L20.
- La proprietà error aggiunta "to be recognizable by CxErrorHandlerEffect (which expects error property in every fail action)" — feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L83-L87.
- Nei 55 file dell'area non ci sono @deprecated, TODO o FIXME (dossier, sezione N).

---

8. Extension point

COSA VUOI FARE	EXTENSION POINT	DEFINIZIONE	PUBBLICO?
Messaggio custom per uno status o per un sotto-caso di uno status	nuovo HttpErrorHandler multi-provider con getPriority() > LOW e/o hasMatch più stretto	core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L15-L74	sì, core-libs/core/src/global-message/index.ts:L10-L10
Modificare un handler esistente	estendere BadRequestHandler, ForbiddenHandler, … e registrarlo con priorità maggiore	core-libs/core/src/global-message/http-interceptors/handlers/index.ts:L7-L15	sì
Mostrare/rimuovere messaggi da codice applicativo	GlobalMessageService	core-libs/core/src/global-message/facade/global-message.service.ts:L21-L73	sì, core-libs/core/src/global-message/index.ts:L8-L8
Cambiare i timeout	GlobalMessageConfig via provideConfig	core-libs/core/src/global-message/config/global-message-config.ts:L15-L27	sì, core-libs/core/src/global-message/index.ts:L7-L7
Leggere lo store direttamente	GlobalMessageSelectors, GlobalMessageActions, GLOBAL_MESSAGE_FEATURE	core-libs/core/src/global-message/store/selectors/index.ts:L7-L8, core-libs/core/src/global-message/store/actions/index.ts:L7-L8	sì, core-libs/core/src/global-message/index.ts:L14-L16
Inviare gli errori a un servizio esterno (Sentry, APM…)	nuovo MultiErrorHandler su MULTI_ERROR_HANDLER (multi: true)	core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L16-L27	sì, core-libs/core/src/error-handling/multi-error-handler/index.ts:L8-L8
Cambiare destinazione/formato dei log	sostituire LoggerService (useClass/useExisting)	core-libs/core/src/logger/logger.service.ts:L15-L39	sì, core-libs/core/src/logger/index.ts:L7-L7

COSA VUOI FARE	EXTENSION POINT	DEFINIZIONE	PUBBLICO?
Inoltrare errori HTTP all'`ErrorHandler` anche nel browser, o filtrarli	estendere `HttpErrorHandlerInterceptor` e sovrascrivere `shouldHandleError` (protected)	core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L72-L98	sì, core-libs/core/src/error-handling/http-error-handler/index.ts:L7-L7
Decidere quali fail action contano come errore	estendere `ErrorActionService` (`isErrorAction`, `shouldHandleError`)	core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L41-L56	sì, core-libs/core/src/error-handling/effects-error-handler/index.ts:L9-L9
Far riconoscere una fail action custom	implementare `ErrorAction` (proprietà `error`)	core-libs/core/src/error-handling/effects-error-handler/error-action.ts:L9-L11	sì
Sostituire del tutto l'`ErrorHandler`	`{ provide: ErrorHandler, useClass: ... }`	core-libs/core/src/error-handling/error-handling.module.ts:L25-L28	`CxErrorHandler` pubblico, core-libs/core/src/error-handling/index.ts:L7-L7
Distinguere errori lato server	`OutboundHttpError`, `CmsPageNotFoundOutboundHttpError` (`instanceof`)	core-libs/core/src/error-handling/http-error-handler/outbound-http-error.ts:L10-L25	sì
Silenziare il logger nei test	`LoggerTestingModule` / `MockLoggerService`	core-libs/core/src/logger/testing/logger-testing.module.ts:L11-L14	sì, core-libs/core/src/logger/index.ts:L8-L8

Non sono extension point pubblici: `HttpErrorInterceptor` (importato ma non riesportato da core-libs/core/src/global-message/http-interceptors/index.ts:L20-L20), `CxErrorHandlerEffect` (core-libs/core/src/error-handling/effects-error-handler/index.ts:L7-L9), `GlobalMessageEffect`, il reducer, `defaultGlobalMessageConfig` (core-libs/core/src/global-message/index.ts:L7-L16) e `provideMultiErrorHandler` (core-libs/core/src/error-handling/multi-error-handler/index.ts:L7-L8). Vedi 54 — API pubblica vs interna.

8.1 Schema: handler custom per uno status

Esempio illustrativo (nomi applicativi inventati, simboli Spartacus esportati da `@spartacus/core`), costruito sullo stesso schema di `BadCartRequestHandler` (§5.9) e della sua registrazione (feature-libs/cart/base/core/cart-base-core.module.ts:L37-L41):

```ts
// ESEMPIO — non presente nel repository
@Injectable({ providedIn: 'root' })
export class MyNotFoundHandler extends HttpErrorHandler {
  responseStatus = HttpResponseStatus.NOT_FOUND;

  hasMatch(errorResponse: HttpErrorResponse): boolean {
    return super.hasMatch(errorResponse) && /* condizione specifica */ true;
  }

  handleError(_request: HttpRequest<any>, _response: HttpErrorResponse): void {
    this.globalMessageService.add(
```

```
      { key: 'myApp.notFound' },
      GlobalMessageType.MSG_TYPE_WARNING
    );
  }

  getPriority(): Priority {
    return Priority.NORMAL; // > LOW del NotFoundHandler di default
  }
}

// providers: [{ provide: HttpErrorHandler, useExisting: MyNotFoundHandler, multi: true }]
```

Vincoli verificati sul codice: il provider deve essere `multi: true`, altrimenti `UnifiedInjector.getMulti` lancia "Multi-providers mixed with single providers" (core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93); se l'handler vive in un modulo lazy viene visto solo dopo il caricamento di quel modulo (core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35).

8.2 Schema: multi error handler verso un servizio esterno

TS

```
// ESEMPIO — non presente nel repository
@Injectable({ providedIn: 'root' })
export class ApmErrorHandler implements MultiErrorHandler {
  handleError(error: unknown): void {
    // inviare error al servizio esterno; non lanciare mai da qui
  }
}

// providers: [{ provide: MULTI_ERROR_HANDLER, useExisting: ApmErrorHandler, multi: true }]
```

Ricorda che i default di Spartacus inoltrano a `ErrorHandler` errori HTTP e fail action **solo in SSR** (§3.3, §3.4); per raccoglierli anche nel browser serve sovrascrivere `shouldHandleError` in `HttpErrorHandlerInterceptor` / `ErrorActionService` — tenendo conto della motivazione di privacy del commento originale (§7).

Ricette correlate in Parte III:

- 50 — Ricette: configurazione, DI, interceptor, eventi: override di servizi via DI (`LoggerService`, `ErrorActionService`, `HttpErrorHandlerInterceptor`), multi-provider, interceptor custom e loro posizione nella catena.
- 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni: override delle chiavi `httpHandlers.*` e aggiunta di chiavi per handler custom.
- 54 — API pubblica vs interna, upgrade, anti-pattern, checklist: perché non estendere `HttpErrorInterceptor` o `CxErrorHandlerEffect` (non esportati).

9. Trappole, casi limite, comportamenti controintuitivi

1. **Nel browser il** `LoggingErrorHandler` **non vede errori HTTP né fail action.** Sia `HttpErrorHandlerInterceptor` sia `ErrorActionService` filtrano con `!isBrowser()`; il commento dell'`UnknownErrorHandler` ("already handled and logged by the HttpErrorHandlerInterceptor") è vero solo in SSR. Nel browser un errore di rete/timeout non lascia traccia in produzione — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L72-L74, core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L54-L56, core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L35-L42.
2. **Un solo handler per errore.** `HttpErrorInterceptor` invoca solo l'handler con priorità più alta: un handler custom con priorità maggiore **sostituisce** il default per quella risposta, non si somma — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L45-L53, core-libs/core/src/util/applicable.ts:L76-L86.
3. **Priorità di default bassa.** Tutti gli handler di default sono LOW (Unknown FALLBACK): un handler custom senza `getPriority` vale NORMAL e vince; a parità vince il provider registrato più tardi, anche se in un modulo lazy — core-libs/core/src/util/applicable.ts:L54-L57, core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93. Attenzione però: `getPriority` è dichiarato `abstract getPriority?()` nella base, quindi è opzionale anche per TypeScript — core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L61-L61.
4. `hasMatch` **troppo largo = handler che "mangia" tutto.** Se sovrascrivi `hasMatch` restituendo `true` (come `UnknownErrorHandler`) con priorità > FALLBACK, intercetti ogni status — core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L30-L32.
5. **Ordine degli interceptor.** `ErrorHandlingModule` deve restare il primo import con interceptor: gli errori lanciati da un interceptor registrato prima (quindi più esterno) non attraversano `HttpErrorHandlerInterceptor` e non arrivano all'`ErrorHandler`. Viceversa, un interceptor registrato dopo (più interno) che cattura o sostituisce l'errore gli consegna un errore già trasformato, o nessuno — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L37-L38, core-libs/core/src/base-core.module.ts:L28-L28, angular:packages/common/http/src/interceptor.ts:L233-L240.
6. **Il 401 non ha handler.** `UNAUTHORIZED` è nell'enum ma non tra i registrati; se un 401 sfugge all'auth interceptor cade nell'`UnknownErrorHandler` (nessun messaggio) — core-libs/core/src/global-message/models/response-status.model.ts:L10-L10, core-libs/core/src/global-message/http-interceptors/index.ts:L22-L63.
7. **Timeout e rete sono silenziosi.** Status 0 → `UnknownErrorHandler` → solo `warn` in dev mode; la chiave `httpHandlers.unknownError` esiste nelle traduzioni ma questo handler non la usa — core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L34-L43, core-libs/assets/src/translations/en/common.json:L131-L131.
8. `HttpErrorInterceptor` **ignora errori non HTTP**, mentre `HttpErrorHandlerInterceptor` li inoltra tutti — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L37-L39, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L54-L58.
9. **Il 404 è volutamente muto** (`NotFoundHandler` vuoto): per le pagine CMS il 404 è gestito dal `CmsPageGuardService`, che carica la pagina CMS `notFound` e la associa al contesto richiesto (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L112-L127), e in SSR diventa un 404 HTTP via `CmsPageNotFoundOutboundHttpError` — core-libs/core/src/global-message/http-interceptors/handlers/not-found/not-found.handler.ts:L18-L20, core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L26-L29.

10. **403 e logout.** `ForbiddenHandler` fa `logout()` solo se l'URL termina con quello OCC di `users/current`; per altri 403 mostra solo il messaggio — core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts:L24-L32.
11. **Un 400 può produrre molti messaggi.** I sei sotto-handler di `BadRequestHandler` girano tutti, uno `add` per ogni `ErrorModel` corrispondente — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L26-L33.
12. **Testo backend non tradotto.** `IllegalArgumentError` e `UnknownIdentifierError` (con message) mostrano `error.message` raw del backend — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L106-L118, core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L143-L152.
13. **Presupposti sul formato della richiesta.** `handleBadPassword` chiama `request.body?.get('grant_type')` (presuppone `HttpParams`), `handleBadLoginResponse` usa `new URL(_request.url)` — core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L42-L42, core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L78-L78. DA VERIFICARE — con un URL relativo (es. `occ.baseUrl` vuoto) il costruttore `URL` senza base lancerebbe; il caso non è coperto da test nel dossier.
14. `hideAfterDelay$` **rimuove l'indice 0, non "quel" messaggio.** Se l'utente chiude manualmente un messaggio, il timer successivo rimuoverà un altro messaggio dello stesso tipo — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L91-L97.
15. **Timer serializzati.** `concatMap` accoda i timer: con tre errori, il secondo inizia a contare solo dopo la scadenza del primo — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L77-L77.
16. `timeout: 0` **non significa "subito".** Supera il filtro (`!== undefined`) ma `delay(0 || config.timeout)` usa il timeout di config — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L86-L90.
17. **Il** `timeout` **non arriva allo store**: il reducer salva solo `text`, quindi non puoi leggerlo da un selettore — core-libs/core/src/global-message/store/reducers/global-message.reducer.ts:L32-L41.
18. **Dedup che sposta in fondo.** `removeDuplicated$` rimuove la prima occorrenza (la più vecchia), così il messaggio riappare in coda — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L52-L62. Il timer del primo `ADD_MESSAGE` resta però in coda in `concatMap` e, alla scadenza, rimuove l'indice 0. Nel caso di un singolo messaggio duplicato, il messaggio sparisce alla scadenza del timer originale; il timer del nuovo `ADD_MESSAGE` trova `count = 0` e non fa nulla — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L77-L97. Il confronto è deep-equal sul `Translatable`: stessa key con `params` diversi = due messaggi distinti — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L47-L51.
19. **In SSR i messaggi non scadono** (`hideAfterDelay$ = () => EMPTY`) — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L73-L105.
20. `RemoveMessagesByType` **lascia** `[]`, non elimina la chiave; `RemoveMessage` con indice fuori range è un no-op (`splice`) — core-libs/core/src/global-message/store/reducers/global-message.reducer.ts:L57-L58, core-libs/core/src/global-message/store/reducers/global-message.reducer.ts:L69-L73.
21. `isErrorAction` **è "duck typing" puro**: qualunque action con chiave `error` (anche `undefined`) è considerata d'errore — core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L41-L43. Viceversa una fail action **senza** `error` sfugge (vedi il commento in pickup-in-store, feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L83-L87).
22. **Override di** `isErrorAction` **e** `this`**.** L'effect passa il metodo non legato: `filter(this.errorActionService.isErrorAction)` — core-libs/core/src/error-handling/effects-error-handler/cx-error-handler.effect.ts:L25-L26; RxJS lo invoca con `predicate.call(thisArg, ...)` e `thisArg` assente — deps:rxjs-7.8.2/package/dist/cjs/internal/operators/filter.js:L6-L11. Un override che usi `this` (es. per iniettare

dipendenze) troverà `this === undefined`: usa una arrow function come proprietà o non dipendere da `this`.

23. **Fail action con `HttpErrorResponse` grezzo**: viene scartata comunque (`instanceof HttpErrorResponse`), anche se non normalizzata — `core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L30-L32`.
24. **Un multi handler che lancia blocca gli altri.** `CxErrorHandler` itera senza `try/catch` — `core-libs/core/src/error-handling/cx-error-handler.ts:L26-L30`. In SSR `LoggingErrorHandler` è registrato prima di `PropagatingToServerErrorHandler`; un handler custom aggiunto in mezzo che lancia impedirebbe la propagazione al server.
25. `provideMultiErrorHandler` **non è pubblico**: per aggiungere handler usa direttamente `MULTI_ERROR_HANDLER` con `multi: true` — `core-libs/core/src/error-handling/multi-error-handler/index.ts:L7-L8`.
26. `userId` **del signal.** `isCmsPageNotFoundHttpError` usa il valore corrente del signal `userId` per costruire l'URL atteso — `core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L46-L46`, `core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L90-L92`. `UserIdService._userId` è un `ReplaySubject<string>(1)` senza valore iniziale (`core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L25-L25`): finché nessuno chiama `setUserId` il signal vale `undefined` e l'URL atteso potrebbe non corrispondere. DA VERIFICARE — l'effetto concreto sul confronto URL non è verificato a runtime.
27. `isSsr()` **falso negativo.** Restituisce `false` se la sottoclasse non passa `platformId` a `super`, come fa `ForbiddenHandler` — `core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L68-L73`, `core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts:L48-L48`.
28. **Errori di effect vs fail action.** Un errore non catturato in un effect arriva a `CxErrorHandler` tramite il `defaultEffectsErrorHandler` di NgRx anche nel browser (e l'effect viene risottoscritto fino a 10 volte) — `deps:ngrx-effects-21.0.1/package/fesm2022/ngrx-effects.mjs:L174-L185`.
29. `getLastValueSync` **"funziona per costruzione"**: solo perché `injectors$` emette sincronamente il root injector (`startWith`) e `shareReplay(1)` memorizza l'ultimo array — `core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35`, `core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L26-L28`.

---

10. Come fare debug

- **Quale handler ha gestito l'errore?** Metti un breakpoint in `HttpErrorInterceptor.handleErrorResponse` e ispeziona `handler` (`core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L49-L52`). Per vedere *tutti* i candidati, ispeziona il risultato di `getLastValueSync(this.handlers$)` in `getResponseHandler` (`core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L62-L64`) e confronta `responseStatus` e `getPriority()` di ciascuno con le regole di `core-libs/core/src/util/applicable.ts:L63-L87`.
- **Il mio handler custom non viene usato.** Verifica: (a) `multi: true` sul provider (`core-libs/core/src/lazy-loading/unified-injector.ts:L83-L88`); (b) priorità > LOW o `hasMatch` corretto; (c) se è in un modulo lazy, che il modulo sia già caricato al momento dell'errore (`core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35`).
- **Nessun messaggio a video.** Status 0 (timeout/rete), 401 o 404 non producono messaggi di default (§9 punti 6, 7, 9). Con Redux DevTools cerca le action `[Global-message] Add a Message` (`core-libs/core/src/global-message/store/actions/global-message.actions.ts:L13-L16`): se ci sono, il problema è nel rende-

ring (`cx-global-message` deve essere nel layout, importato via `GlobalMessageComponentModule` da core-libs/storefront/base-storefront.module.ts:L26-L30) o nelle traduzioni mancanti.

- **Messaggi che spariscono "a caso".** Ricorda indice 0 e `concatMap` (§9 punti 14-15). Seguire nella timeline le coppie `Add a Message` / `Remove a Message` e il loro `index`.
- **Errori in dev mode.** `UnknownErrorHandler` scrive `An unknown http error occurred` via `LoggerService.warn` solo con `isDevMode()` (core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L37-L42); `tryNormalizeHttpError` scrive `Error passed to tryNormalizeHttpError is not HttpErrorResponse instance` se riceve un errore non HTTP (core-libs/core/src/util/try-normalize-http-error.ts:L59-L65).
- **SSR risponde 500 inatteso.** Cerca nei log server l'`Outbound HTTP Error` scritto da `LoggingErrorHandler` (core-libs/core/src/error-handling/multi-error-handler/logging-error-handler.ts:L23-L25) e ispeziona la sua `cause` (core-libs/core/src/error-handling/http-error-handler/outbound-http-error.ts:L10-L14): è la richiesta OCC fallita. Un errore non HTTP proveniente da una fail action arriva invece "nudo" (§3.4). Approfondimento in 39 — SSR ottimizzato.
- **Simulare il comportamento SSR nel browser.** Un `HttpErrorHandlerInterceptor` esteso con `shouldHandleError` che restituisce `true` fa passare gli errori HTTP da `CxErrorHandler` anche lato client (core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L72-L74) — solo in sviluppo, per la nota di privacy di §7.
- **Test unitari rumorosi.** Importa `LoggerTestingModule` (sostituisce `LoggerService` con `MockLoggerService`) — core-libs/core/src/logger/testing/logger-testing.module.ts:L11-L14. Vedi 44 — Strategia di test.

---

11. Domande di autoverifica

1. **Perché `ErrorHandlingModule.forRoot()` è il primo import di `BaseCoreModule`?** Perché registra `HttpErrorHandlerInterceptor`, che deve essere il più esterno per vedere gli errori di tutti gli interceptor successivi; Angular annida gli interceptor con `reduceRight`, quindi il primo registrato è l'ultimo a ricevere l'errore. — core-libs/core/src/base-core.module.ts:L26-L29, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L37-L38, angular:packages/common/http/src/interceptor.ts:L232-L240.
2. **Un backend risponde 500 nel browser. Quante volte viene chiamato `CxErrorHandler`?** Zero (per questo percorso): `HttpErrorInterceptor` mostra `httpHandlers.internalServerError`, ma `HttpErrorHandlerInterceptor.shouldHandleError` è `false` nel browser. — core-libs/core/src/global-message/http-interceptors/handlers/internal-server/internal-server.handler.ts:L16-L28, core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L72-L74.
3. **Registro un handler per il 400 senza `getPriority`. Chi vince fra il mio, `BadRequestHandler` e `UnknownErrorHandler`?** Il mio: senza `getPriority` vale NORMAL (0), che batte LOW (−10) e FALLBACK (−50). — core-libs/core/src/util/applicable.ts:L7-L24, core-libs/core/src/util/applicable.ts:L76-L86.
4. **Due handler con la stessa priorità e lo stesso `hasMatch`: quale viene scelto?** Quello registrato più tardi, perché il `reduce` sostituisce l'accumulatore quando `lastPriority <= currPriority`. — core-libs/core/src/util/applicable.ts:L54-L57, core-libs/core/src/util/applicable.ts:L81-L85.
5. **Perché un timeout HTTP non mostra alcun messaggio di default?** `HttpTimeoutInterceptor` produce un `HttpErrorResponse` con status 0; nessun handler ha quello status, quindi vince `UnknownErrorHandler`, che fa solo `warn` in dev mode. — core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L101-L110, core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L34-L43.

6. **In SSR, come fa il server a rispondere 404 invece di 500 per una pagina CMS inesistente?** `HttpErrorHandlerInterceptor` riconosce il 404 sull'endpoint `pages` e crea `CmsPageNotFoundOutboundHttpError`; `PropagatingToServerErrorHandler` lo propaga e `defaultExpressErrorHandlers` lo mappa su 404 con `instanceof`. — core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L76-L98, core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts:L35-L37, core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L26-L30.
7. **Una fail action porta un `HttpErrorModel`. Arriva all'`ErrorHandler` in SSR?** No: `ErrorActionService.handle` scarta `HttpErrorModel` e `HttpErrorResponse` perché già gestiti dall'interceptor, evitando il doppio log. — core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L27-L36.
8. **Chiamo `globalMessageService.add('Salvato', MSG_TYPE_CONFIRMATION, 0)`. Dopo quanto sparisce?** Dopo 3000 ms: `timeout: 0` passa il filtro `!== undefined`, ma `delay(0 || config.timeout)` usa il default di config per CONFIRMATION. Inoltre la stringa diventa `{ raw: 'Salvato' }`. — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L84-L90, core-libs/core/src/global-message/config/default-global-message-config.ts:L12-L14, core-libs/core/src/global-message/facade/global-message.service.ts:L50-L50.
9. **Come aggiungo l'invio degli errori a un servizio esterno senza perdere il logging di default?** Registrando un `MultiErrorHandler` su `MULTI_ERROR_HANDLER` con `multi: true` (il token è pubblico; `provideMultiErrorHandler` no); `CxErrorHandler` chiamerà tutti gli handler in ordine. — core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L16-L27, core-libs/core/src/error-handling/cx-error-handler.ts:L26-L30, core-libs/core/src/error-handling/multi-error-handler/index.ts:L7-L8.
10. **Perché un 403 su `users/current` causa un logout, ma un 403 su un altro endpoint no?** `ForbiddenHandler` confronta `request.url` con `occEndpoints.buildUrl('user', { urlParams: { userId: 'current' } })` tramite `endsWith` e chiama `authService.logout()` solo in quel caso; il messaggio `httpHandlers.forbidden` è mostrato sempre. — core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts:L23-L37.

TAVOLA 41.1 **4.1 Registrazione e ordine della catena di interceptor**

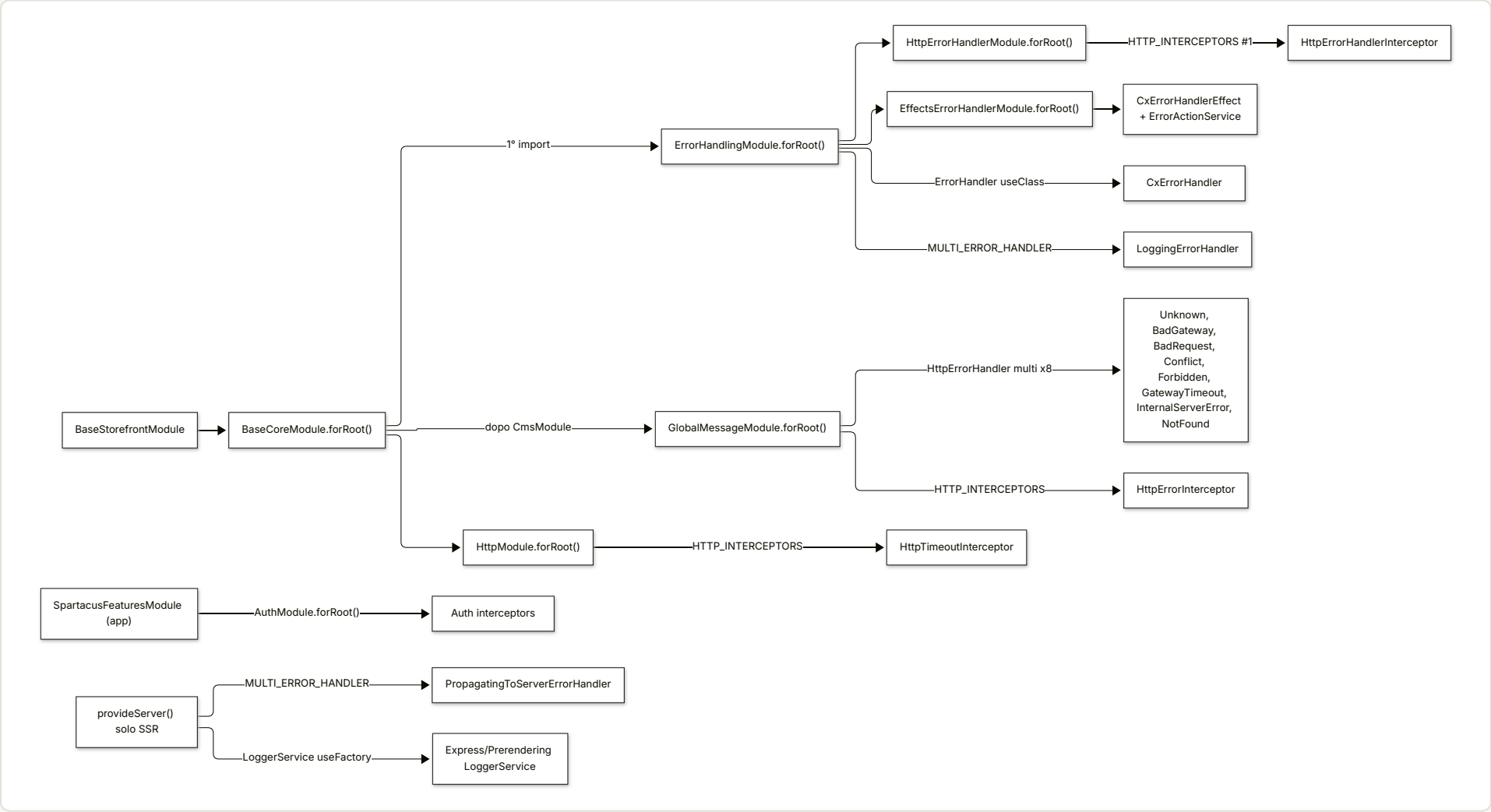
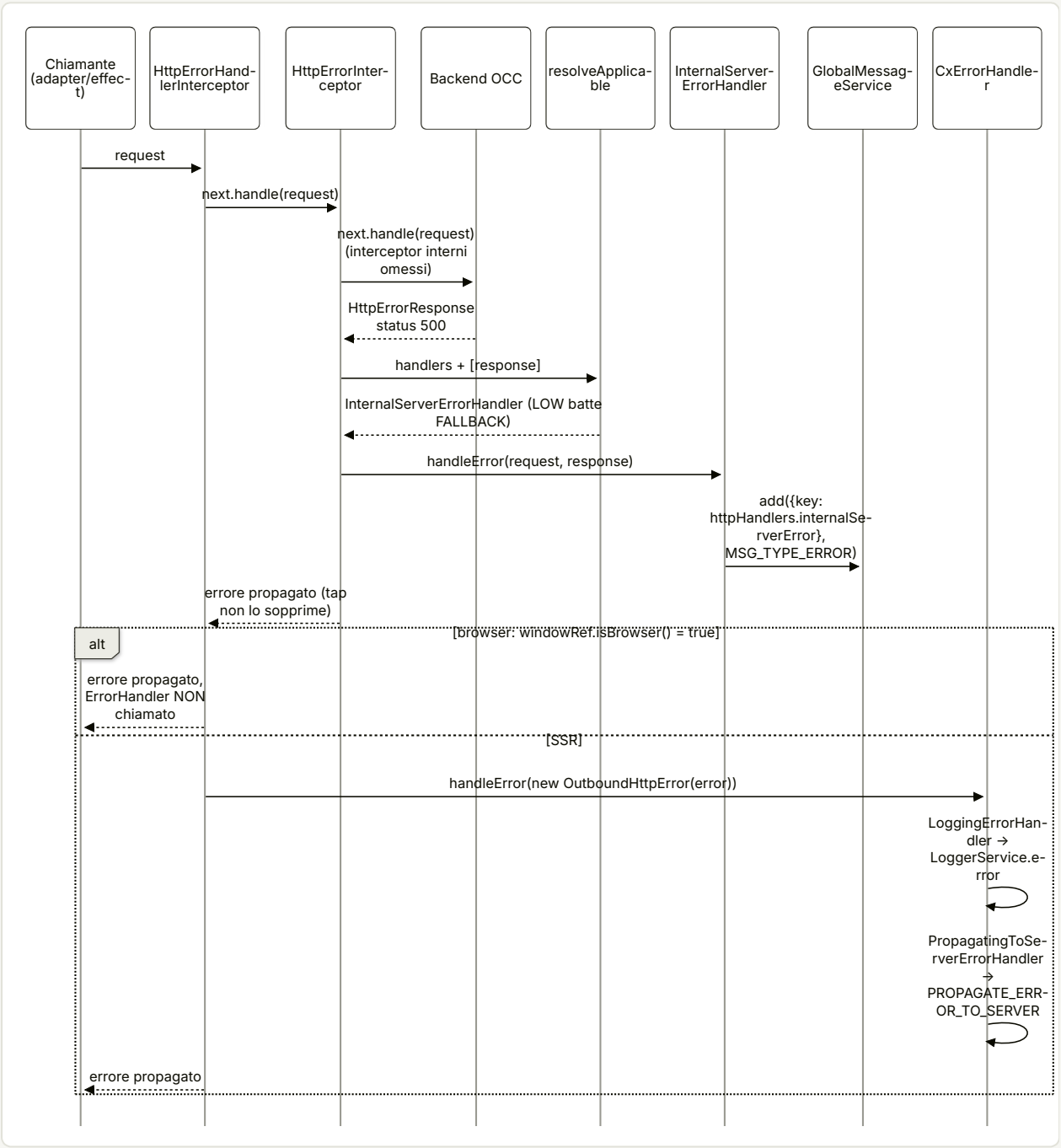


TAVOLA 41.2 **4.2 Un errore 500 che risale la catena: browser vs SSR**



42

Feature toggle e deprecazioni

Correzioni protette da flag con nome, per non rompere le estensioni a valle. Gestione delle deprecazioni.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Spartacus è una libreria che i clienti estendono per ereditarietà, sostituzione via DI e override SCSS. Ogni correzione che cambia DOM, stili o comportamento rischia quindi di rompere un'estensione a valle. La risposta del progetto è un meccanismo di **feature toggle**: la correzione viene scritta accanto al vecchio comportamento, protetta da un flag booleano con nome, e il cliente la attiva quando è pronto. Lo dice la JSDoc del token principale: "Properties for gradual adaptation to breaking changes in Spartacus via feature toggles. Each toggle has its own lifespan defined in the official docs." (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L10-L14).

Allo SHA convivono **due generazioni** di API:

GENERAZIONE	SIMBOLI	STATO	DOVE LEGGE
Legacy	`FeaturesConfig` (`features.*` dentro il `Config` globale), `FeatureConfigService.isEnabled/isLevel`, `features.level`, `FeatureLevelDirective`	`@deprecated` (core-libs/core/src/features-config/config/features-config.ts:L10-L26, core-libs/core/src/features-config/services/feature-config.service.ts:L15-L22)	`Config.features`, costruito con `deepMerge`
Attuale	`FeatureToggles` (classe astratta DI + interfaccia), `provideFeatureToggles(...)`, `provideDefaultFeatureToggles(...)`	raccomandata (core-libs/core/src/features-config/config/features-config.ts:L10-L26)	oggetto piatto costruito con `Object.assign` (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L15-L24)

Le due generazioni sono collegate da un **ponte a senso unico** (toggle → config) installato da `FeaturesConfigModule.forRoot()` "for backward compatibility with the deprecated FeaturesConfig" (core-libs/core/src/features-config/features-config.module.ts:L28-L52). La direttiva strutturale `*cxFeature`, che nei template è il modo più diffuso di leggere un toggle, passa ancora dal servizio legacy (core-libs/core/src/features-config/directives/feature.directive.ts:L11-L17): per questo il ponte non si può togliere.

Il capitolo copre anche il lato **stili**: un toggle può attivare regole SCSS tramite il mixin `forFeature()` e la funzione TS `useFeatureStyles()`, che insieme fanno comparire classi `cxFeat_<chiave>` sull'elemento root (core-libs/core/src/features-config/services/feature-styles.service.ts:L17-L26, core-libs/styles/scss/_features.scss:L41-L62). Il vecchio approccio per versione, `forVersion()`, è deprecato (core-libs/styles/scss/_versioning.scss:L1-L2).

Infine si guarda alla **politica di deprecazione** come emerge dal codice: tag `@deprecated` (165 occorrenze nelle librerie, tabella completa in A10 — API deprecate), interfaccia dei toggle volutamente *strict* (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11), TODO "Remove feature flag next major release." e regole ESLint dedicate.

L'elenco delle 107 chiavi, con default, descrizione e conteggi d'uso, è in A09 — Feature toggle. Questo capitolo spiega il meccanismo; l'appendice è il catalogo.

> Nota sul focus dell'OUTLINE: una **pipe** dedicata ai feature toggle allo SHA **non esiste**. Allo SHA nessuna `@Pipe` legge i feature toggle: le pipe del repo (es. `cxTranslate`, `cxUrl`, `cxDate`, `cxArgs`, `formatTimer`, `cxMultiLine`) non hanno a che fare con `FeatureToggles` (core-libs/core/src/i18n/translate.pipe.ts:L25-L28, core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts:L11-L11,

> core-libs/core/src/i18n/date.pipe.ts:L13-L13, feature-libs/asm/core/utils/args/args.pipe.ts:L33-L33).

Capitoli collegati: la meccanica di `Config`/`ConfigChunk`/`deepMerge` è in 11 — Configurazione; i token `providedIn: 'root'` e i multi provider in 01 — Dependency injection; gli `APP_BOOTSTRAP_LISTENER` nel contesto del bootstrap in 10 — Bootstrap e inizializzazione; gli stili in generale in 38 — Stili e theming; le migrazioni in 43 — Schematics e migrazioni; il test tooling in 44 — Test tooling.

---

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/features-config/config/features-config.ts	classe astratta deprecata `FeaturesConfig` (`useExisting: Config`) + module augmentation di `Config`; `level`, index signature, `consignmentTracking`, `disableConfigUpdates`
core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts	`FeatureTogglesInterface` (107 chiavi, strict) e `defaultFeatureToggles` (`Required<...>`): **API privata**, non esportata
core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts	`FeatureToggles` (classe+interfaccia), `FeatureToggleKey`, `FeatureToggleExpression`, `DefaultFeatureToggles`, `RootFeatureToggles`, `FeatureTogglesChunk`, `DefaultFeatureTogglesChunk`
core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts	`provideFeatureToggles`, `provideFeatureTogglesFactory`, `provideDefaultFeatureToggles`, `provideDefaultFeatureTogglesFactory`, tipo `FeatureTogglesFactory`
core-libs/core/src/features-config/feature-toggles/populate-feature-toggles-to-features-config.ts	ponte retrocompatibile toggle → `Config.features` (livello root e default)
core-libs/core/src/features-config/feature-toggles/index.ts	barrel pubblico, esclude `config/feature-toggles`
core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts	`MockFeatureTogglesController`, `provideMockFeatureToggles`, `MockFeatureConfigService` (privato)
core-libs/core/src/features-config/feature-toggles/testing/index.ts	barrel del testing (non pubblicato)
core-libs/core/src/features-config/features-config.module.ts	`FeaturesConfigModule` e `forRoot(defaultLevel='3.0')`: default toggle, ponte, livello, `APP_BOOTSTRAP_LISTENER`
core-libs/core/src/features-config/services/feature-config.service.ts	`FeatureConfigService` deprecato (`isLevel`, `isEnabled`)
core-libs/core/src/features-config/services/feature-styles.service.ts	`FeatureStylesService`: contatore d'uso → classi `cxFeat_*` sull'elemento root
core-libs/core/src/features-config/utils/feature-config-utils.ts	`isFeatureEnabled`, `isFeatureLevel` (deprecata), helper privati `isFeatureConfig`, `isInLevel`
core-libs/core/src/features-config/utils/use-feature-styles.ts	`useFeatureStyles(feature)` (injection context + `DestroyRef`)
core-libs/core/src/features-config/directives/feature.directive.ts	`FeatureDirective` `[cxFeature]`
core-libs/core/src/features-config/directives/feature-level.directive.ts	`FeatureLevelDirective` `[cxFeatureLevel]`, deprecata e inutilizzata

PATH	RUOLO
core-libs/core/src/features-config/index.ts	barrel della cartella, riesportato da core-libs/core/public_api.ts:L19-L19
core-libs/core/src/config/config-tokens.ts	Config, RootConfig, DefaultConfig (deepMerge dei chunk)
core-libs/core/src/config/config-providers.ts	provideConfigFactory, provideDefaultConfigFactory usati dal ponte
core-libs/core/src/config/services/configuration.service.ts	legge disableConfigUpdates
core-libs/core/src/base-core.module.ts	importa FeaturesConfigModule.forRoot()
core-libs/styles/scss/_features.scss	mixin forFeature, funzione getFeatureCssClass, prefisso cxFeat_
core-libs/styles/scss/_versioning.scss	mixin deprecato forVersion, $styleVersion, $useLatestStyles
core-libs/styles/scss/root.scss	unico uso di produzione di forFeature(..., 'currentSelector')
core-libs/storefront/layout/main/storefront.component.ts	registra sei stili di feature globali; aggiunge a mano una classe su <html>
core-libs/core/src/config/test-config.module.ts	toggle ad alta priorità da TEST_CONFIG
core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts	ng add genera provideFeatureToggles con tutte le chiavi a true
tools/eslint-rules/rules/use-provide-default-feature-toggles.ts, tools/eslint-rules/rules/feature-toggles-must-be-private.ts, tools/eslint-rules/rules/no-storefrontapp-false-feature-toggles.ts	regole di lint sui toggle

3. Come funziona passo per passo

3.1 Il catalogo: `FeatureTogglesInterface` e `defaultFeatureToggles`

Tutte le chiavi stanno in un solo file. L'interfaccia apre con un commento programmatico:

- niente index signature, così che una chiave rimossa provochi un errore di compilazione nel codice cliente: "customers using a property that was recently removed, will know they have to adapt their code" (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11);
- l'interfaccia va da L11 a L1011 e dichiara 107 proprietà `?: boolean` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L11-L1011);
- `defaultFeatureToggles` è tipato `Required<FeatureTogglesInterface>`: il compilatore obbliga a dare un default a ogni chiave. Allo SHA sono 32 `true` e 75 `false` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1013-L1121);
- l'ordine delle chiavi nei default non coincide con quello dell'interfaccia (es. `alignNavigationMenuWithHeader` sta a L469 nell'interfaccia e a L1016 nei default: core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L469-L469, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1016-L1016).

Le JSDoc seguono convenzioni ricorrenti: una descrizione, spesso `Affects: <Classi>`, note `NOTE:` sulle dipendenze esterne (es. `authorizationCodeFlowByDefault` richiede un Authorization Server CCv2 "September 2025 update or higher": core-libs/core/src/features-config/feature-toggles/config/feature-toggles.

ts:L153-L161) o da altri toggle (`authorizationCodeFlowByDefaultCsrfTokenRefresh` "Only applies when `authorizationCodeFlowByDefault` is also enabled": core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L163-L172). Alcune elencano anche i file SCSS toccati (`a11yHighContrastFix`: core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L995-L1010). Non tutti i toggle riguardano la UI: `enableCartSlowNetworkResilience` attiva i fix CXSPA-10582 sul carrello e descrive cosa torna "pre-fix" quando è OFF (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L554-L583).

Il barrel `feature-toggles/index.ts` **non** esporta questo file: "don't export from './config/feature-toggles' as it contains private API" (core-libs/core/src/features-config/feature-toggles/index.ts:L7-L9). Dall'esterno si usa il tipo `FeatureToggles`, che estende l'interfaccia.

3.2 I token: due livelli, quattro token

`feature-toggles-tokens.ts` definisce una piramide a due livelli, **default** (librerie) e **root** (applicazione):

1. `DefaultFeatureTogglesChunk` e `FeatureTogglesChunk` sono `InjectionToken<FeatureToggles[]>` `providedIn: 'root'` con factory `() => []`, alimentati come multi provider (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L60-L78). La JSDoc del primo fissa la regola: "all FeatureToggles provided in libraries should be provided as `default`" (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L69-L78).
2. `DefaultFeatureToggles` unisce i chunk default con `Object.assign({}, ...chunks)`, con fallback `[]` (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L33-L43).
3. `RootFeatureToggles` fa lo stesso con i `FeatureTogglesChunk` (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L48-L58). Se due chunk hanno la stessa chiave, "last provided wins" (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.spec.ts:L58-L58).
4. `FeatureToggles` è una classe astratta `providedIn: 'root'` con `useFactory` che restituisce `Object.assign({}, inject(DefaultFeatureToggles), inject(RootFeatureToggles))`: i valori root vincono sui default, e il merge è **shallow** (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L15-L24).
5. La riga `export interface FeatureToggles extends FeatureTogglesInterface {}` usa il declaration merging di TypeScript: la classe astratta, che serve da token DI, riceve le 107 proprietà tipate (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L25-L25).
6. `FeatureToggleKey = keyof FeatureToggles` e `FeatureToggleExpression = FeatureToggleKey | \!${FeatureToggleKey}`: la negazione è un template literal type, quindi '!chiavelnesistente' non compila (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L27-L28`).

3.3 I provider

`feature-toggles-providers.ts` espone quattro helper simmetrici:

- `provideFeatureToggles(toggles = {})` → `ValueProvider` multi su `FeatureTogglesChunk`. La JSDoc dice che va fornito "in the root injector (in an eagerly loaded module)" e rimanda le librerie a `provideDefaultFeatureToggles()` (core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L16-L33).
- `provideFeatureTogglesFactory(factory)` → `FactoryProvider` multi sullo stesso token, senza `deps`: la factory usa `inject()` (core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L44-L52). Il tipo della factory è `(...props: any[]) => FeatureToggles` (core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L14-L14).

- `provideDefaultFeatureToggles` / `provideDefaultFeatureTogglesFactory` → stessi provider su `DefaultFeatureTogglesChunk`, con "CAUTION: It should be provided only in the root injector" (core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L54-L86).

3.4 `FeaturesConfigModule.forRoot()` e il ponte verso `Config.features`

`BaseCoreModule` importa `FeaturesConfigModule.forRoot()` senza argomenti, dopo `ProcessModule.forRoot()` e prima di `SiteContextModule.forRoot()` (core-libs/core/src/base-core.module.ts:L36-L38). `forRoot(defaultLevel = '3.0')` registra quattro cose (core-libs/core/src/features-config/features-config.module.ts:L28-L52):

1. `provideDefaultFeatureToggles(defaultFeatureToggles)`: il catalogo diventa il primo chunk default;
2. `...populateFeatureTogglesToFeaturesConfig`: il ponte;
3. `provideDefaultConfig({ features: { level: defaultLevel || '*' } })`: il livello legacy;
4. un `APP_BOOTSTRAP_LISTENER` che chiama `FeatureStylesService.init(compRef)` (core-libs/core/src/features-config/features-config.module.ts:L41-L49), descritto in §3.8.

Il ponte è un array di due `FactoryProvider` (core-libs/core/src/features-config/feature-toggles/populate-feature-toggles-to-features-config.ts:L18-L33):

- `provideConfigFactory(() => ({ features: { ...inject(RootFeatureToggles) } }))` → un `ConfigChunk`;
- `provideDefaultConfigFactory(() => ({ features: { ...inject(DefaultFeatureToggles) } }))` → un `DefaultConfigChunk`.

`provideConfigFactory` e `provideDefaultConfigFactory` producono multi provider su `ConfigChunk` / `DefaultConfigChunk` (core-libs/core/src/config/config-providers.ts:L37-L48, core-libs/core/src/config/config-providers.ts:L69-L79). `RootConfig` e `DefaultConfig` sono il `deepMerge` dei rispettivi chunk e `Config = deepMerge({}, DefaultConfig, RootConfig)` (core-libs/core/src/config/config-tokens.ts:L10-L21, core-libs/core/src/config/config-tokens.ts:L23-L39). `FeaturesConfig` è `@Injectable({ providedIn: 'root', useExisting: Config })` (core-libs/core/src/features-config/config/features-config.ts:L27-L31) e il file aggiunge `FeaturesConfig` a `Config` con module augmentation (core-libs/core/src/features-config/config/features-config.ts:L120-L122): iniettare `FeaturesConfig` restituisce lo stesso oggetto `Config` globale.

Il ponte va **in un solo verso**: copia i toggle nella config, mai la config nei toggle. Chi scrive `provideConfig({ features: { x: true } })` lo vede con `FeatureConfigService.isEnabled('x')` e con `*cxFeature`, ma **non** con `inject(FeatureToggles).x` (vedi §9).

3.5 Lettura legacy: `FeatureConfigService` e `isFeatureEnabled`

`FeatureConfigService` è `providedIn: 'root'`, `@deprecated - Use FeatureToggles instead`, e riceve `FeaturesConfig`, cioè il `Config`, nel costruttore (core-libs/core/src/features-config/services/feature-config.service.ts:L15-L22). Ha due metodi (core-libs/core/src/features-config/services/feature-config.service.ts:L24-L33):

- `isLevel(version)` (deprecato, "features level is no longer used") → `isFeatureLevel(config, version)`;
- `isEnabled(feature)` → `isFeatureEnabled(config, feature)`.

`isFeatureEnabled` (core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L65):

1. se `config.features` è falsy, restituisce `false` (guardia `isFeatureConfig`: core-libs/core/src/features-config/utils/feature-config-utils.ts:L11-L13);
2. toglie l'eventuale `!` e legge `config.features[key]`;

3. se il valore è una **stringa**, lo tratta come livello di versione e delega a `isFeatureLevel`; altrimenti usa `valore ?? false`;
4. se l'espressione era negata, inverte il risultato.

La firma accetta anche il letterale `'disableConfigUpdates'` oltre a `FeatureToggleExpression` (core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L51): così `ConfigurationService.emitUnifiedConfig()` legge quella chiave per decidere se fare il deep-merge della config unificata (inclusi i chunk dei moduli lazy) dentro l'oggetto `Config` globale (core-libs/core/src/config/services/configuration.service.ts:L92-L106).

`isFeatureLevel` (core-libs/core/src/features-config/utils/feature-config-utils.ts:L36-L46) supporta la negazione `'!x.y'` e restituisce `false` se `features.level` manca. Il confronto vero è nel privato `isInLevel` (core-libs/core/src/features-config/utils/feature-config-utils.ts:L15-L34): `'*'` vale sempre; altrimenti scorre le parti numeriche della versione richiesta (le parti mancanti del livello valgono 0) e alla prima differenza restituisce `livello > versione`; se tutte le parti coincidono, `true`.

3.6 Lettura nei template: `*cxFeature`

`FeatureDirective` (selettore `[cxFeature]`) inietta `TemplateRef`, `ViewContainerRef` e il `FeatureConfigService` deprecato (core-libs/core/src/features-config/directives/feature.directive.ts:L11-L17). Il setter dell'input (core-libs/core/src/features-config/directives/feature.directive.ts:L19-L29):

- toggle abilitato e nessuna vista → `createEmbeddedView(templateRef)` e `hasView = true`;
- toggle disabilitato e vista presente → `viewContainer.clear()` e `hasView = false`.

La direttiva non si sottoscrive a nulla e rivaluta solo quando cambia il valore dell'input. Il pattern tipico è la coppia positivo/negato, vecchia UI contro nuova, come in `ProductImageZoomViewComponent` (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.html:L15-L15, feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.html:L35-L35). Allo SHA ci sono 165 occorrenze di `*cxFeature=` in 62 file `.html`, 67 delle quali negate; `cxFeatureLevel` non compare in nessun template, solo nella direttiva e nel mock core-libs/storefront/shared/test/mock-feature-level-directive.ts:L9-L9 (conteggi del dossier, F50).

`FeatureLevelDirective` (`[cxFeatureLevel]`) ha la stessa logica con `isLevel(level.toString())` ed è `@deprecated Unused directive` (core-libs/core/src/features-config/directives/feature-level.directive.ts:L10-L31). Entrambe le direttive non dichiarano `standalone`, che in Angular 21 vale `true` per default (`standalone: directiveDefinition.standalone ?? true`: angular:packages/core/src/render3/definition.ts:L637-L637; JSDoc dell'opzione: angular:packages/core/src/metadata/directives.ts:L335-L337). Per questo `FeaturesConfigModule` le mette in `imports` ed `exports` invece che in `declarations` (core-libs/core/src/features-config/features-config.module.ts:L23-L27).

3.7 Lettura in TypeScript: `inject(FeatureToggles)`

Il pattern raccomandato è un campo `private` (`private featureToggles = inject(FeatureToggles)`) letto come proprietà: `this.featureToggles.a11yKeyboardAccessibleZoom` (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.ts:L92-L92, feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.ts:L234-L234). Il dossier conta 126 occorrenze di `inject(FeatureToggles)` in 117 file non-spec e 180 accessi `featureToggles.<chiave>` in 99 file. Fuori da `features-config`, a runtime `FeatureConfigService` è iniettato solo da `CxApiService`, come campo `public` opzionale (core-libs/

storefront/cms-structure/page/component/services/cx-api.service.ts:L46-L46). Le altre occorrenze sono dati delle migrazioni schematics, regole ESLint e commenti.

I toggle si leggono anche **dentro le factory di config di default**. `defaultAuthConfigFactory` destruttura `authorizationCodeFlowByDefault`, `asyncAuthConfigInitializer` e `oauthCallbackPage` da `inject(FeatureToggles)` (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L76-L83) ed è registrata con `provideDefaultConfigFactory(defaultAuthConfigFactory)` in `UserAuthModule` (core-libs/core/src/auth/user-auth/user-auth.module.ts:L106-L106). Qui si usa `FeatureToggles` e non `FeatureConfigService` per evitare un ciclo di DI: `FeatureConfigService` dipende da `Config`, che a sua volta dipende da tutti i `DefaultConfigChunk`, compresa questa factory. La regola ESLint lo dice esplicitamente: `inject(FeatureToggles)` "is preferred over `inject(FeatureConfigService)` inside `provideDefaultConfigFactory()` to avoid circular DI" (tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L23-L26). Il flusso auth completo è in 22 — Autenticazione.

3.8 Stili per feature: `useFeatureStyles` + `forFeature` + `FeatureStylesService`

Un toggle può attivare regole CSS senza toccare il DOM del componente. I pezzi sono tre.

SCSS. `_core.scss` importa prima `features` e poi `versioning` (core-libs/styles/scss/_core.scss:L1-L2). `_features.scss` definisce il prefisso `$CSS_FEATURE_FLAG_PREFIX: 'cxFeat_'` e la funzione `getFeatureCssClass` (core-libs/styles/scss/_features.scss:L3-L14). Il mixin `forFeature($feature, $classTarget: 'ancestorSelector')`:

- valida i parametri con `@error`: `$feature` deve essere una stringa, `$classTarget` deve valere `'ancestorSelector'` o `'currentSelector'` (core-libs/styles/scss/_features.scss:L42-L49);
- con `ancestorSelector` emette `@at-root .cxFeat_<f> & { ... }`, cioè la classe sta su un antenato; con `currentSelector` emette `@at-root &.cxFeat_<f> { ... }`, cioè la classe sta sullo stesso elemento (core-libs/styles/scss/_features.scss:L51-L61). Il secondo caso va usato "ONLY in one edge case", per gli stili dell'elemento root (core-libs/styles/scss/_features.scss:L38-L40). L'unico uso di produzione è in `root.scss`, dentro `:root`, per `a11yPreventWindowsHighContrastOverride` (core-libs/styles/scss/root.scss:L36-L44).

TS, lato componente. `useFeatureStyles(feature)` chiama `assertInInjectionContext`, inietta `FeatureStylesService` e `DestroyRef`, registra l'uso e lo cancella in `destroyRef.onDestroy` (core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40). La JSDoc impone di chiamarla nel costruttore di **ogni** componente il cui SCSS usa `forFeature`, altrimenti gli stili "MIGHT NOT be activated EVEN if the flag is enabled" (core-libs/core/src/features-config/utils/use-feature-styles.ts:L11-L31). Fuori da un injection context, `assertInInjectionContext` lancia `RuntimeError` `MISSING_INJECTION_CONTEXT` (angular:packages/core/src/di/contextual.ts:L77-L88).

TS, lato servizio. `FeatureStylesService` (core-libs/core/src/features-config/services/feature-styles.service.ts:L17-L26):

1. legge `FeatureToggles` (non `FeaturesConfig`) e crea un `Renderer2` con `createRenderer(null, null)` (core-libs/core/src/features-config/services/feature-styles.service.ts:L27-L41);
2. tiene `usagesCounter: Map<string, number>`: quando un contatore passa a >0 aggiunge la classe, quando torna a 0 la toglie (core-libs/core/src/features-config/services/feature-styles.service.ts:L50-L56);
3. `registerUsage` aggiunge la classe solo se il flag è abilitato e il contatore era 0, poi incrementa il contatore **anche se il flag è spento** (core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L95);
4. `unregisterUsage` in dev mode avvisa se il contatore è già 0, toglie la classe se il flag è abilitato e il contatore vale 1, poi decrementa con `Math.max(..., 0)` (core-libs/core/src/features-config/services/

feature-styles.service.ts:L103-L115);

5. addClass/removeClass non fanno nulla se rootElement non è ancora impostato o se getCssClass restituisce undefined (core-libs/core/src/features-config/services/feature-styles.service.ts:L130-L150). getCssClass restituisce cxFeat_<feature> solo se il flag è truthy (core-libs/core/src/features-config/services/feature-styles.service.ts:L158-L162), e isEnabled è !!this.featureToggles[feature] (core-libs/core/src/features-config/services/feature-styles.service.ts:L120-L122).

Bootstrap. Il servizio non conosce l'elemento root finché non riceve init. Angular chiama gli APP_BOOTSTRAP_LISTENER in _loadComponent, **dopo** il primo tick() del componente root, leggendoli dall'injector in modo lazy "to prevent DI cycles" (angular:packages/core/src/application/application_ref.ts:L775-L794; token: angular:packages/core/src/application/application_ref.ts:L64-L66). Il listener di forRoot() passa il ComponentRef a init, che salva rootComponent.location.nativeElement e aggiunge le classi per tutti i contatori >0 già registrati (core-libs/core/src/features-config/services/feature-styles.service.ts:L65-L77). Senza questo recupero si perderebbero le registrazioni fatte nei costruttori dei componenti creati prima dell'invocazione dei listener (il commento del sorgente cita il costruttore del root component), che arrivano prima dell'init. DA VERIFICARE: in quale fase esatta (creazione del root component o primo tick()) vengono costruiti i componenti figli come StorefrontComponent non è stato tracciato nel renderer di Angular; per il meccanismo conta solo che avvenga prima dei listener.

StorefrontComponent registra così sei stili globali nel costruttore (core-libs/storefront/layout/main/storefront.component.ts:L118-L123). Per uno di questi la classe sull'elemento root dell'app non basta: le regole stanno su :root, cioè <html>. Per questo ngOnInit aggiunge a mano cxFeat_a11yPreventWindowsHighContrastOverride a document.documentElement ("TODO: Required to use feature flag with root styles": core-libs/storefront/layout/main/storefront.component.ts:L141-L147).

Il dossier conta 56 chiamate a useFeatureStyles(in 44 file TS non-spec (esclusa features-config) e 70 @include forFeature(in 40 file .scss di produzione (78 in 41 contando lo spec core-libs/styles/test/features.spec.scss) (F73).

3.9 Il vecchio approccio SCSS: _versioning.scss

_versioning.scss apre con "🚨 CAUTION": forVersion($level) è deprecato e sostituito da forFeature($featureName) (core-libs/styles/scss/_versioning.scss:L1-L2). Il file contiene:

- le versioni interne $_fullVersion: (major: 221121, minor: 22, patch: 0) e $_majorVersion: 2211 (core-libs/styles/scss/_versioning.scss:L29-L37);
- le variabili di opt-in del cliente, $styleVersion: $_majorVersion !default e $useLatestStyles: false !default (core-libs/styles/scss/_versioning.scss:L41-L46);
- il mixin forVersion($from, $to: 0), @deprecated since 2211.20, che emette il contenuto solo se $from <= $_fullVersion e la versione è stabile e valida, oppure è richiesta come breaking change (core-libs/styles/scss/_versioning.scss:L63-L77);
- useBreakingChanges(), vera se $useLatestStyles è attivo o $styleVersion > $_majorVersion (core-libs/styles/scss/_versioning.scss:L228-L231).

forVersion non ha usi di produzione: compare solo nel commento d'esempio (core-libs/styles/scss/_versioning.scss:L18-L18) e nella utility di test (core-libs/styles/test/versioning-test-util.scss:L6-L6). La storefrontapp imposta comunque $useLatestStyles: true; (projects/storefrontapp/src/styles-config.scss:L2-L2).

3.10 Chi fornisce i toggle nell'applicazione

- **App demo**: `provideFeatureTogglesFactory` restituisce un oggetto tipato `Required<FeatureToggles>`, quindi aggiungere un toggle alla libreria obbliga ad aggiornare anche l'app (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L297-L301). La regola `no-storefrontapp-false-feature-toggles` impone che tutti i valori siano `true`; un `false` è ammesso solo con un eslint-disable che citi un ticket CX-SPA (tools/eslint-rules/rules/no-storefrontapp-false-feature-toggles.ts:L33-L45, projects/storefrontapp/eslint.config.mjs:L26-L28).
- **App nuova (`ng add`)**: lo schematic genera `provideFeatureToggles({...})` con **tutte** le chiavi di `defaultFeatureToggles` a `true` (core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L77-L93). Per non dipendere a runtime da `@spartacus/core`, lo schematic legge una copia del file TS del core (core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L95-L113). La copia è fatta in pre-build dallo script `copy-feature-toggles` (core-libs/schematics/package.json:L16-L16) con un semplice `fs.copyFileSync` verso `./src/feature-toggles.copied-from-core-lib.ts` (core-libs/schematics/copy-feature-toggles.js:L5-L10); il path di lettura è la costante `FEATURE_TOGGLES_COPIED_FROM_CORE_LIB_PATH` (core-libs/schematics/src/shared/constants.ts:L1176-L1177). Gli schematics sono trattati in 43 — Schematics e migrazioni.
- **Test con `TestConfigModule`**: fornisce di proposito toggle ad alta priorità con `provideFeatureTogglesFactory(() => inject(TEST_CONFIG)?.features)`, disattivando la regola di lint per quella riga (core-libs/core/src/config/test-config.module.ts:L94-L98).

3.11 Strumenti di test

`provideMockFeatureToggles(initial)` crea un `MockFeatureTogglesController` e lo lega **come stesso oggetto** a `MockFeatureTogglesController`, `FeatureToggles`, `RootFeatureToggles`, `DefaultFeatureToggles` e `FeatureTogglesChunk` (multi), e lo inserisce anche in un `ConfigChunk` (`features: controller`) tramite `provideConfigFactory` (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L150-L204). `Config.features`, però, nasce dal deep-merge dei chunk e non condivide il riferimento: per questo il mock sostituisce anche `FeatureConfigService` (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L121-L125). Il controller è un oggetto che fa sia da toggle sia da API di controllo: `set` assegna una chiave, `reset` cancella tutto tranne `set`/`reset` e poi applica `Object.assign` (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L95-L119). Viene creato da capo a ogni setup di TestBed, quindi le mutazioni non passano da un test all'altro (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L154-L159).

Il mock sostituisce anche `FeatureConfigService` con un `MockFeatureConfigService` che legge il controller (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L206-L216). Senza questa sostituzione `*cxFeature` leggerebbe lo snapshot `Config.features`, già deep-merged al bootstrap, e non vedrebbe le chiamate a `set(...)`. `MockFeatureConfigService` riusa `isFeatureEnabled`/`isFeatureLevel` passando un finto `Config { features: this.toggles }` (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L132-L148).

Il testing **non** è un entry point pubblicato: le librerie lo raggiungono tramite alias nei tsconfig o nella config vitest, per esempio `@spartacus/core/testing/feature-toggles` in storefront (core-libs/storefront/tsconfig.spec.json:L17-L19) e `@spartacus/core/testing/mock-feature-toggles` in checkout (feature-libs/checkout/vitest.config.ts:L26-L26). `provideMockFeatureToggles` compare in 87 file `.spec.ts` (F95). Dettagli in 44 — Test tooling.

3.12 Deprecazioni: cosa dice il codice

- **Conteggi.** `@deprecated` nei `.ts` non-spec delle librerie, esclusi i path con `schematics`, `testing` e `mock`: 165 occorrenze in 124 file, uguale al totale di A10 (`docs/spartacus-book/appendici/A10-deprecated.md:L11-L30`). La distribuzione è molto sbilanciata: `@spartacus/epd-visualization` ne ha 91, `@spartacus/core` 14. Delle 14 di core, 10 stanno nella sola cartella `features-config` (F109).
- **Formati.** Non sono uniformi: con versione (`@deprecated since 2211.22 - should not be a public API`: `core-libs/core/src/config/config-initializer/config-initializer.module.ts:L33-L33`; `@deprecated since 2211.32. Use initOnce() ...`: `core-libs/core/src/site-context/services/site-context-routes-handler.ts:L51-L51`), con alternativa e senza versione (`@deprecated - Use FeatureToggles instead`), oppure in un commento `//` con una scadenza ("will be removed in 12 months": `core-libs/core/src/cms/facade/page-meta.service.ts:L39-L39`).
- **Da token a toggle.** `USE_AUTHORIZATION_CODE_FLOW_BY_DEFAULT` è "@deprecated since 221121.1 Please use the `authorizationCodeFlowByDefault` feature toggle instead" (`core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L11-L19`). Il provider legacy `provideAuthorizationCodeFlowByDefault(enable)` chiama `inject(FeatureToggles)` nel proprio corpo e usa `authorizationCodeFlowByDefault ?? enable` (`core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L37-L46`).
- **Fine vita dei toggle.** Il codice da rimuovere è segnato con TODO "Remove feature flag next major release.", di solito con un ticket CXSPA: 23 occorrenze in 14 file ts/html/scss non-spec (F111), per esempio `core-libs/storefront/cms-components/product/product-intro/product-intro.component.html:L6-L6`. La "lifespan" di ogni toggle è "defined in the official docs" (`core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L16-L19`). DA VERIFICARE: nel repo non c'è una policy di rimozione codificata oltre a questi TODO; la documentazione ufficiale non è stata verificata.
- **Lint come politica.** `feature-toggles-must-be-private` richiede che `FeatureToggles` sia iniettato `private`: la rimozione di un toggle, "the normal end-of-life of a toggle", non deve diventare un breaking change TypeScript per chi lo referenzia da una sottoclasse (`tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L11-L26`). Questa regola e `feature-config-service-must-be-private` sono `error` su `**/*.ts` (`eslint.config.mjs:L255-L256`). Nei `**/*.module.ts` delle librerie, `use-provide-default-feature-toggles(-factory)` sono `error` (`eslint.config.mjs:L237-L238`, messaggio: `tools/eslint-rules/rules/use-provide-default-feature-toggles.ts:L34-L35`); la storefrontapp le spegne (`projects/storefrontapp/eslint.config.mjs:L14-L15`).

---

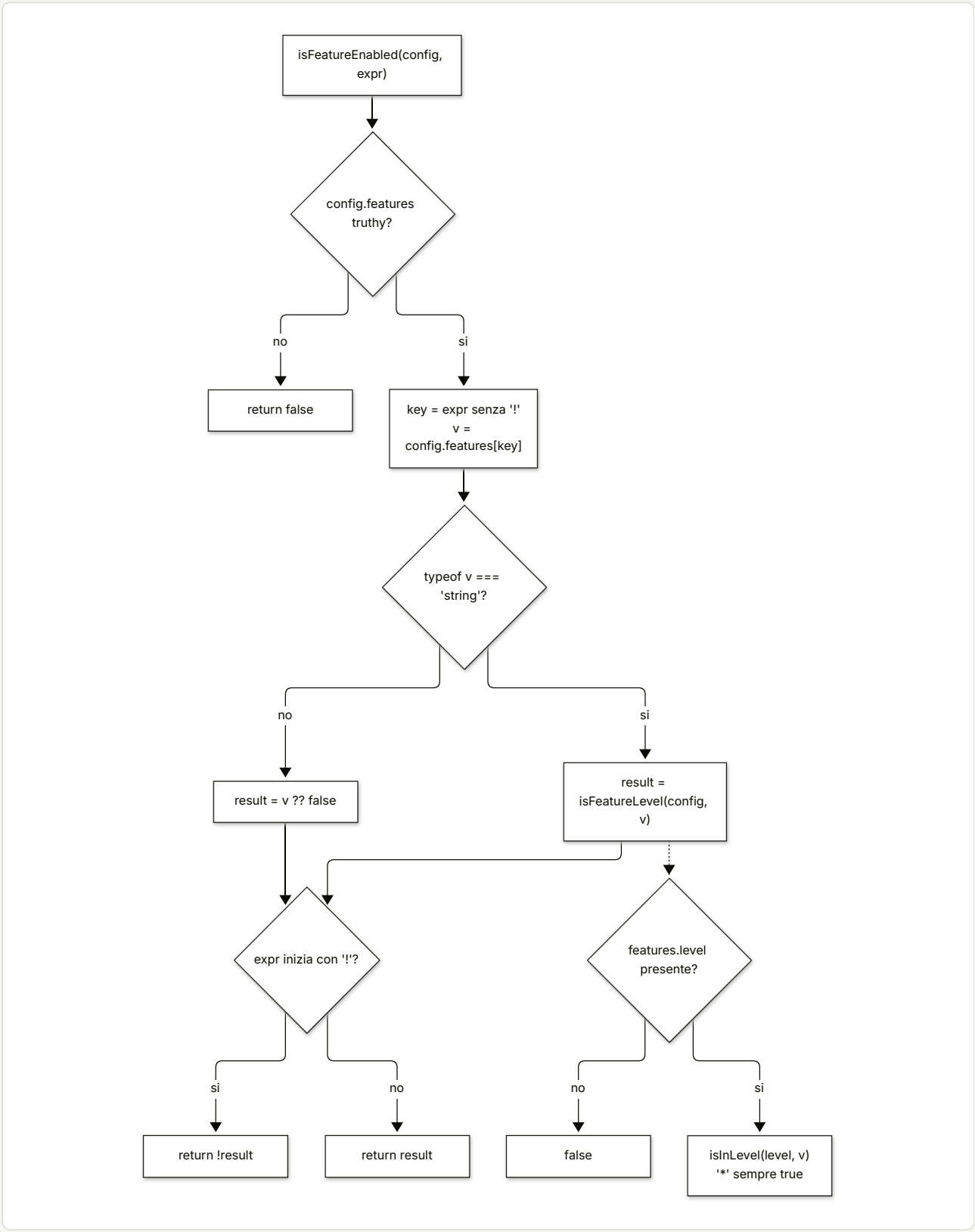
4. Diagrammi

4.1 Costruzione di `FeatureToggles` e ponte verso `Config.features`

■ **TAVOLA 42.1** diagramma a tutta pagina, in fondo al capitolo

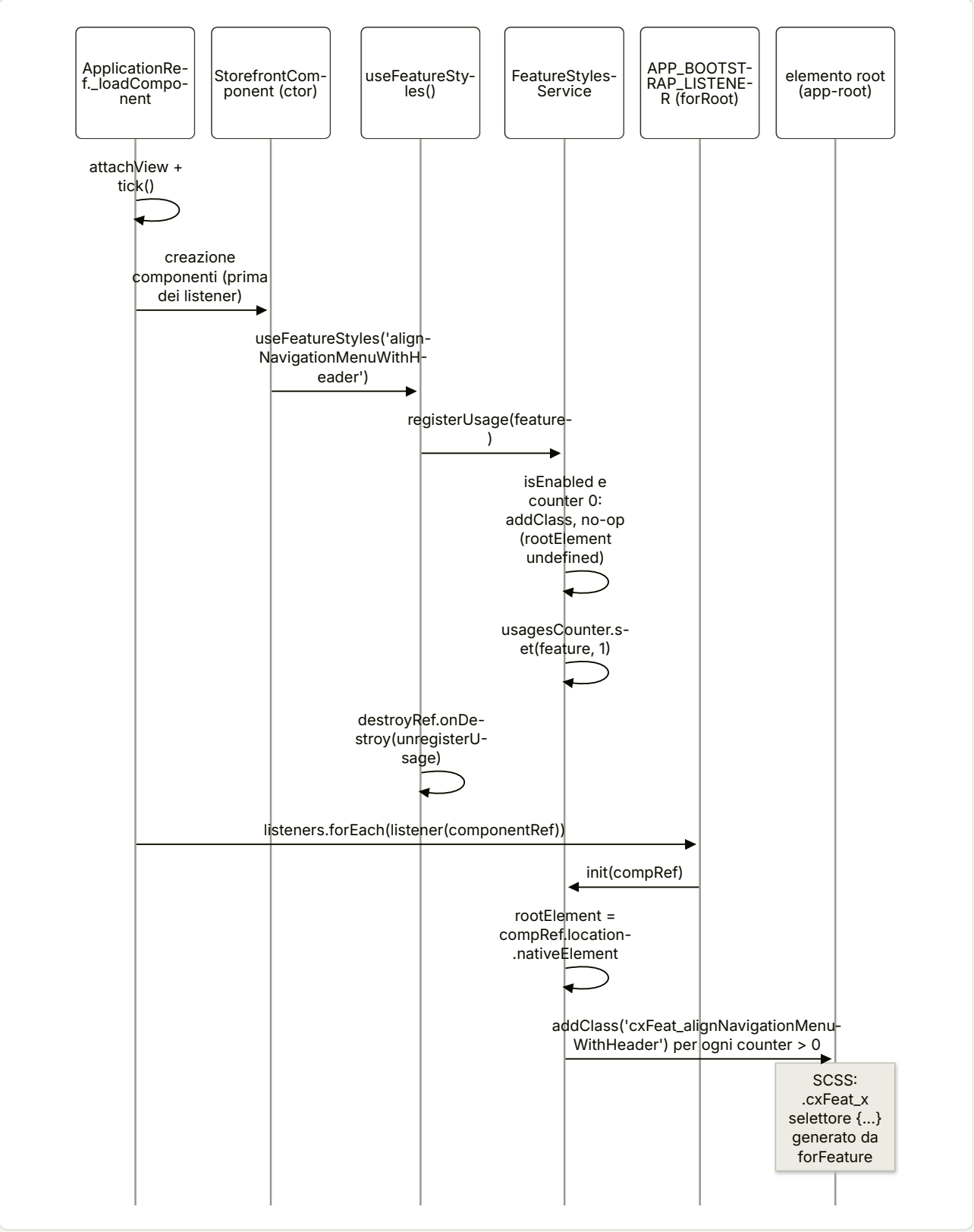
Riferimenti: chunk e token `core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L15-L78`; ponte `core-libs/core/src/features-config/feature-toggles/populate-feature-toggles-to-features-config.ts:L18-L33`; `Config` `core-libs/core/src/config/config-tokens.ts:L10-L39`; `useExisting` `core-libs/core/src/features-config/config/features-config.ts:L27-L31`. Non c'è nessuna freccia da `Config` verso `FeatureToggles`: il ponte è a senso unico.

4.2 Valutazione di un'espressione legacy (`isFeatureEnabled`)



Riferimenti: `core-libs/core/src/features-config/utils/feature-config-utils.ts:L11-L65`.

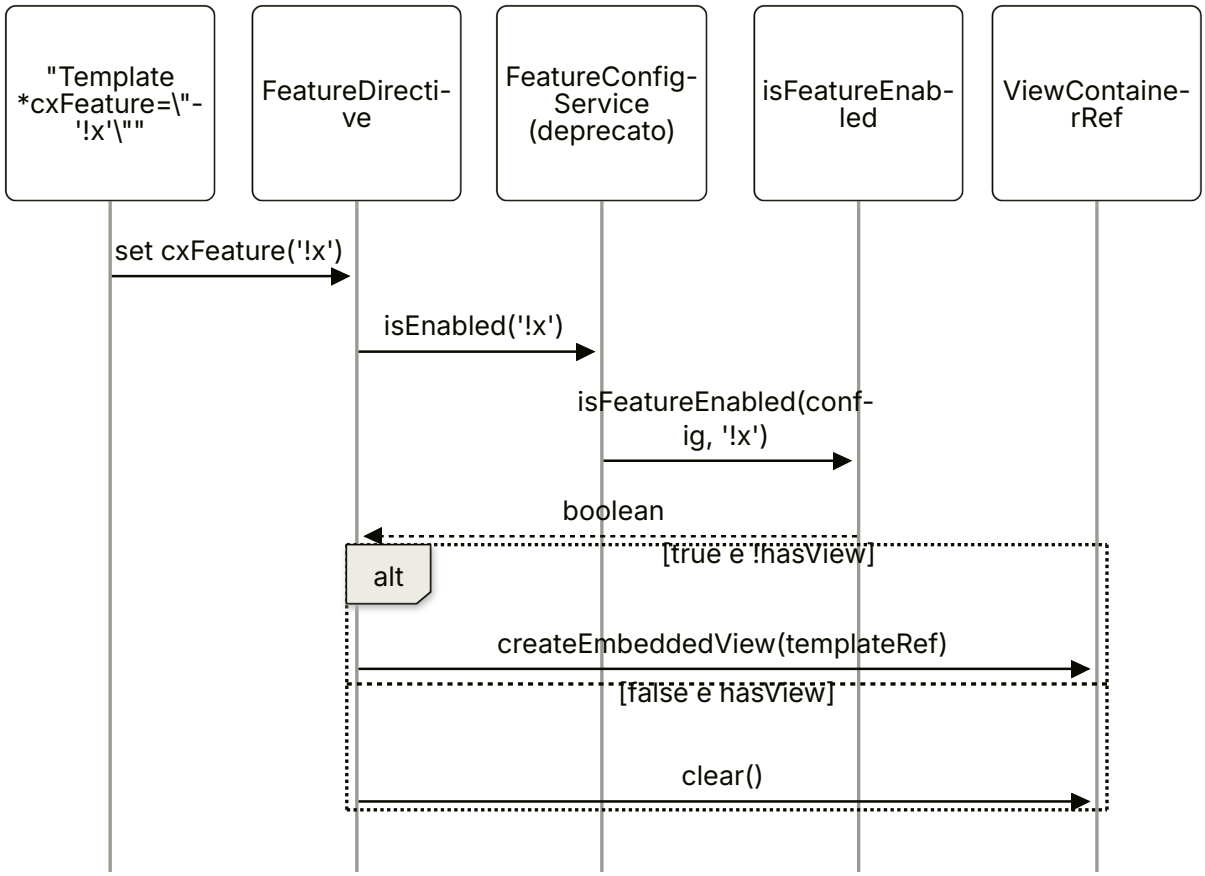
4.3 Stili per feature durante il bootstrap



Riferimenti: angular:packages/core/src/application/application_ref.ts:L775-L794; core-libs/storefront/layout/main/storefront.component.ts:L118-L123; core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40; core-libs/core/src/features-config/services/feature-styles.service.ts:L65-L95;

core-libs/core/src/features-config/services/feature-styles.service.ts:L130-L136; core-libs/core/src/features-config/features-config.module.ts:L41-L49; core-libs/styles/scss/_features.scss:L51-L56.

4.4 `*cxFeature` nel template



Riferimenti: core-libs/core/src/features-config/directives/feature.directive.ts:L19-L29; core-libs/core/src/features-config/services/feature-config.service.ts:L31-L33.

---

5. Estratti di codice reale

Il token principale e il tipo dell'espressione — core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L15-L28

```ts
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    Object.assign(
      {},
      inject(DefaultFeatureToggles),
      inject(RootFeatureToggles)
    ),
```

```
})
export abstract class FeatureToggles {}
export interface FeatureToggles extends FeatureTogglesInterface {}

export type FeatureToggleKey = keyof FeatureToggles;
export type FeatureToggleExpression = FeatureToggleKey | `!${FeatureToggleKey}`;
```

Interfaccia strict per scelta — core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11

```ts
// Let's NOT add here any wildcard property like
//  `[key: string]: boolean | undefined;`
// We want this interface to be STRICT and cause a compilation error when a removed property is used.
// Thanks to that, customers using a property that was recently removed, will know they have to adapt
their code.
export interface FeatureTogglesInterface {
```

Provider per l'applicazione — core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L25-L33

```ts
export function provideFeatureToggles(
  toggles: FeatureToggles = {}
): ValueProvider {
  return {
    provide: FeatureTogglesChunk,
    useValue: toggles,
    multi: true,
  };
}
```

Il ponte verso la config legacy — core-libs/core/src/features-config/feature-toggles/populate-feature-toggles-to-features-config.ts:L18-L33

```ts
export const populateFeatureTogglesToFeaturesConfig: FactoryProvider[] = [
  // Copies RootFeatureToggles to RootConfig
  provideConfigFactory(() => {
    const flags = inject(RootFeatureToggles);
    return { features: { ...flags } };
  }),

  // Copies DefaultFeatureToggles to DefaultFeaturesConfig
  provideDefaultConfigFactory(() => {
    const flags = inject(DefaultFeatureToggles);
    return { features: { ...flags } };
  }),
];
```

forRoot() — core-libs/core/src/features-config/features-config.module.ts:L28-L52

```ts
  static forRoot(
    defaultLevel = '3.0'
  ): ModuleWithProviders<FeaturesConfigModule> {
    return {
      ngModule: FeaturesConfigModule,
      providers: [
        provideDefaultFeatureToggles(defaultFeatureToggles),
        ...populateFeatureTogglesToFeaturesConfig, // for backward compatibility with the deprecated
FeaturesConfig
        provideDefaultConfig(<FeaturesConfig>{
          features: {
            level: defaultLevel || '*',
          },
        }),
        {
          provide: APP_BOOTSTRAP_LISTENER,
          multi: true,
          useFactory: (): ((compRef: ComponentRef<any>) => void) => {
            const featureStylesService = inject(FeatureStylesService);
            return (compRef: ComponentRef<any>) =>
              featureStylesService.init(compRef);
          },
        },
      ],
    };
  }
```

Valutazione legacy — core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L65

```ts
export function isFeatureEnabled(
  config: Config,
  feature: FeatureToggleExpression | 'disableConfigUpdates'
): boolean {
  if (isFeatureConfig(config)) {
    const featureConfig = feature.startsWith('!')
      ? config.features[feature.substring(1)]
      : config.features[feature];

    const result =
      typeof featureConfig === 'string'
        ? isFeatureLevel(config, featureConfig)
        : (featureConfig ?? false);

    return feature.startsWith('!') ? !result : result;
  }
  return false;
}
```

La direttiva strutturale — core-libs/core/src/features-config/directives/feature.directive.ts:L19-L29

```ts
  private hasView = false;

  @Input() set cxFeature(feature: FeatureToggleExpression) {
    if (this.featureConfig.isEnabled(feature) && !this.hasView) {
      this.viewContainer.createEmbeddedView(this.templateRef);
      this.hasView = true;
    } else if (!this.featureConfig.isEnabled(feature) && this.hasView) {
      this.viewContainer.clear();
      this.hasView = false;
    }
  }
```

Registrazione d'uso degli stili — core-libs/core/src/features-config/services/feature-styles.service.ts:L65-L95

```ts
  init(rootComponent: ComponentRef<any>): void {
    this.rootElement = rootComponent.location.nativeElement;

    // Handle edge-case:
    // Add classes for all features that were used (called `registerUsage()`) before `init()` was
invoked.
    // (e.g. it's the case when `registerUsage` was called in the constructor of the root component,
    //       which happens before the hook APP_BOOTSTRAP_LISTENER is invoked by Angular, therefore
before `init` is invoked.)
    for (const [feature, count] of this.usagesCounter.entries()) {
      if (count > 0) {
        this.addClass(feature as FeatureToggleKey);
      }
    }
  }
  // ...
  registerUsage(feature: FeatureToggleKey): void {
    const currentCounter = this.usagesCounter.get(feature) ?? 0;
    if (this.isEnabled(feature) && currentCounter === 0) {
      this.addClass(feature);
    }
    this.usagesCounter.set(feature, currentCounter + 1);
  }
```

useFeatureStyles — core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40

```ts
export function useFeatureStyles(feature: FeatureToggleKey): void {
  assertInInjectionContext(useFeatureStyles);

  const featureStylesService = inject(FeatureStylesService);
  const destroyRef = inject(DestroyRef);
```

```
  featureStylesService.registerUsage(feature);
  destroyRef.onDestroy(() => featureStylesService.unregisterUsage(feature));
}
```

Il mixin SCSS — core-libs/styles/scss/_features.scss:L41-L62

```scss
@mixin forFeature($feature, $classTarget: 'ancestorSelector') {
  // Validate params
  @if (type-of($feature) != 'string') {
    @error "forFeature: $feature param must be a string, but #{type-of($feature)} was provided.";
  }
  @if (not index(('ancestorSelector', 'currentSelector'), $classTarget)) {
    @error "forFeature: $classTarget must be either 'ancestorSelector' or 'currentSelector', but #
{$classTarget} was provided.";
  }
  // Validate params end

  $featureClass: getFeatureCssClass($feature);
  @if ($classTarget == 'ancestorSelector') {
    @at-root .#{$featureClass} & {
      @content;
    }
  }
  @if ($classTarget == 'currentSelector') {
    @at-root &.#{$featureClass} {
      @content;
    }
  }
}
```

Coppia TS + SCSS in un componente reale — feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.ts:L181-L188 e feature-libs/product/image-zoom/styles/_product-image-zoom-view.scss:L14-L19

```ts
  constructor(
    protected currentProductService: CurrentProductService,
    protected renderer: Renderer2,
    protected cdRef: ChangeDetectorRef,
    protected breakpointService: BreakpointService
  ) {
    useFeatureStyles('a11yKeyboardAccessibleZoom');
  }
```

```scss
      // TODO: (CXSPA-7492) - Remove feature flag next major release.
      @include forFeature('a11yKeyboardAccessibleZoom') {
        cx-media {
```

```
          display: unset;
        }
      }
```

Workaround per gli stili su `:root` — core-libs/storefront/layout/main/storefront.component.ts:L141-L147

```ts
    // TODO: Required to use feature flag with root styles.
    //       Remove this entire block once the a11yPreventWindowsHighContrastOverride feature flag is removed
    if (this.featureToggles.a11yPreventWindowsHighContrastOverride) {
      this.document?.documentElement.classList.add(
        'cxFeat_a11yPreventWindowsHighContrastOverride'
      );
    }
```

Lo schematic accende tutto — core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L80-L93

```ts
function createFeatureTogglesProvider(): string {
  const defaultFeatureToggles = getDefaultFeatureToggles();

  //for each key in `defaultFeatureToggles` map it to a key with value true
  const featureTogglesAllEnabled: Record<string, boolean> = Object.keys(
    defaultFeatureToggles
  ).reduce((acc, key) => ({ ...acc, [key]: true }), {});
  const featureTogglesString = JSON.stringify(
    featureTogglesAllEnabled,
    null,
    2
  );
  return `provideFeatureToggles(${featureTogglesString})`;
}
```

Il mock dei test sostituisce anche il servizio legacy — core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L206-L216

```ts
    // Override `FeatureConfigService` (used by the `*cxFeature` directive
    // and any other `isEnabled()` consumer) so it queries our mutable
    // controller directly. Otherwise the directive would read from the
    // immutable, deep-merged `Config.features` snapshot built at bootstrap
    // and would NOT see runtime mutations done via `controller.set(...)`.
    {
      provide: FeatureConfigService,
      useFactory: (c: MockFeatureTogglesController) =>
        new MockFeatureConfigService(c as unknown as Record<string, unknown>),
      deps: [MockFeatureTogglesController],
    },
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`FeatureToggles.<chiave>` (107 chiavi, catalogo in A09)	`boolean`	da `defaultFeatureToggles`: 32 `true` / 75 `false`	interfaccia `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L11-L1011`; default `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1013-L1121`	`inject(FeatureToggles)`; `FeatureConfigService.isEnabled` / `*cxFeature` tramite il ponte; `FeatureStylesService` (`core-libs/core/src/features-config/services/feature-styles.service.ts:L120-L122`)
`features.<chiave>` (legacy)	`string | boolean | undefined`	copia dei toggle tramite il ponte	`core-libs/core/src/features-config/config/features-config.ts:L72-L88`; ponte `core-libs/core/src/features-config/feature-toggles/populate-feature-toggles-to-features-config.ts:L18-L33`	`isFeatureEnabled` (`core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L65`)
`features.level` (deprecata)	`string`	`'3.0'` (default di `forRoot`), `'*'` se l'argomento è falsy	`core-libs/core/src/features-config/config/features-config.ts:L49-L70`; `core-libs/core/src/features-config/features-config.module.ts:L28-L40`	`isFeatureLevel` (`core-libs/core/src/features-config/utils/feature-config-utils.ts:L36-L46`), anche quando un toggle legacy ha valore stringa (`core-libs/core/src/features-config/utils/feature-config-utils.ts:L57-L60`)
`features.consignmentTracking`	`boolean`	`'1.2'` (stringa, fornita da `OrderDetailsModule` con `provideDefaultConfig`, interpretata come livello: `true` con `level '3.0'`; `feature-libs/order/components/order-details/order-details.module.ts:L173-L175`). La chiave non è in `FeatureTogglesInterface`: arriva solo dalla config legacy	`core-libs/core/src/features-config/config/features-config.ts:L90-L99`	`*cxFeature` in `OrderConsignedEntriesComponent` (`feature-libs/order/components/order-details/order-detail-items/order-consigned-entries/order-consigned-entries.component.html:L19-L19`)
`features.disableConfigUpdates` (deprecata)	`boolean`	non definita → `false` (`?? false`)	`core-libs/core/src/features-config/config/features-config.ts:L101-L116`	`core-libs/core/src/config/services/configuration.service.ts:L92-L106`
`$styleVersion` (SCSS, deprecata)	numero di versione	`$_majorVersion` (2211) `!default`	`core-libs/styles/scss/_versioning.scss:L41-L41`	`useBreakingChanges()` (`core-libs/styles/scss/_versioning.scss:L228-L231`)

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`$useLatestStyles` (SCSS, deprecata)	bool	`false !default;` storefrontapp `true`	core-libs/styles/scss/_versioning.scss:L46-L46; projects/storefrontapp/src/styles-config.scss:L2-L2	core-libs/styles/scss/_versioning.scss:L228-L231
`$CSS_FEATURE_FLAG_PREFIX` (SCSS) / `CSS_FEATURE_FLAG_PREFIX` (TS)	string	`'cxFeat_'`	core-libs/styles/scss/_features.scss:L6-L6; core-libs/core/src/features-config/services/feature-styles.service.ts:L31-L36	`getFeatureCssClass` (core-libs/styles/scss/_features.scss:L12-L14); `getCssClass` (core-libs/core/src/features-config/services/feature-styles.service.ts:L158-L162)
`forFeature(..., $classTarget)`	`'ancestorSelector' | 'currentSelector'`	`'ancestorSelector'`	core-libs/styles/scss/_features.scss:L41-L41	core-libs/styles/scss/_features.scss:L51-L61

7. Commenti e note notevoli dal sorgente

- "Let's NOT add here any wildcard property ... We want this interface to be STRICT" (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L10): la rottura in compilazione quando un toggle sparisce è una **scelta**, non un effetto collaterale.
- "don't export from './config/feature-toggles' as it contains private API" (core-libs/core/src/features-config/feature-toggles/index.ts:L7-L7).
- "for backward compatibility with the deprecated FeaturesConfig" (core-libs/core/src/features-config/features-config.module.ts:L35-L35).
- JSDoc di `provideFeatureToggles`: "For using in a library, call `provideDefaultFeatureToggles() instead."` (nel sorgente il backtick non è chiuso: core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L21-L21`).
- I `@deprecated` di `FeaturesConfig` coprono la classe, `features`, `level`, la index signature e `disableConfigUpdates` (core-libs/core/src/features-config/config/features-config.ts:L11-L11, core-libs/core/src/features-config/config/features-config.ts:L33-L33, core-libs/core/src/features-config/config/features-config.ts:L54-L54, core-libs/core/src/features-config/config/features-config.ts:L73-L73, core-libs/core/src/features-config/config/features-config.ts:L102-L102). L'ultimo rimanda a CXSPA-13592 per lo spostamento in un injection token.
- "The properties below should be moved in the future to the global config, because they are not temporary feature toggles but rather permanent config options" (core-libs/core/src/features-config/config/features-config.ts:L90-L99): una distinzione esplicita tra *toggle temporaneo* e *opzione permanente*.
- `@deprecated Unused directive. Please use named feature flags instead of feature level.` (core-libs/core/src/features-config/directives/feature-level.directive.ts:L11-L11).
- "🚨 CAUTION. The approach of using `forVersion($level)` in this file is deprecated" (core-libs/styles/scss/_versioning.scss:L1-L2) e `// @deprecated since 2211.20` (core-libs/styles/scss/_versioning.

scss:L63-L63): una deprecazione dentro SCSS, dove non c'è nessun compilatore che la segnali.

- "IMPORTANT: Please mind to call `useFeatureStyles(feature)` ... Otherwise the feature-flagged styles MIGHT NOT be activated" (core-libs/styles/scss/_features.scss:L18-L21).
- `$featureMessagingItems: true;` (core-libs/styles/scss/_features.scss:L1-L1): variabile che non è referenziata da nessun'altra parte del repo (F80).
- Lo spec avverte che la precedenza tra `FeaturesConfig` e `FeatureToggles` "might not be intuitive" (core-libs/core/src/features-config/features-config.module.spec.ts:L84-L93).
- Il commento della config vitest di checkout spiega perché serve l'alias di `@spartacus/core` verso il sorgente: altrimenti "`*cxFeature` never sees the mocked toggles" (feature-libs/checkout/vitest.config.ts:L35-L42).
- JSDoc incoerente: `a11yCardNotificationMessage` descrive un fix sulla tabella organization, non solo sul messaggio della card (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L237-L242).

8. Extension point

PUNTO	TIPO	DEFINIZIONE	NOTE
`provideFeatureToggles`, `provideFeatureTogglesFactory`	funzioni provider (app)	core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L25-L52	da usare nel root injector
`provideDefaultFeatureToggles`, `provideDefaultFeatureTogglesFactory`	funzioni provider (librerie)	core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L54-L86	regola ESLint nelle librerie Spartacus
`FeatureToggles`	classe astratta DI + interfaccia	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L15-L25	iniettare `private`
`FeatureTogglesChunk`, `DefaultFeatureTogglesChunk`, `RootFeatureToggles`, `DefaultFeatureToggles`	`InjectionToken`	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L33-L78	"Should not be used directly" per i chunk
`FeatureToggleKey`, `FeatureToggleExpression`, `FeatureTogglesFactory`	tipi	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L27-L28, core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L14-L14	—
`FeatureDirective` (`*cxFeature`)	direttiva standalone	core-libs/core/src/features-config/directives/feature.directive.ts:L11-L29	importabile anche via `FeaturesConfigModule`

PUNTO	TIPO	DEFINIZIONE	NOTE
`FeatureStylesService`	servizio con metodi `protected` sovrascrivibili (`isEnabled`, `addClass`, `removeClass`, `getCssClass`)	`core-libs/core/src/features-config/services/feature-styles.service.ts:L25-L163`	sostituibile via DI
`useFeatureStyles`	funzione da injection context	`core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40`	nel costruttore
`forFeature($feature, $classTarget)`	mixin SCSS	`core-libs/styles/scss/_features.scss:L41-L62`	da accoppiare a `useFeatureStyles`
`isFeatureEnabled`	funzione pura	`core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L65`	legge un `Config`
`FeaturesConfig`, `FeatureConfigService`, `FeatureLevelDirective`, `isFeatureLevel`, `forVersion`	legacy	vedi §3.4–3.9	`@deprecated`: non costruirci sopra codice nuovo
`provideMockFeatureToggles`, `MockFeatureTogglesController`	test util	`core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L48-L218`	**non** nel public API; solo con alias
`FeatureTogglesInterface`, `defaultFeatureToggles`	—	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L11-L11`, `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1013-L1013`	**non** esportati (`core-libs/core/src/features-config/feature-toggles/index.ts:L7-L9`)

Tutto ciò che è pubblico arriva da `core-libs/core/public_api.ts:L19-L19` → `core-libs/core/src/features-config/index.ts:L7-L15`.

Ricette pertinenti della Parte III:

- 50 — Ricette: configurazione, DI, interceptor, eventi: fornire i toggle (`provideFeatureToggles` / `provideFeatureTogglesFactory`) accanto agli altri override di configurazione; scegliere tra livello root e default.
- 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni: stili condizionati da toggle (`forFeature` + `useFeatureStyles`) negli override SCSS; toggle di default in una feature lib custom con `provideDefaultFeatureToggles`.
- 54 — API pubblica vs interna, upgrade, anti-pattern, checklist: cosa succede quando un toggle viene rimosso, come usare A09 e A10 per pianificare un upgrade, perché non estendere classi referenziando `featureToggles`.

Schema minimo (usa solo simboli esportati da `@spartacus/core`):

```ts
// Applicazione (root injector, modulo eager)
providers: [
  provideFeatureToggles({ a11yKeyboardAccessibleZoom: true }),
];

// Libreria propria: default sovrascrivibili dall'app
providers: [provideDefaultFeatureToggles({ a11yKeyboardAccessibleZoom: false })];

// Componente
export class MyComponent {
  private featureToggles = inject(FeatureToggles); // private: regola ESLint
  constructor() {
    useFeatureStyles('a11yKeyboardAccessibleZoom'); // se lo SCSS usa forFeature(...)
  }
}
```

Nota: una libreria esterna non può aggiungere **nuove chiavi** a `FeatureToggles` senza una module augmentation dell'interfaccia, che però non è esportata (core-libs/core/src/features-config/feature-toggles/index.ts:L7-L9). DA VERIFICARE: nel repo non c'è un pattern documentato per estendere `FeatureTogglesInterface` da librerie di terze parti.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **Ponte a senso unico.** `provideConfig({ features: { x: true } })` non arriva a `inject(FeatureToggles).x`: il ponte copia solo toggle → config (core-libs/core/src/features-config/feature-toggles/populate-feature-toggles-to-features-config.ts:L18-L33). In un'app che usa ancora la config legacy, `*cxFeature="'x'"` e `this.featureToggles.x` possono quindi dare risultati diversi **sulla stessa chiave**. Lo stesso vale per gli stili: `FeatureStylesService` legge `FeatureToggles` (core-libs/core/src/features-config/services/feature-styles.service.ts:L27-L29), quindi la classe `cxFeat_x` non compare.
2. **Allo stesso livello la config legacy vince.** Con `provideConfig({features:{x:true}})` e `provideFeatureToggles({x:false})` nello stesso livello, `isEnabled('x')` restituisce `true` qualunque sia l'ordine dei provider, perché il chunk del ponte entra presto nell'albero DI tramite `forRoot()` (core-libs/core/src/features-config/features-config.module.spec.ts:L84-L111). Tra livelli diversi vale invece ROOT > DEFAULT (core-libs/core/src/features-config/features-config.module.spec.ts:L37-L44).
3. **Due merge diversi.** I toggle si uniscono con `Object.assign` (shallow, vince l'ultimo); la config con `deepMerge` (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L18-L22 contro core-libs/core/src/config/config-tokens.ts:L10-L12). Per valori booleani la differenza non si vede, ma l'ordine di precedenza segue regole diverse (punto 2).
4. **Moduli lazy.** I token sono `providedIn: 'root'` e i provider "should be provided in the root injector" (core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L16-L19, core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L54-L57): un `provideFeatureToggles` dentro un modulo lazy non arriva né a `FeatureToggles` né a `Config.features`. `ConfigurationService` raccoglie dagli injector lazy solo `ConfigChunk` e `DefaultConfigChunk` (core-libs/core/src/config/services/configuration.service.ts:L58-L65), e il ponte che converte i toggle in `ConfigChunk` è registrato soltanto nel root da `forRoot()` (core-libs/core/src/features-config/features-config.module.ts:L33-L35, core-libs/core/src/features-config/feature-toggles/populate-feature-

toggles-to-features-config.ts:L21-L33). Solo un `provideConfig({ features: ... })` legacy in un modulo lazy viene unito in `Config` da `ConfigurationService`, se `disableConfigUpdates` è falso (core-libs/core/src/config/services/configuration.service.ts:L92-L106).

5. `*cxFeature` **non è reattiva.** Rivaluta solo quando cambia l'input; se cambia un toggle con la stessa stringa d'input, la vista non viene ri-renderizzata (core-libs/core/src/features-config/directives/feature.directive.ts:L19-L29).
6. **Stringhe interpretate come livelli.** `features: { x: '2.1' }` non è "truthy": viene letto come livello, quindi `isFeatureLevel(config, '2.1')`, che con il livello di default `'3.0'` restituisce `true` (core-libs/core/src/features-config/utils/feature-config-utils.ts:L57-L60, core-libs/core/src/features-config/features-config.module.ts:L28-L40). Con `forRoot('')` il livello diventa `'*'`, cioè tutti i livelli (core-libs/core/src/features-config/features-config.module.ts:L38-L38).
7. **Stili che non si accendono.** Se un componente non chiama `useFeatureStyles` nel costruttore, i blocchi `forFeature` restano spenti anche con il toggle a `true` (core-libs/core/src/features-config/utils/use-feature-styles.ts:L11-L31). La chiamata fuori da un injection context lancia `MISSING_INJECTION_CONTEXT` (angular:packages/core/src/di/contextual.ts:L77-L88). Un indizio utile: se la classe compare solo quando è montato *un altro* componente che registra lo stesso flag, significa che manca la registrazione nel componente stilizzato.
8. **Contatori anche per flag spenti.** `registerUsage` incrementa sempre; `unregisterUsage` con contatore 0 avvisa solo in dev mode (core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L115).
9. **Nessuna reazione alle mutazioni.** Se un toggle passa da `true` a `false` dopo `addClass`, `unregisterUsage` salta `removeClass` perché `isEnabled(feature)` è falso (core-libs/core/src/features-config/services/feature-styles.service.ts:L111-L113), e in ogni caso `getCssClass` restituirebbe `undefined`: la classe **non** viene rimossa (core-libs/core/src/features-config/services/feature-styles.service.ts:L144-L150, core-libs/core/src/features-config/services/feature-styles.service.ts:L158-L162). In produzione `FeatureToggles` è un oggetto costruito una sola volta, quindi il caso riguarda soprattutto i test.
10. **Il "root" non è** `<html>`. Le classi finiscono su `compRef.location.nativeElement`, cioè l'elemento del componente di bootstrap (core-libs/core/src/features-config/services/feature-styles.service.ts:L65-L66). Per gli stili su `:root` servono `currentSelector` (core-libs/styles/scss/root.scss:L36-L44) e l'aggiunta manuale su `document.documentElement` in `StorefrontComponent` (core-libs/storefront/layout/main/storefront.component.ts:L141-L147).
11. **App nuova contro app migrata.** `ng add` accende tutti i 107 toggle (core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L84-L86), mentre i default della libreria ne hanno 75 a `false` (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1013-L1121). Due progetti con la stessa versione di Spartacus possono quindi avere DOM e stili diversi.
12. **Rimozione di un toggle = errore di compilazione.** Chi referenzia una chiave rimossa non compila più: è voluto (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L10). È anche il motivo per cui `FeatureToggles` va iniettato `private` (tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L11-L21). Se una sottoclasse cliente legge `this.featureToggles.x` da un campo `protected`, la rimozione di `x` rompe il suo codice.
13. `provideAuthorizationCodeFlowByDefault` **e** `inject()`**.** La funzione chiama `inject()` nel proprio corpo, quindi va invocata in un injection context; inoltre `authorizationCodeFlowByDefault ?? enable` ignora `enable` ogni volta che il toggle è definito, e il default lo definisce sempre (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L37-L46).
14. **Mock nei test.** `MockFeatureDirective` di storefront ignora i valori e mostra sempre il ramo non negato (core-libs/storefront/shared/test/mock-feature-directive.ts:L16-L22). `provideMockFeatureToggles` funziona con `*cxFeature` solo se il token `FeatureConfigService` ha la stessa identità di classe; da qui l'a-

lias di `@spartacus/core` verso il sorgente in checkout (feature-libs/checkout/vitest.config.ts:L35-L42).

15. **Non esiste una pipe** per i toggle (nessuna delle pipe del repo, es. `cxTranslate` in core-libs/core/src/i18n/translate.pipe.ts:L25-L28 o `cxArgs` in feature-libs/asm/core/utils/args/args.pipe.ts:L33-L33, legge `FeatureToggles`). Nei template si usa `*cxFeature`; per valori calcolati si espone un campo dal componente letto da `FeatureToggles`.

---

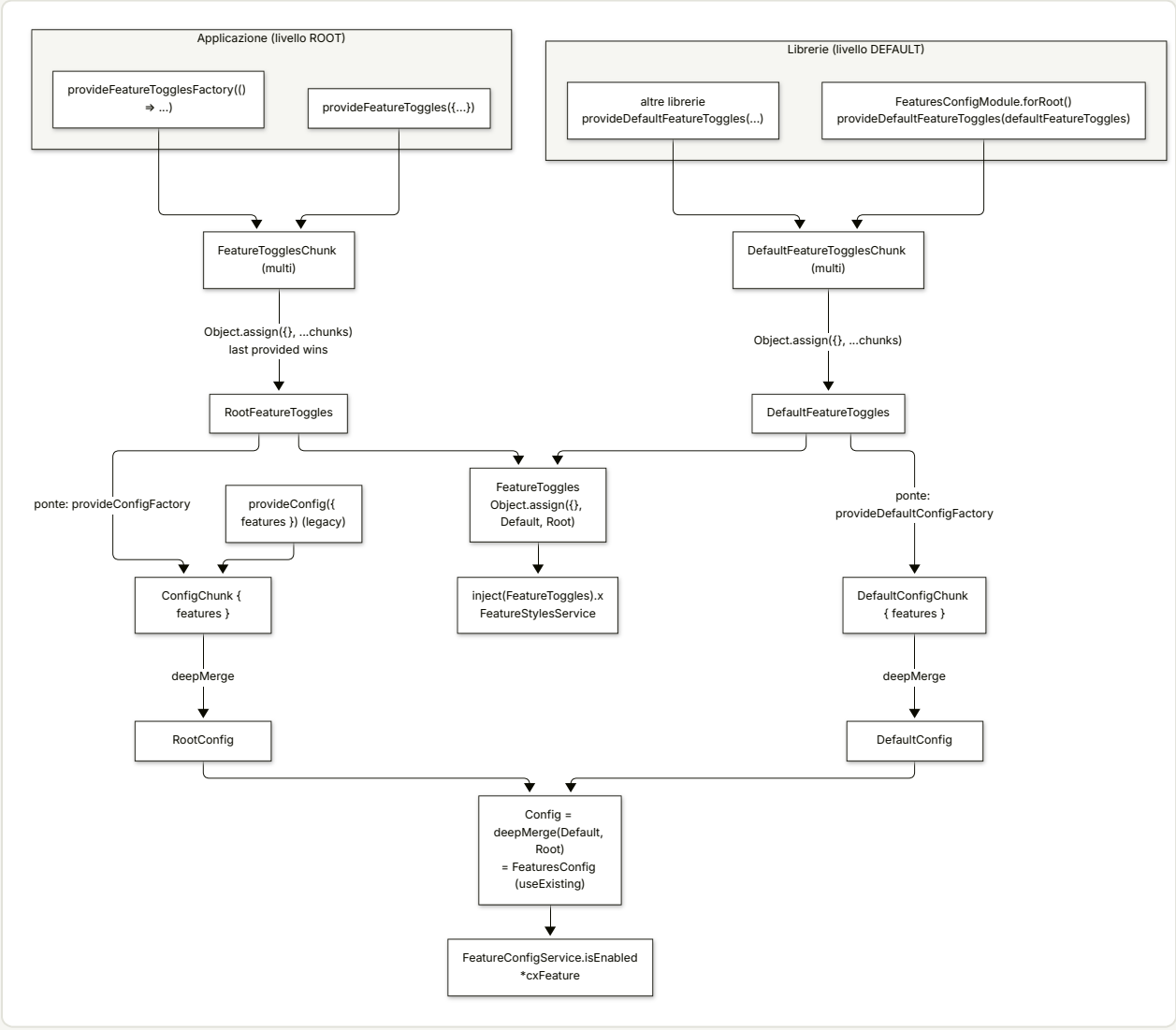
10. Come fare debug

- **Valore effettivo di un toggle.** In un componente o in un test: `inject(FeatureToggles)` restituisce l'oggetto finale già unito (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L15-L24). Per capire *da dove* arriva un valore, confronta `inject(DefaultFeatureToggles)` e `inject(RootFeatureToggles)`: sono token pubblici (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L33-L58).
- **Divergenza tra template e TS.** Se `*cxFeature` e `this.featureToggles.x` non concordano, cerca `provideConfig(` con `features` nell'app (trappole 1 e 2). Il valore che vede la direttiva è `inject(Config).features.x`, letto da `isFeatureEnabled` (core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L65).
- **Classe CSS assente.** Ispeziona l'elemento root dell'app (non `<html>`) e cerca `cxFeat_<chiave>`. Se manca: (a) il toggle è falsy in `FeatureToggles` (core-libs/core/src/features-config/services/feature-styles.service.ts:L120-L122); (b) nessun componente montato ha chiamato `useFeatureStyles('<chiave>')` (il dossier conta 56 chiamate: `git grep -n "useFeatureStyles('<chiave>')"`); (c) il componente è stato distrutto e il contatore è tornato a 0 (core-libs/core/src/features-config/services/feature-styles.service.ts:L103-L115). In dev mode, il warning `Feature flag CSS: "<chiave>" is already not used` indica una de-registrazione senza registrazione corrispondente (core-libs/core/src/features-config/services/feature-styles.service.ts:L103-L108).
- **Regola SCSS che non si applica pur con la classe presente.** Controlla il CSS compilato: con `ancestorSelector` la regola è `.cxFeat_x <selettore>` e richiede che l'elemento stilizzato stia *dentro* l'elemento root; per stili sull'elemento root stesso serve `currentSelector` (core-libs/styles/scss/_features.scss:L51-L61). Un errore di compilazione `forFeature: $classTarget must be either ...` indica un secondo argomento sbagliato (core-libs/styles/scss/_features.scss:L46-L48).
- **Catalogo e uso di una chiave.** A09 riporta per ogni chiave le righe di interfaccia e default e i conteggi d'uso in TS/HTML/SCSS. Per i dettagli, `git grep -n "'<chiave>'"` trova sia `*cxFeature` sia `useFeatureStyles`/`forFeature`.
- **Deprecazioni nel proprio codice.** A10 elenca i 165 `@deprecated` con simbolo e riferimento. Per le migrazioni automatiche tra versioni, vedi 43 — Schematics e migrazioni.
- **Nei test.** Usa `provideMockFeatureToggles({...})` e poi `TestBed.inject(MockFeatureTogglesController).set(key, value)` seguito da `fixture.detectChanges()`, come nell'esempio JSDoc (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L39-L58). Se `*cxFeature` non reagisce, verifica che `FeatureConfigService` sia la stessa classe di quella sostituita dal mock (trappola 14).

11. Domande di autoverifica

1. **Perché** `FeatureTogglesInterface` **non ha una index signature?** Perché una chiave rimossa deve causare un errore di compilazione nel codice cliente, così chi la usa sa di dover adattare il codice. — `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11`
2. **Un'app fornisce** `provideConfig({ features: { x: true } })`**. Che valore restituisce** `inject(FeatureToggles).x`**?** Il default di libreria (o `undefined`), non `true`: il ponte copia solo `FeatureToggles` → `Config.features`, mai il contrario. — `core-libs/core/src/features-config/feature-toggles/populate-feature-toggles-to-features-config.ts:L18-L33`
3. **Con** `provideConfig({features:{x:true}})` **e** `provideFeatureToggles({x:false})` **nell'app, cosa restituisce** `FeatureConfigService.isEnabled('x')`**? E se si invertono i provider?** `true` in entrambi i casi: allo stesso livello `FeaturesConfig` vince indipendentemente dall'ordine. — `core-libs/core/src/features-config/features-config.module.spec.ts:L84-L102`
4. **Perché** `defaultAuthConfigFactory` **usa** `inject(FeatureToggles)` **e non** `FeatureConfigService`**?** Perché la factory contribuisce a `DefaultConfig`/`Config`, da cui dipende `FeatureConfigService`: si creerebbe un ciclo di DI. `FeatureToggles` non dipende da `Config`. — `core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L76-L83`, `tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L23-L26`
5. **Un componente ha** `@include forFeature('x')` **nel suo SCSS, il toggle** `x` **è** `true`**, ma lo stile non si applica. Causa più probabile?** Il componente non chiama `useFeatureStyles('x')` nel costruttore, quindi `FeatureStylesService` non aggiunge `cxFeat_x` all'elemento root. — `core-libs/core/src/features-config/utils/use-feature-styles.ts:L11-L40`, `core-libs/core/src/features-config/services/feature-styles.service.ts:L89-L95`
6. **Perché** `FeatureStylesService.init` **scorre** `usagesCounter` **invece di limitarsi a salvare l'elemento root?** Perché i costruttori dei componenti (a partire dal root component) chiamano `registerUsage` prima che Angular invochi gli `APP_BOOTSTRAP_LISTENER`; in quel momento `addClass` non fa nulla perché `rootElement` non è ancora impostato. — `core-libs/core/src/features-config/services/feature-styles.service.ts:L65-L77`, `angular:packages/core/src/application/application_ref.ts:L775-L794`
7. **Quando si usa** `forFeature(..., 'currentSelector')`**?** Solo per gli stili dell'elemento root stesso; l'unico uso di produzione è in `root.scss` per `a11yPreventWindowsHighContrastOverride`. — `core-libs/styles/scss/_features.scss:L38-L40`, `core-libs/styles/scss/root.scss:L36-L44`
8. **Cosa restituisce** `isFeatureEnabled(config, 'x')` **se** `config.features.x === '2.1'` **e** `features.level` **vale** `'3.0'`**?** `true`: la stringa è interpretata come livello e `isInLevel('3.0', '2.1')` confronta 3 > 2. — `core-libs/core/src/features-config/utils/feature-config-utils.ts:L15-L34`, `core-libs/core/src/features-config/utils/feature-config-utils.ts:L57-L60`
9. **Perché** `FeatureToggles` **deve essere iniettato** `private`**?** Perché la rimozione di un toggle (il suo normale fine vita) diventerebbe un breaking change TypeScript per i clienti che lo referenziano da una sotto-classe. — `tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L11-L21`
10. **Un'app creata con** `ng add` **e una con i soli default di libreria partono con gli stessi toggle?** No: lo schematic imposta tutte le chiavi a `true`, mentre i default ne hanno 75 su 107 a `false`. — `core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L80-L93`, `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1013-L1121`

TAVOLA 42.1 **4.1 Costruzione di FeatureToggles e ponte verso Config.features**



43

Schematics e migrazioni di Spartacus

`@spartacus/schematics`: installazione in un'app `ng new` e migrazioni ai breaking change.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Spartacus non è un'applicazione da clonare: è un insieme di librerie npm che vanno *innestate* in un'applicazione Angular generata con `ng new`. L'innesto richiede decine di modifiche coordinate (moduli `app/spartacus/*`, config OCC, i18n, stili SCSS, `angular.json`, `package.json`, eventualmente SSR) e, a ogni release, modifiche inverse per adattare il codice del cliente ai breaking change. Il pacchetto `@spartacus/schematics` automatizza entrambe le cose con il meccanismo standard degli Angular schematics (vedi 08 — Schematics e builder per la parte Angular):

- **installazione** (`ng add @spartacus/schematics`, poi `ng add @spartacus/<feature-lib>`): scaffolding dell'app, configurazione, feature lazy-loaded, SSR;
- **generatori** di supporto (`add-cms-component`, `wrapper-module`, `ai-context`);
- **migrazioni** eseguite da `ng update` (`migrations.json`) e **modernizzazioni** manuali (`modernize-app-*`) per gli upgrade più invasivi.

Il pacchetto si dichiara con versione `221121.22.0` e registra la collection in `./src/collection.json` e le migrazioni in `./src/migrations/migrations.json` (core-libs/schematics/package.json:L2-L3, core-libs/schematics/package.json:L90-L90, core-libs/schematics/package.json:L43-L89). Nota: la versione dello schematics (`221121.22.0`) non coincide con la `2612.0.0` della root del monorepo; la cosa conta perché da quella versione dipendono `~<versione>` delle dipendenze installate e il "feature level" generato (core-libs/schematics/src/shared/utils/package-utils.ts:L139-L149).

Il pacchetto esporta anche utility per gli schematics delle altre librerie, con un avvertimento esplicito: "it exports utils for other Spartacus libs' schematics, but is not intended as a public API for external use. It may change without deprecation." (core-libs/schematics/index.ts:L7-L13). Ogni feature lib (es. `@spartacus/checkout`) ha infatti una propria piccola collection che riusa queste utility (feature-libs/checkout/schematics/add-checkout/index.ts:L13-L43).

Infine, un pezzo recente: il pacchetto `@spartacus/skills` e lo schematic `ai-context`, che copiano nel progetto del cliente una "skill" per assistenti AI (`.claude/skills/...`, `.agents/skills/...`), più una *living migration* che la tiene aggiornata a ogni `ng update` (core-libs/schematics/src/ai-context/index.ts:L16-L22, core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L22-L31).

> Nota: la semantica del devkit e della CLI citata nel capitolo (`chain`, task, `--dry-run`, selezione delle migrazioni di `ng update`, `ng-add.save`) è verificata sui sorgenti pubblicati di `@angular-devkit/schematics` e `@angular/cli` 21.2.23 (prefisso `deps:devkit/…`); vedi §3.9, §6 e le trappole 8, 16, 18 e 19 di §9.

Capitoli collegati: i feature toggle generati da `ng add` in 42 — Feature toggle e deprecazioni; lazy loading e `featureModules` in 12 — Feature module e lazy loading; `provideConfig` in 11 — Configurazione; lo stack SSR che `add-ssr` imposta in 39 — SSR ottimizzato; stili in 38 — Stili e theming; test tooling in 44 — Test tooling; strategia di upgrade in 54 — Upgrade safe.

2. Mappa dei file

PATH	RUOLO
`core-libs/schematics/package.json`	versione `221121.22.0`, `ng-add.save: devDependencies`, `ng-update.migrations`, `packageGroup` di tutte le `@spartacus/*`, script `build/clean/copy-feature-toggles/test`
`core-libs/schematics/index.ts`	"public API" (non stabile): `SpartacusOptions` + `src/shared/index`
`core-libs/schematics/copy-feature-toggles.js`	pre-build: copia `feature-toggles.ts` da core in `src/feature-toggles.copied-from-core-lib.ts` (git-ignored)
`core-libs/schematics/jest.schematics.config.js`, `core-libs/schematics/setup-jest.ts`, `core-libs/schematics/project.json`	test Jest (`*_spec.ts`), target Nx `test-jest`
`core-libs/schematics/src/collection.json`	9 schematics: `ng-add`, `add-spartacus`, `add-ssr`, `add-cms-component`, `wrapper-module`, `ai-context`, 3 × `modernize-app-*`
`core-libs/schematics/src/dependencies.json`	generato da `tools/schematics/dependency-collector.ts`: peer deps di ogni lib + deps della root
`core-libs/schematics/src/ng-add/index.ts`	`ng-add` = `add-spartacus` (+ `add-ssr` se `ssr`)
`core-libs/schematics/src/add-spartacus/index.ts`	`addSpartacus`: chain di ~20 rule; stili, budget, index.html, preprocessor, deps
`core-libs/schematics/src/add-spartacus/schema.json`, `core-libs/schematics/src/add-spartacus/schema.ts`	opzioni condivise da `ng-add`/`add-spartacus`/`add-ssr`
`core-libs/schematics/src/add-spartacus/{create-app-module,update-app-config,add-storefront-component-to-app-component,store,spartacus,spartacus-features,configuration,spartacus-feature-toggles,update-app-module}.ts`	singole rule di installazione
`core-libs/schematics/src/add-spartacus/ai-context.ts`	`addAiContext`, `scheduleAiContext`
`core-libs/schematics/src/add-ssr/index.ts` + `core-libs/schematics/src/add-ssr/*.ts`	`addSSR`: `@angular/ssr:ng-add` + adattamenti Spartacus
`core-libs/schematics/src/add-ssr/files/server.__typescriptExt__`	template `server.ts` (Express 5, `NgExpressEngineDecorator`)
`core-libs/schematics/src/add-cms-component/index.ts`	genera modulo + componente CMS (`ConfigModule.withConfig`)
`core-libs/schematics/src/wrapper-module/index.ts`	`generateWrapperModule`: "wrapper" per feature che devono caricarsi dopo un'altra
`core-libs/schematics/src/ai-context/index.ts`	copia + verifica SHA-256 della skill `spartacus-developer`
`core-libs/schematics/src/migrations/migrations.json`	25 migrazioni (2211.19.0, 2211.36.0, 221121.7.0, 221121.22.0)
`core-libs/schematics/src/migrations/<versione>/*`	wrapper per versione (`migrate()`) + `data/generated-*.migration.ts`

PATH	RUOLO
core-libs/schematics/src/migrations/mechanism/*	motori generici: constructor, methods/properties, removed public API, rename symbol, config, component, dependency management, missing packages
core-libs/schematics/src/migrations/add-or-update-ai-skills/*	living migration delle skill AI (prompt TTY, sentinel, scheduling)
core-libs/schematics/src/migrations/test/*	migrazioni fittizie usate dagli spec dei mechanism
core-libs/schematics/src/modernize-app-*/	schematics di modernizzazione (application builder, Angular 19, standalone bootstrap)
core-libs/schematics/src/shared/lib-configs/*-schematics-config.ts	un SchematicConfig per ogni feature (es. Checkout)
core-libs/schematics/src/shared/schematics-config-mappings.ts	SCHEMATICS_CONFIGS e mappe derivate
core-libs/schematics/src/shared/utils/lib-utils.ts	SchematicConfig, LibraryOptions, addLibraryFeature, stili/assets/i18n, sort, finalize
core-libs/schematics/src/shared/utils/feature-utils.ts	addFeatures, analyzeApplication
core-libs/schematics/src/shared/utils/graph-utils.ts, core-libs/schematics/src/shared/utils/dependency-utils.ts	grafo e ordinamento topologico di librerie e feature
core-libs/schematics/src/shared/utils/{program,tree-file-system,new-module-utils,import-utils}.ts	ts-morph sopra il Tree e manipolazione @NgModule
core-libs/schematics/src/shared/utils/{file-utils,module-file-utils}.ts	API AST TypeScript "classica", tipi di deprecazione, commitChanges
core-libs/schematics/src/shared/utils/{workspace-utils,package-utils}.ts	workspace, progetto di default, scaffolding, deps, isSsrUsed
tools/breaking-changes/*.ts	generatori dei dati di deprecazione
feature-libs/checkout/schematics/*	esempio di collection di feature lib
core-libs/skills/*	pacchetto @spartacus/skills (skill spartacus-developer)

3. Come funziona passo per passo

3.1 Pacchetto, build e collection

1. **Build senza ng-packagr.** npm run build rigenera dependencies.json (generate:deps) e compila con tsc -p ./tsconfig.schematics.json; clean esegue copy-feature-toggles (core-libs/schematics/package.json:L14-L16). generate:deps è ts-node ./tools/schematics/dependency-collector.ts (package.json:L62-L64).
2. dependencies.json. Il collector scrive core-libs/schematics/src/dependencies.json: per ogni lib sotto projects, core-libs, feature-libs, integration-libs raccoglie le peerDependencies, per il package.json root (name storefrontapp) le dependencies (tools/schematics/dependency-collector.ts:L21-L31, tools/schematics/dependency-collector.ts:L46-L76). Questo file è la "fonte di verità" delle versioni installate da ng add e da ng update.
3. **Feature toggle copiati.** copy-feature-toggles.js copia core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts in src/feature-toggles.copied-from-core-lib.ts (core-libs/

schematics/copy-feature-toggles.js:L5-L10), file ignorato da git (core-libs/schematics/.gitignore:L21-L22). Serve a addFeatureToggles senza dipendere a runtime da @spartacus/core (§3.3).

4. **Collection.** collection.json registra 9 schematics; ng-add è private/hidden con alias install, gli altri hanno alias brevi (spa, ssr, cms, wrapper, ai) (core-libs/schematics/src/collection.json:L3-L53). ng-add, add-spartacus e add-ssr condividono lo stesso ./add-spartacus/schema.json (core-libs/schematics/src/collection.json:L7-L7, core-libs/schematics/src/collection.json:L15-L15, core-libs/schematics/src/collection.json:L21-L21); wrapper-module non dichiara uno schema (core-libs/schematics/src/collection.json:L30-L34).
5. ng-add è un semplice chain di schematic('add-spartacus') e, se options.ssr, schematic('add-ssr') (core-libs/schematics/src/ng-add/index.ts:L17-L24).

3.2 Opzioni (Schema / LibraryOptions)

- LibraryOptions è la base comune: project, lazy, features, debug, internal.existingSpartacusApplication, options (core-libs/schematics/src/shared/utils/lib-utils.ts:L85-L110).
- Schema di add-spartacus aggiunge baseUrl, occPrefix, baseSite, currency, language, urlParameters, useMetaTags, featureLevel (@deprecated), overwriteAppComponent, ssr, theme, aiTools; AiTool = 'claude' | 'agents' (core-libs/schematics/src/add-spartacus/schema.ts:L9-L27).
- Lo schema JSON propone 16 feature di default con x-prompt a lista (core-libs/schematics/src/add-spartacus/schema.json:L14-L34, core-libs/schematics/src/add-spartacus/schema.json:L94-L96) e default lazy: true, baseUrl: "https://localhost:9002", overwriteAppComponent: true, ssr: false, debug: false, aiTools: ["agents"] (core-libs/schematics/src/add-spartacus/schema.json:L333-L346, core-libs/schematics/src/add-spartacus/schema.json:L376-L385, core-libs/schematics/src/add-spartacus/schema.json:L390-L397).

3.3 add-spartacus: la catena di installazione

addSpartacus calcola **prima** le feature con le dipendenze transitive (analyzeCrossFeatureDependencies) e l'elenco delle dipendenze npm (prepareDependencies), poi restituisce un chain (core-libs/schematics/src/add-spartacus/index.ts:L531-L590). Le rule, nell'ordine:

1. detectAndSetupAppStructure — pretende un'app standalone con app.config.ts e il naming "2016" (app.component.ts); altrimenti SchematicsException con il consiglio di ricreare l'app con --file-name-style-guide=2016 e senza --standalone=false (core-libs/schematics/src/add-spartacus/index.ts:L324-L369).
2. createAppModule — crea un app.module.ts vuoto (@NgModule({})) accanto ad app.component.ts (core-libs/schematics/src/add-spartacus/create-app-module.ts:L22-L81). Spartacus continua quindi a configurarsi in un NgModule, agganciato all'app standalone tramite:
3. updateAppConfig — aggiunge provideHttpClient(withFetch(), withInterceptorsFromDi()) e importProvidersFrom(AppModule) ai providers di appConfig, evitando duplicati per confronto testuale (core-libs/schematics/src/add-spartacus/update-app-config.ts:L19-L46, core-libs/schematics/src/add-spartacus/update-app-config.ts:L55-L110).
4. addStorefrontComponentToAppComponent — importa StorefrontComponent e **sostituisce** l'array imports del @Component (removeOldImports: true) (core-libs/schematics/src/add-spartacus/add-storefront-component-to-app-component.ts:L36-L48, core-libs/schematics/src/add-spartacus/add-imports-to-component-decorator.ts:L56-L89).
5. analyzeApplication — imposta options.internal.existingSpartacusApplication se esiste già SpartacusFeaturesModule e, per app esistenti, verifica che i "marker module" degli importAfter siano installati (core-libs/schematics/src/shared/utils/feature-utils.ts:L279-L370).

6. setupStoreModules — `StoreModule.forRoot({})` ed `EffectsModule.forRoot([])` in `app.module.ts` (core-libs/schematics/src/add-spartacus/store.ts:L48-L70).
7. scaffoldStructure — via `ensureModuleExists` crea `app/spartacus/spartacus.module.ts` (dichiarato in `app`), `spartacus-features.module.ts` e `spartacus-configuration.module.ts` (dichiarati in `spartacus`) (core-libs/schematics/src/shared/utils/workspace-utils.ts:L170-L198).
8. setupSpartacusModule — `BaseStorefrontModule` in `imports` (order 0) e `exports` di `SpartacusModule` (core-libs/schematics/src/add-spartacus/spartacus.ts:L53-L84).
9. setupSpartacusFeaturesModule — moduli core/storefront fissi: `AuthModule.forRoot()`, `LogoutModule`, `LoginRouteModule`, moduli CMS base, consensi, Product, `CostCenterOccModule`, event module, `ExternalRoutesModule.forRoot()` (core-libs/schematics/src/add-spartacus/spartacus-features.ts:L51-L276).
10. addSpartacusConfiguration — in `spartacus-configuration.module.ts`: `provideConfigFactory(layoutConfigFactory)`, `provideConfig(mediaConfig)`, `...defaultCmsContentProviders`, poi config OCC, site context, i18n e `features.level` (core-libs/schematics/src/add-spartacus/configuration.ts:L83-L122, core-libs/schematics/src/add-spartacus/configuration.ts:L157-L208). Dettagli: con `useMetaTags` il `baseUrl` è omesso e `prefix` compare solo se `occPrefix` è valorizzato (core-libs/schematics/src/add-spartacus/configuration.ts:L210-L221); le CSV di site context passano per `parseCSV` (valute maiuscole, lingue minuscole) (core-libs/schematics/src/add-spartacus/configuration.ts:L124-L151, core-libs/schematics/src/shared/utils/transform-utils.ts:L7-L19); i18n usa `translationsEn`, `translationChunksConfig`, `fallbackLang: 'en'` (core-libs/schematics/src/add-spartacus/configuration.ts:L183-L196, core-libs/schematics/src/add-spartacus/configuration.ts:L223-L231); `features.level` è `featureLevel` o il `major.minor` dello schematics, funzione marcata `@deprecated` (core-libs/schematics/src/add-spartacus/configuration.ts:L233-L242, core-libs/schematics/src/shared/utils/package-utils.ts:L147-L149).
11. addFeatureToggles — `provideFeatureToggles({...})` con **tutte** le chiavi di `defaultFeatureToggles` a `true`, lette con `require` dal file copiato (core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L62-L113, core-libs/schematics/src/shared/constants.ts:L1176-L1177).
12. updateAppModule — rimuove l'`AppRoutingModule` locale (import e file `app-routing.module.ts`) e importa quello di `@spartacus/storefront` con order 2 (core-libs/schematics/src/add-spartacus/update-app-module.ts:L80-L145).
13. **Stili** — `createStylesConfig` crea `styles-config.scss` con `$styleVersion` (core-libs/schematics/src/add-spartacus/index.ts:L70-L86, core-libs/schematics/src/shared/utils/styling-utils.ts:L10-L13); `installStyles` prende il primo elemento di `build.options.styles` e, se è un `.scss`, vi appende gli import di styles-config, `@spartacus/styles/scss/core`, Bootstrap vendorizzato sotto `@spartacus/styles/vendor/bootstrap/scss/*`, `@spartacus/styles/index` e, opzionalmente, il tema; se non è un `.scss` si limita a un warning (core-libs/schematics/src/add-spartacus/index.ts:L89-L89, core-libs/schematics/src/add-spartacus/index.ts:L120-L128, core-libs/schematics/src/add-spartacus/index.ts:L102-L187).
14. updateMainComponent — `<cx-storefront/>` in `app.component.html`, sostituendo tutto se `overwriteAppComponent` (core-libs/schematics/src/add-spartacus/index.ts:L189-L229). Con `useMetaTags`, `updateIndexFile` aggiunge le meta `occ-backend-base-url`/`media-backend-base-url` (core-libs/schematics/src/add-spartacus/index.ts:L231-L254).
15. increaseBudgets — `maximumError: '3.5mb'` per il budget `initial` in production (core-libs/schematics/src/add-spartacus/index.ts:L256-L315).
16. createStylePreprocessorOptions — `node_modules/` negli `includePaths` di build e test, `sass.silenceDeprecations` solo in build (core-libs/schematics/src/add-spartacus/index.ts:L371-L464), con `'import'` silenziato e la motivazione CXSPA-447 (core-libs/schematics/src/shared/utils/workspace-utils.ts:L209-L243).
17. addFeatures — per ogni feature `addLibraryFeature` + eventuali `wrapper-module` (§3.4).

18. `addAiContext` — `@spartacus/skills` in devDependencies (`overwrite: false`) se `aiTools` non è vuoto (core-libs/schematics/src/add-spartacus/ai-context.ts:L27-L48).
19. **Dipendenze** — sotto-chain: `addPackageJsonDependencies(prepareDependencies(features))`, forzatura di `rxjs` alla versione Spartacus, `installPackageJsonDependencies()`, `replaceCaretWithTildeForSpartacusDependencies` (core-libs/schematics/src/add-spartacus/index.ts:L571-L585). `prepareDependencies` unisce le core deps (`core`, `storefront`, `assets`, `styles`), le librerie delle feature (con `~<versione schematics>`) e le terze parti (core-libs/schematics/src/add-spartacus/index.ts:L466-L480, core-libs/schematics/src/shared/utils/package-utils.ts:L256-L294). Il passaggio ^→~ è motivato dalla scelta, dalla 2211.19, di rilasciare breaking change nei "minor" (core-libs/schematics/src/add-spartacus/index.ts:L482-L529).
20. `scheduleAiContext` — accoda `NodePackageInstallTask` e, dipendente da esso, `RunSchematicTask('ai-context', …)` (core-libs/schematics/src/add-spartacus/ai-context.ts:L54-L76).
21. `finalizeInstallation` — per un'app già Spartacus, warning "🚨 Detected Spartacus installation..." con l'elenco delle feature da verificare e ordinare (core-libs/schematics/src/shared/utils/lib-utils.ts:L897-L909).

3.4 Feature: `SchematicConfig`, `addFeatures`, `addLibraryFeature`

Ogni feature installabile è descritta da un oggetto dichiarativo `SchematicConfig`: `library {featureName, mainScope, featureScope?, b2b?}`, `folderName`, `moduleName`, `featureModule` (singolo o array), `rootModule?`, `lazyLoadingChunk?`, `i18n?`, `styles?`, `assets?`, `customConfig?`, `dependencyFeatures?`, `importAfter?` (core-libs/schematics/src/shared/utils/lib-utils.ts:L115-L199). Tutte le config sono elencate in `SCHEMATICS_CONFIGS`, che "*Must* be updated when adding a new schematics library or a feature" (core-libs/schematics/src/shared/schematics-config-mappings.ts:L99-L198); `generateMappings()` ne ricava 4 mappe (`libraryFeatureMapping`, `featureFeatureModuleMapping`, `featureRootModuleMapping`, `featureSchematicConfigMapping`) (core-libs/schematics/src/shared/schematics-config-mappings.ts:L203-L286).

1. **Closure delle dipendenze.** `analyzeCrossFeatureDependencies` raccoglie ricorsivamente le `dependencyFeatures` (es. Digital Payments trascina User-Account, User-Profile, Cart, Order, Checkout) e le ordina (core-libs/schematics/src/shared/utils/dependency-utils.ts:L29-L60).
2. **Ordine globale.** All'import di `graph-utils.ts` vengono calcolati, come costanti di modulo, `crossLibraryInstallationOrder` e `crossFeatureInstallationOrder` (core-libs/schematics/src/shared/utils/graph-utils.ts:L41-L49) con l'algoritmo di Kahn, che lancia `Circular dependency detected.` sui cicli e restituisce l'ordine **invertito** (dipendenze prima) (core-libs/schematics/src/shared/utils/graph-utils.ts:L85-L135). Il grafo delle librerie usa le dipendenze `@spartacus/*` di `dependencies.json` (core-libs/schematics/src/shared/utils/graph-utils.ts:L137-L166), quello delle feature un arco per ogni `dependencyFeatures` (core-libs/schematics/src/shared/utils/graph-utils.ts:L168-L184); `groupFeatures` avvicina le feature della stessa libreria (core-libs/schematics/src/shared/utils/graph-utils.ts:L51-L69). Le voci assenti dall'ordine finiscono in fondo (core-libs/schematics/src/shared/utils/lib-utils.ts:L878-L891).
3. `addFeatures`. In un'app nuova installa tutte le feature della closure; in un'app esistente solo `options.features` ("we don't want to force-install the dependent features"). Per ciascuna applica `customConfig(options).options` se c'è, accoda `addLibraryFeature` e uno `externalSchematic('@spartacus/schematics', 'wrapper-module', …)` per ogni `importAfter` (core-libs/schematics/src/shared/utils/feature-utils.ts:L102-L154).
4. `addLibraryFeature`. Se manca la struttura fa scaffolding, poi `handleFeature`, stili e assets (core-libs/schematics/src/shared/utils/lib-utils.ts:L239-L273). `handleFeature` crea `<moduleName>-feature.module.ts` in `app/spartacus/features/<folder>`, dichiarato in `spartacus-features`, e aggiunge root mo-

dule, feature module, traduzioni, custom config e, per b2b, `configureB2bFeatures` (core-libs/schematics/src/shared/utils/lib-utils.ts:L293-L326).

5. **Lazy vs eager.** Con `lazy` il feature module diventa `provideConfig(<CmsConfig>{ featureModules: { [CHUNK]: { module: () => import('…').then((m) => m.X) } } })`, altrimenti è importato staticamente (core-libs/schematics/src/shared/utils/lib-utils.ts:L397-L446). Questo è esattamente il formato consumato dal lazy loading descritto in 12.
6. **Traduzioni, stili, assets, B2B.** `addFeatureTranslations` → `provideConfig(<I18nConfig>{…})` (core-libs/schematics/src/shared/utils/lib-utils.ts:L448-L494); `addLibraryStyles` → `<sourceRoot>/styles/spartacus/<file>.scss` aggiunto a `styles` di build e test (core-libs/schematics/src/shared/utils/lib-utils.ts:L641-L728); `addLibraryAssets` → glob `./node_modules/@spartacus/<input>` in `assets` (core-libs/schematics/src/shared/utils/lib-utils.ts:L541-L629); per b2b `provideConfig(defaultB2bOccConfig)` + dipendenza `@spartacus/setup` (core-libs/schematics/src/shared/utils/lib-utils.ts:L786-L841, core-libs/schematics/src/shared/utils/config-utils.ts:L123-L139).

Esempio concreto: `CHECKOUT_BASE_SCHEMATICS_CONFIG` (feature `Checkout`, `CheckoutModule` da `@spartacus/checkout/base`, root `CheckoutRootModule`, chunk `CHECKOUT_FEATURE`, dipendenza da `Order`) e `CHECKOUT_B2B_SCHEMATICS_CONFIG` (`b2b: true`, stesso `moduleName`, `importAfter` per caricare `CheckoutB2BModule` dopo `CheckoutModule`) (core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L40-L125).

3.5 `wrapper-module`: caricare una feature "dopo" un'altra

Alcune feature (es. Checkout B2B) estendono un'altra feature e devono essere importate **nello stesso chunk lazy, dopo** di essa. Il wrapper risolve il problema creando un modulo che importa staticamente entrambi, e ri-puntando il dynamic import a quel modulo. `generateWrapperModule` è un chain di sei rule (core-libs/schematics/src/wrapper-module/index.ts:L527-L541):

1. `checkWrapperModuleExists` — se un file importa già staticamente il marker module, il suo path finisce in `options.internal.wrapperModulePath`: le rule successive leggono questa **mutazione** delle opzioni (core-libs/schematics/src/wrapper-module/index.ts:L89-L114).
2. `createWrapperModule` — crea `app/spartacus/features/<folder>/<name>-wrapper.module.ts` importandolo temporaneamente nel feature module, workaround perché ts-morph "does not 'see' the newly created TS file if it is not referenced anywhere" (core-libs/schematics/src/wrapper-module/index.ts:L167-L207). Il nome file toglie `module`/`Module` e aggiunge `-wrapper` (core-libs/schematics/src/shared/utils/lib-utils.ts:L328-L331).
3. `updateFeatureModule` — `import('@spartacus/user/profile').then(...)` diventa `import('./profile-wrapper.module').then(...)` e l'import temporaneo è rimosso (core-libs/schematics/src/wrapper-module/index.ts:L222-L336).
4. `removeLibraryDynamicImport` / `cleanupConfig` — toglie `module` dal provider della feature dipendente e, se l'oggetto resta `{}`, l'intero provider (core-libs/schematics/src/wrapper-module/index.ts:L338-L405, core-libs/schematics/src/wrapper-module/index.ts:L413-L427).
5. `updateWrapperModule` ×2 — import statico di marker e poi feature module nel wrapper (core-libs/schematics/src/wrapper-module/index.ts:L457-L521).

Le opzioni comprendono `markerModuleName`, `featureModuleName` e `internal.wrapperModulePath` ("Should not be set by the user") (core-libs/schematics/src/wrapper-module/schema.ts:L9-L38).

3.6 `add-ssr`

`addSSR` riusa lo schematic ufficiale di Angular e poi lo "piega" verso lo stack SSR di Spartacus (`core-libs/schematics/src/add-ssr/index.ts:L437-L465`):

1. dipendenze `@spartacus/setup`, `@angular/platform-server`, `@angular/ssr`, con le versioni prese da `dependencies.json` alla chiave `storefrontapp` (`core-libs/schematics/src/add-ssr/index.ts:L59-L59`, `core-libs/schematics/src/add-ssr/index.ts:L131-L153`);
2. `externalSchematic('@angular/ssr', 'ng-add')`, che a sua volta delega a `@schematics/angular:ssr` (`deps:angular-ssr-21.2.23/package/schematics/ng-add/index.js:L11-L11`);
3. rimozione di `build.options.outputMode` ("This could be removed when we migrate to new SSR API") (`core-libs/schematics/src/add-ssr/index.ts:L362-L412`) e di `app.routes.server.ts` ("not supported by Spartacus SSR") (`core-libs/schematics/src/add-ssr/index.ts:L414-L435`);
4. `scripts["build:ssr"] = "ng build"` per la build CCv2 (TODO CXSPA-6466) (`core-libs/schematics/src/add-ssr/index.ts:L155-L181`);
5. `AppServerModule` vuoto in `app.module.server.ts` (`core-libs/schematics/src/add-ssr/create-app-server-module.ts:L22-L77`); in `app.config.server.ts` rimuove `withRoutes(serverRoutes)` e aggiunge `importProvidersFrom(AppServerModule)` (`core-libs/schematics/src/add-ssr/update-app-config-server.ts:L17-L58`, `core-libs/schematics/src/add-ssr/update-app-config-server.ts:L104-L140`); in `app.module.server.ts` inserisce `...provideServer({ serverRequestOrigin: process.env['SERVER_REQUEST_ORIGIN'] })` da `@spartacus/setup/ssr` (`core-libs/schematics/src/add-ssr/index.ts:L61-L96`);
6. meta OCC in `index.html` e `provideClientHydration(withEventReplay(), withNoHttpTransferCache())` in `appConfig` — transfer cache "temporarily disabled" (CXSPA-10430); il meta `occ-backend-base-url` è aggiunto da `modifyIndexHtmlFile` (`core-libs/schematics/src/add-ssr/index.ts:L98-L117`), l'hydration da `updateAppConfigInSsr` (`core-libs/schematics/src/add-ssr/update-app-config-in-ssr.ts:L47-L77`, `core-libs/schematics/src/add-ssr/update-app-config-in-ssr.ts:L104-L149`). I providers sono individuati cercando una variabile chiamata `appConfig` / `serverConfig` con un object literal che ha `providers` array (`core-libs/schematics/src/add-ssr/get-providers-array-from-named-object.ts:L22-L64`);
7. merge del template `server.ts` con `MergeStrategy.Overwrite` (`core-libs/schematics/src/add-ssr/index.ts:L455-L458`): `NgExpressEngineDecorator.get(ngExpressEngine)`, `getOriginValidationMiddleware`, `index.server.html`, `trust proxy: loopback`, `defaultExpressErrorHandlers` (`core-libs/schematics/src/add-ssr/files/server.__typescriptExt__:L1-L63`), route regex Express 5 e porta di default `4000` (`core-libs/schematics/src/add-ssr/files/server.__typescriptExt__:L45-L73`);
8. `prerender: false`, nuova configurazione `noSsr` agganciata a `serve` production/development (`core-libs/schematics/src/add-ssr/index.ts:L183-L360`); install.

3.7 `add-cms-component`

Valida che `cmsComponentDataModel` sia presente se `cmsComponentData` (`core-libs/schematics/src/add-cms-component/index.ts:L356-L362`); se non viene passato `declareCmsModule` genera un nuovo modulo con `@schematics/angular:module`, poi il componente con `@schematics/angular:component`, quindi aggiorna modulo/componente/template (`core-libs/schematics/src/add-cms-component/index.ts:L398-L447`). La registrazione usa la vecchia API `ConfigModule.withConfig(<CmsConfig>{ cmsComponents: … })` (`core-libs/schematics/src/add-cms-component/index.ts:L114-L136`).

3.8 AI context: `@spartacus/skills` e schematic `ai-context`

1. `@spartacus/skills` (versione `221121.22.0`, nessun target Nx) contiene la sola skill `skills/spartacus-developer` (`SKILL.md` + `references/`) (`core-libs/skills/package.json:L2-L14`, `core-libs/skills/project.json:L1-L7`). `SKILL.md` ha frontmatter `name: spartacus-developer` e una Quick Reference di regole per chi

sviluppa su Spartacus (core-libs/skills/skills/spartacus-developer/SKILL.md:L1-L5, core-libs/skills/skills/spartacus-developer/SKILL.md:L32-L58).

2. Lo schematic `ai-context` normalizza `aiTools` (solo `claude`, `agents`, nell'ordine fisso) e risolve la sorgente: se `SPARTACUS_SKILLS_DIR` è definita viene usata **in esclusiva** (path inesistente → `null` → warning, anche se `@spartacus/skills` è installato) (core-libs/schematics/src/ai-context/index.ts:L85-L88); altrimenti `require.resolve('@spartacus/skills/package.json', { paths: [cwd] })`, poi il fallback `node_modules/@spartacus/skills/skills/spartacus-developer`; se non trova nulla emette un warning ed esce (core-libs/schematics/src/ai-context/index.ts:L41-L111).
3. Legge i file dal **file system reale** (`fs.readdirSync`/`readFileSync`), li ordina e li scrive nel `Tree` (create/overwrite) (core-libs/schematics/src/ai-context/index.ts:L113-L132, core-libs/schematics/src/ai-context/index.ts:L178-L191).
4. Con `deleteBeforeCopy` svuota prima la destinazione e, se resta qualcosa, lancia "so the transactional Tree commit is aborted" (core-libs/schematics/src/ai-context/index.ts:L159-L176).
5. `verifySkillTree` confronta lo SHA-256 di ogni file e rifiuta file extra sotto la destinazione (core-libs/schematics/src/ai-context/index.ts:L193-L254).

3.9 `ng update`: struttura delle migrazioni

1. `ng-update.migrations` punta a `./src/migrations/migrations.json` (core-libs/schematics/package.json:L46-L47). La CLI legge `packageNode['ng-update'].migrations` e chiama `executeMigrations(..., from, options.to || packageNode.version, ...)` (deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L252-L253, deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L303-L303); seleziona le voci la cui `version` soddisfa il range `>from <=to` e le esegue in ordine di versione e poi di **nome** (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L107-L118, deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L127-L127).
2. `migrations.json` contiene 25 voci: 10 per `2211.19.0`, 4 per `2211.36.0`, 10 per `221121.7.0` e 1 per `221121.22.0` (core-libs/schematics/src/migrations/migrations.json:L3-L127); le chiavi hanno un prefisso `00-`…`09-` per versione (core-libs/schematics/src/migrations/migrations.json:L3-L52): poiché la CLI ordina per versione e poi per nome, è questo prefisso a fissare l'ordine d'esecuzione dentro una stessa versione (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L127-L127).
3. Ogni voce punta a `./<versione>/<nome>/<file>#migrate`. Per le migrazioni "di codice" il `migrate()` per versione è un wrapper sottilissimo che passa i dati generati al mechanism (core-libs/schematics/src/migrations/221121_7/constructor-deprecations/constructor-deprecations.ts:L7-L19).
4. I dati (`data/generated-*.migration.ts`, "Generated file, don't update manually.") sono prodotti da `tools/breaking-changes/generate-*.ts` per `NEW_VERSION = '221121_7'`, con prefisso di commento `// TODO:Spartacus -` (core-libs/schematics/src/migrations/221121_7/constructor-deprecations/data/generated-constructor.migration.ts:L7-L9, tools/breaking-changes/common.ts:L12-L19). Dimensioni dei file dati 221121_7 (`wc -l`, righe totali compresi header e import): 191 costruttori, 350 metodi/proprietà, 306 API rimosse, 17 rename (una sola voce). L'unico rename sposta `cdsTranslationChunksConfig` da `@spartacus/cds` a `@spartacus/cds/assets` (core-libs/schematics/src/migrations/221121_7/rename-symbol/data/generated-rename-symbols.migration.ts:L11-L17).
5. Alcune migrazioni 2211_19 hanno dati vuoti (`config-deprecations`, `dependency-management`, `missing-packages`) (core-libs/schematics/src/migrations/2211_19/config-deprecations/config-deprecations.ts:L11-L11, core-libs/schematics/src/migrations/2211_19/dependency-management/dependency-management.ts:L10-L10, core-libs/schematics/src/migrations/2211_19/missing-packages/missing-

packages.ts:L13-L13); css si limita a un warning con link (core-libs/schematics/src/migrations/2211_19/css/css.ts:L10-L16).

Catalogo delle migrazioni di configurazione (non guidate da dati):

VERSIONE	MIGRAZIONE	COSA FA	RIFERIMENTO
2211.19.0	ssr	solo con legacy server builder: ripristina server.ts.bak, zone.js/dist/zone-node→zone.js/node, @nguniversal/express-engine→@spartacus/setup/ssr, elimina express.tokens.ts	core-libs/schematics/src/migrations/2211_19/ssr/ssr.ts:L11-L15, core-libs/schematics/src/migrations/2211_19/update-ssr/update-ssr-files.ts:L25-L137
2211.19.0	angular-json-styling	riusa createStylePreprocessorOptions()	core-libs/schematics/src/migrations/2211_19/angular-json-styles/angular-json-styles.ts:L7-L14
2211.36.0	ssr	index.html → join(serverDistFolder, 'index.server.html') (se isSsrUsed)	core-libs/schematics/src/migrations/2211_36/ssr/update-ssr/update-server-files.ts:L15-L149
2211.36.0	bootstrap	disinstalla bootstrap (npm/yarn/pnpm) e riscrive gli import verso @spartacus/styles/vendor/bootstrap/scss/*	core-libs/schematics/src/migrations/2211_36/bootstrap/bootstrap.ts:L16-L131, core-libs/schematics/src/migrations/2211_36/bootstrap/replace-bootstrap-imports.ts:L9-L198
2211.36.0	silence-sass-deprecations	silenceDeprecations: ['import']	core-libs/schematics/src/migrations/2211_36/silence-sass-deprecations/silence-sass-deprecations.ts:L14-L32
221121.7.0	angular-config	rimuove build.options.index	core-libs/schematics/src/migrations/221121_7/angular-config/index.ts:L13-L71
221121.7.0	angular-providers	provideBrowserGlobalErrorListeners() in app.module.ts	core-libs/schematics/src/migrations/221121_7/angular-providers/index.ts:L21-L88
221121.7.0	express-v5	express ^5.1.0, '*.*'→/.*\..*/, '*'→/.*/ (solo SSR)	core-libs/schematics/src/migrations/221121_7/ssr/express-v5/index.ts:L59-L109, core-libs/schematics/src/migrations/221121_7/ssr/express-v5/index.ts:L199-L261
221121.7.0	tsconfig-update	rimuove outDir/esModuleInterop/moduleResolution, module: "preserve", files: [] + references	core-libs/schematics/src/migrations/221121_7/tsconfig-update/index.ts:L10-L117
221121.7.0	move-zone-change-detection-to-main	sposta provideZoneChangeDetection da main.ts ad app.module.ts	core-libs/schematics/src/migrations/221121_7/move-zone-change-detection-to-main/index.ts:L11-L22
221121.7.0	build-ssr-script	ri-aggiunge build:ssr (solo SSR)	core-libs/schematics/src/migrations/221121_7/ssr/build-ssr-script/index.ts:L16-L73

VERSIONE	MIGRAZIONE	COSA FA	RIFERIMENTO
221121.22.0	`add-or-update-ai-skills`	living migration delle skill AI	`core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L32-L86`

3.10 I mechanism generici

- **Costruttori** — `migrateConstructorDeprecation` scorre tutti i `.ts` del sourceRoot del primo progetto; per ogni deprecazione candidata aggiunge e poi rimuove parametri, ricaricando il sorgente dopo ogni commit (`core-libs/schematics/src/migrations/mechanism/constructor-deprecations/constructor-deprecations.ts:L22-L79`). Un file è candidato se estende la classe deprecata importata dal path atteso, importa i tipi dei parametri e ha costruttore + `super()` corrispondenti (`core-libs/schematics/src/shared/utils/file-utils.ts:L352-L412`); `addConstructorParam` gestisce anche `@Inject(token)` e aggiorna `super(...)` (`core-libs/schematics/src/shared/utils/file-utils.ts:L532-L611`).
- **Metodi e proprietà** — commento `// TODO:Spartacus - '<old>' has been removed. Please try using '<new>' instead.` sopra gli usi (`core-libs/schematics/src/migrations/mechanism/methods-and-properties-deprecations/methods-and-properties-deprecations.ts:L20-L58`).
- **API rimosse** — commento sopra gli import dei simboli rimossi (`core-libs/schematics/src/migrations/mechanism/removed-public-api-deprecations/removed-public-api-deprecation.ts:L19-L53`).
- **Rename symbol** — mechanism su ts-morph (come `missing-packages`, `core-libs/schematics/src/migrations/mechanism/missing-packages/missing-packages.ts:L9-L9`, `core-libs/schematics/src/migrations/mechanism/missing-packages/missing-packages.ts:L30-L33`): sostituisce la dichiarazione di import (preservando l'alias) e chiama `saveAndFormat` (`core-libs/schematics/src/migrations/mechanism/rename-symbol/rename-symbol.ts:L14-L85`).
- **Config** — commento sopra ogni proprietà `propertyName` in un object literal (`core-libs/schematics/src/migrations/mechanism/config-deprecations/config-deprecation.ts:L17-L39`).
- **Componenti** — commenta nei template gli input/output rimossi e, nelle sottoclassi, le proprietà rimosse, usando `@angular/compiler` passato come parametro (`core-libs/schematics/src/migrations/mechanism/component-deprecations/component-deprecations.ts:L27-L156`). Allo SHA è usato solo dalle migrazioni di test: in `migrations.json` non c'è una migrazione component-deprecations (`core-libs/schematics/src/migrations/test/component-deprecations/component-deprecations.ts:L9-L31`).
- **Dependency management** — aggiorna con `overwrite: true` tutte le deps `@spartacus/*` (tranne `@spartacus/schematics`) secondo `dependencies.json` e installa (`core-libs/schematics/src/migrations/mechanism/dependency-management/dependency-management.ts:L28-L139`).
- **Missing packages** — avvisa se si importa da un pacchetto assente nel `package.json` (`core-libs/schematics/src/migrations/mechanism/missing-packages/missing-packages.ts:L18-L67`).

3.11 Living migration delle skill AI

`add-or-update-ai-skills` ha `version: "221121.22.0"` ma la sua JSDoc spiega che la versione "is updated to match the package version on every release so it always fires" (`core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L22-L31`, `core-libs/schematics/src/migrations/migrations.json:L123-L127`). Dato il range `>from <=to` della CLI, gira in ogni `ng update` che passa da una versione inferiore a quella allineata al pacchetto (allo SHA `221121.22.0`); non gira se `from` è già pari a quella versione (`deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L107-L118`).

1. Senza TTY (`stdin.isTTY && stdout.isTTY` falso), se trova skill installate (sentinel `SKILL.md` in `.claude/...` o `.agents/...`) le ri-sincronizza, altrimenti stampa un avviso (`core-libs/schematics/src/migrations/`

add-or-update-ai-skills/index.ts:L33-L48, core-libs/schematics/src/migrations/add-or-update-ai-skills/ai-context.utils.ts:L35-L51).

2. Con TTY chiede `Install/Update … (Y/n)` via `readline`; in update riusa gli strumenti già presenti, altrimenti chiede uno per uno (core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L50-L85, core-libs/schematics/src/migrations/add-or-update-ai-skills/ai-context.utils.ts:L53-L86).
3. `scheduleSkillsDeleteAndCopy` aggiunge `@spartacus/skills` con `overwrite: true`, accoda l'install e poi `RunSchematicTask('@spartacus/schematics', 'ai-context', { aiTools, deleteBeforeCopy: true })` (core-libs/schematics/src/migrations/add-or-update-ai-skills/ai-context.utils.ts:L95-L115).

3.12 Modernize-app (schematics manuali)

Tre schematics da lanciare a mano con `ng generate` per gli upgrade che non si possono automatizzare in sicurezza dentro `ng update`:

- `modernize-app-migrated-from-6_8-to-2211_19`: application builder, tsconfig, app module e, solo con legacy server builder, 8 step SSR (core-libs/schematics/src/modernize-app-migrated-from-6_8-to-2211_19/index.ts:L26-L48);
- `modernize-app-migrated-from-2211_32-to-2211_36` (Angular 19): angular.json, tsconfig, assets e favicon in `public`, `main.ts`, i18n lazy loading, server.ts (core-libs/schematics/src/modernize-app-migrated-from-2211_32-to-2211_36/index.ts:L29-L48);
- `modernize-app-to-standalone-bootstrap-application`: `bootstrapModule(AppModule)` → `bootstrapApplication(AppComponent, appConfig)` (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/index.ts:L26-L54), con un `app.config.ts` generato che contiene `provideBrowserGlobalErrorListeners()`, `provideZoneChangeDetection({ eventCoalescing: true })`, `provideHttpClient(...)`, `importProvidersFrom(AppModule)` (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/create-app-config.ts:L16-L59).

Ogni step è avvolto in `withFallbackDocs*`: un'eccezione viene loggata con il link alla guida manuale e viene restituito lo stesso `tree` (le modifiche già applicate dallo step prima dell'eccezione non vengono annullate) (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L12-L43, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L32-L41).

3.13 Utility AST: due mondi

Gli schematics usano **due** API di manipolazione del codice:

- **ts-morph** (installazione, wrapper, rename): `createProgram` crea un `Project` dal tsconfig sopra un `TreeFileSystem` che adatta il `Tree` del devkit (core-libs/schematics/src/shared/utils/program.ts:L11-L35); `saveAndFormat` fa `organizeImports()`, `formatText({indentSize: 2})`, `saveSync()` (core-libs/schematics/src/shared/utils/program.ts:L37-L48); `writeFileSync` evita la scrittura se il contenuto è identico (core-libs/schematics/src/shared/utils/tree-file-system.ts:L84-L94). `addModuleImport/Export/Provider` aggiungono agli array di `@NgModule` con un `order` opzionale (core-libs/schematics/src/shared/utils/new-module-utils.ts:L87-L176) e deduplicano con confronto testuale normalizzato (core-libs/schematics/src/shared/utils/new-module-utils.ts:L222-L284, core-libs/schematics/src/shared/utils/config-utils.ts:L52-L84).
- **TypeScript AST "classico"** + `InsertChange/RemoveChange/ReplaceChange` applicati con `UpdateRecorder` in `commitChanges` (core-libs/schematics/src/shared/utils/file-utils.ts:L292-L332): usato da quasi tutti i mechanism di migrazione.

3.14 Schematics delle feature lib

Esistono cartelle `schematics/` in 20 feature-libs e 12 integration-libs (censimento del dossier). Esempio `@spartacus/checkout`: `"schematics": "./schematics/collection.json"`, build `tsc -p ./tsconfig.schematics.json` (feature-libs/checkout/package.json:L21-L23, feature-libs/checkout/package.json:L51-L51), collection con `ng-add` (privato) e `add` su `addCheckoutFeatures` (feature-libs/checkout/schematics/collection.json:L3-L17). `addCheckoutFeatures` valida che `@spartacus/core` sia installato, calcola la closure, poi `analyzeApplication` → `addFeatures` → `addPackageJsonDependenciesForLibrary(peerDependencies)` → `finalizeInstallation` (feature-libs/checkout/schematics/add-checkout/index.ts:L25-L43, core-libs/schematics/src/shared/utils/workspace-utils.ts:L161-L168). Lo schema offre `Checkout`, `Checkout-B2B`, `Checkout-Scheduled-Replenishment` con default `["Checkout"]` (feature-libs/checkout/schematics/add-checkout/schema.json:L24-L33). Lo stesso schema vale per tutte le 32 collection: ogni `*/schematics/collection.json` di feature-libs e integration-libs dichiara sia `"ng-add"` sia `"add"`, e ogni `schematics/add-*/index.ts` chiama `validateSpartacusInstallation` e `finalizeInstallation` (verifica con `grep` su tutti i 32 file, es. feature-libs/checkout/schematics/collection.json:L3-L17, feature-libs/checkout/schematics/add-checkout/index.ts:L25-L43).

---

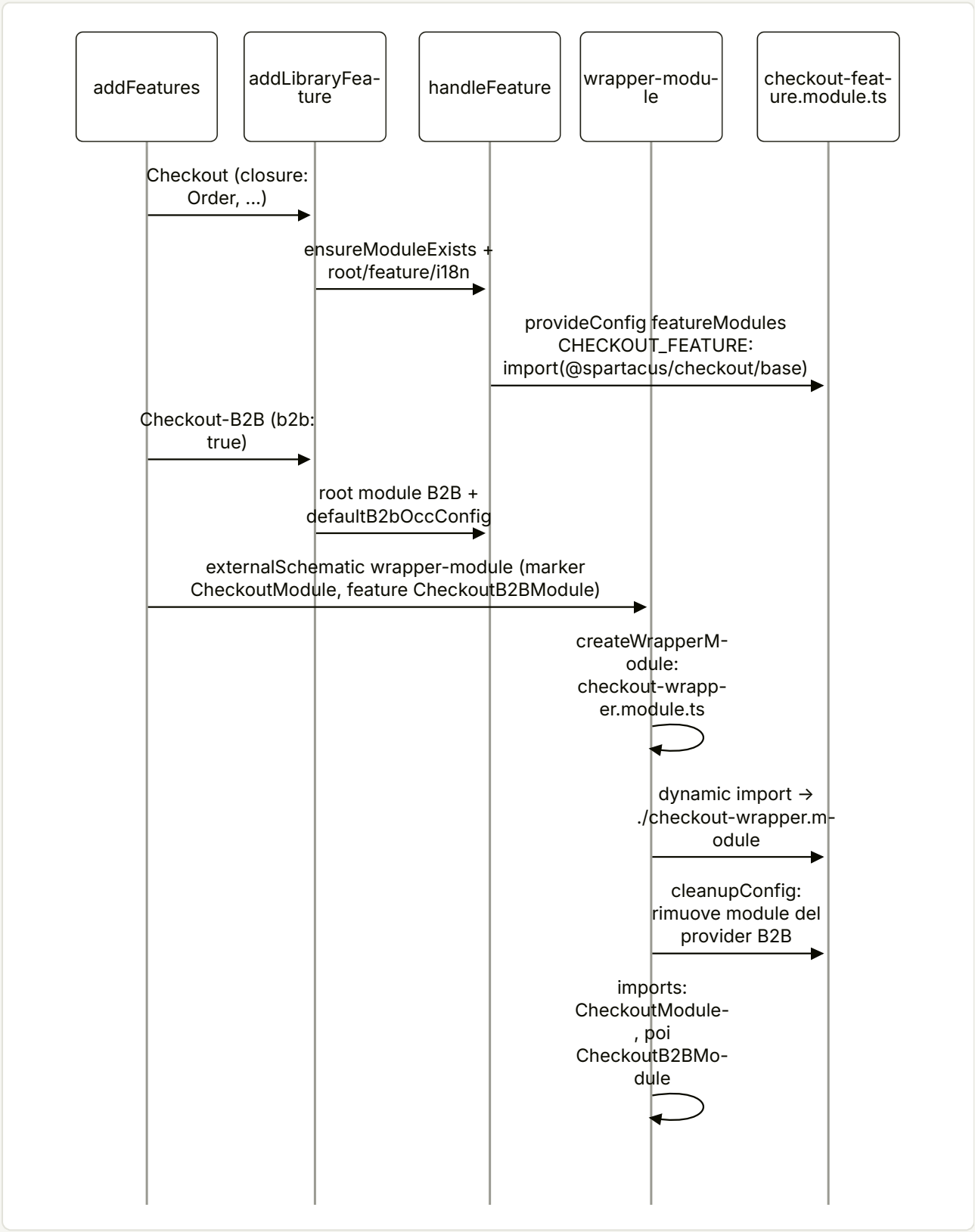
4. Diagrammi

4.1 `ng add @spartacus/schematics` (chain di `add-spartacus`)

■ TAVOLA 43.1 diagramma a tutta pagina, in fondo al capitolo

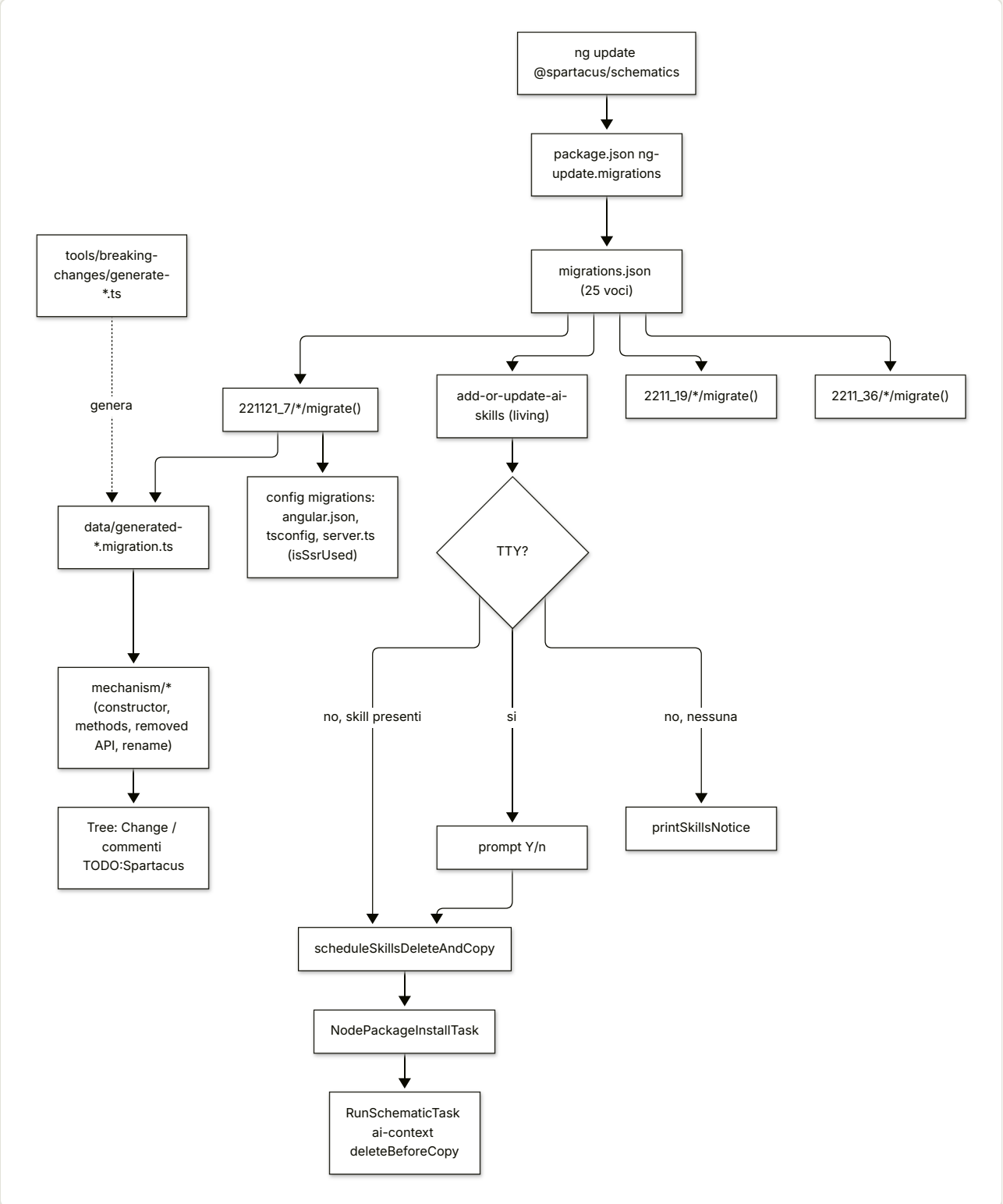
Ogni nodo corrisponde a una rule elencata in §3.3 (core-libs/schematics/src/add-spartacus/index.ts:L531-L590, core-libs/schematics/src/ng-add/index.ts:L17-L24).

4.2 Installazione di una feature con `importAfter` (Checkout-B2B)



Riferimenti: `core-libs/schematics/src/shared/utils/feature-utils.ts:L102-L154`, `core-libs/schematics/src/shared/utils/lib-utils.ts:L293-L326`, `core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L83-L125`, `core-libs/schematics/src/wrapper-module/index.ts:L527-L541`.

4.3 ng update @spartacus/schematics



Riferimenti: core-libs/schematics/package.json:L46-L47, core-libs/schematics/src/migrations/migrations.json:L3-L127, core-libs/schematics/src/migrations/221121_7/constructor-deprecations/constructor-deprecations.ts:L11-L19, core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L32-L86, tools/breaking-changes/common.ts:L12-L19.

4.4 Schematic `ai-context`

■ **TAVOLA 43.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/schematics/src/ai-context/index.ts:L41-L254 (risoluzione della sorgente: core-libs/schematics/src/ai-context/index.ts:L84-L111).

---

5. Estratti di codice reale

5.1 ng-add

core-libs/schematics/src/ng-add/index.ts:L17-L24

```ts
export default function (options: SpartacusOptions): Rule {
  return (host: Tree, context: SchematicContext) => {
    return chain([
      schematic('add-spartacus', options),
      options.ssr ? schematic('add-ssr', options) : noop(),
    ])(host, context);
  };
}
```

5.2 La chain di addSpartacus (abbreviata)

core-libs/schematics/src/add-spartacus/index.ts:L531-L590

```ts
export function addSpartacus(options: SpartacusOptions): Rule {
  return (tree: Tree, context: SchematicContext) => {
    const features = analyzeCrossFeatureDependencies(options.features ?? []);
    const dependencies = prepareDependencies(features);
    const spartacusRxjsDependency: NodeDependency[] = [
      dependencies.find((dep) => dep.name === RXJS) as NodeDependency,
    ];
    const packageJsonFile = readPackageJson(tree);
    return chain([
      detectAndSetupAppStructure(options),

      createAppModule(options),
      updateAppConfig(options),
      addStorefrontComponentToAppComponent(options),

      analyzeApplication(options, features),
      // ...
      addFeatures(options, features),

      addAiContext(options),
```

```
      chain([
        addPackageJsonDependencies(
          prepareDependencies(features),
          packageJsonFile
        ),
        /**
         * Force installing versions of dependencies used by Spartacus.
         * E.g. ng13 uses rxjs 7, but Spartacus uses rxjs 6.
         */
        updatePackageJsonDependencies(spartacusRxjsDependency, packageJsonFile),
        installPackageJsonDependencies(),
        replaceCaretWithTildeForSpartacusDependencies(options),
      ]),

      scheduleAiContext(options),

      finalizeInstallation(options, features),
    ])(tree, context);
  };
}
```

5.3 Feature toggle tutti a `true`, letti dal file copiato

core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L80-L113

```ts
function createFeatureTogglesProvider(): string {
  const defaultFeatureToggles = getDefaultFeatureToggles();

  //for each key in `defaultFeatureToggles` map it to a key with value true
  const featureTogglesAllEnabled: Record<string, boolean> = Object.keys(
    defaultFeatureToggles
  ).reduce((acc, key) => ({ ...acc, [key]: true }), {});
  // ...
  return `provideFeatureToggles(${featureTogglesString})`;
}
// ...
function getDefaultFeatureToggles(): Record<string, boolean> {
  const { defaultFeatureToggles } = require(
    FEATURE_TOGGLES_COPIED_FROM_CORE_LIB_PATH
  );
  return defaultFeatureToggles;
}
```

5.4 addFeatures

core-libs/schematics/src/shared/utils/feature-utils.ts:L115-L152

```ts
/**
 * In an existing Spartacus application, we don't want to
 * force-install the dependent features.
 */
const featuresToInstall = options.internal?.existingSpartacusApplication
  ? (options.features ?? [])
  : features;

const rules: Rule[] = [];
for (const feature of featuresToInstall) {
  const schematicsConfiguration =
    featureSchematicConfigMapping.get(feature);
  // ...
  // TODO:#schematics - fix the interactivity for the CDS / ASM, etc.
  const libraryOptions =
    schematicsConfiguration.customConfig?.(options).options ?? options;

  rules.push(addLibraryFeature(libraryOptions, schematicsConfiguration));

  const wrappers = analyzeWrappers(schematicsConfiguration, libraryOptions);
  for (const { wrapperOptions } of wrappers) {
    rules.push(
      externalSchematic(
        SPARTACUS_SCHEMATICS,
        'wrapper-module',
        wrapperOptions
      )
    );
  }
}

return chain(rules);
```

5.5 Generazione del provider lazy

core-libs/schematics/src/shared/utils/lib-utils.ts:L401-L434

```ts
const configFeatures = ([] as Module[]).concat(config.featureModule);
for (let i = 0; i < configFeatures.length; i++) {
  const featureModule = configFeatures[i];

  // if it's already in a wrapper module
  if (findFeatureModule(featureModule, appSourceFiles)) {
    break;
  }

  let content = `${PROVIDE_CONFIG_FUNCTION}(<${CMS_CONFIG}>{
        featureModules: {`;
```

```
    if (options.lazy) {
      let lazyLoadingChunkName = config.moduleName;
      if (config.lazyLoadingChunk) {
        const namedImportsContent = config.lazyLoadingChunk.namedImports[i];
        lazyLoadingChunkName = `[${namedImportsContent}]`;
        createImports(sourceFile, config.lazyLoadingChunk);
      }
      content =
        content +
        `${lazyLoadingChunkName}: {
              module: () =>
                import('${featureModule.importPath}').then((m) => m.${featureModule.name}),
            },`;

      addModuleProvider(sourceFile, {
        // ...
        content: content + `})`,
      });
```

5.6 `SchematicConfig` di Checkout B2B

core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L83-L125

```ts
export const CHECKOUT_B2B_SCHEMATICS_CONFIG: SchematicConfig = {
  library: {
    featureName: CHECKOUT_B2B_FEATURE_NAME,
    mainScope: SPARTACUS_CHECKOUT,
    featureScope: SPARTACUS_CHECKOUT_B2B,
    b2b: true,
  },
  folderName: CHECKOUT_FOLDER_NAME,
  moduleName: CHECKOUT_BASE_MODULE_NAME,
  featureModule: {
    name: CHECKOUT_B2B_MODULE,
    importPath: SPARTACUS_CHECKOUT_B2B,
  },
  // ...
  dependencyFeatures: [
    ORDER_FEATURE_NAME,
    CHECKOUT_BASE_FEATURE_NAME,
    USER_PROFILE_FEATURE_NAME,
  ],
  importAfter: [
    {
      markerModuleName: CHECKOUT_BASE_MODULE,
      featureModuleName: CHECKOUT_B2B_MODULE,
```

```
    },
  ],
};
```

5.7 Algoritmo di Kahn

core-libs/schematics/src/shared/utils/graph-utils.ts:L108-L134

```ts
  // Create a queue which stores the vertex without dependencies
  const queue = vertices.filter((vertex) => !inDegree[vertex]);
  const topNums: Record<string, number> = {};
  let index = 0;

  while (queue.length) {
    const vertex = queue.shift();
    // ...
    topNums[vertex] = index;
    index += 1;
    // adjust the incoming degree of its neighbors
    for (const neighbor of graph.getAdjacentVertices()[vertex]) {
      inDegree[neighbor]--;
      if (inDegree[neighbor] === 0) {
        queue.push(neighbor);
      }
    }
  }

  if (index !== vertices.length) {
    throw new Error(`Circular dependency detected.`);
  }

  return Object.keys(topNums).reverse();
```

5.8 Il chain di `generateWrapperModule`

core-libs/schematics/src/wrapper-module/index.ts:L527-L541

```ts
export function generateWrapperModule(options: SpartacusWrapperOptions): Rule {
  return (_tree: Tree, _context: SchematicContext): Rule => {
    return chain([
      checkWrapperModuleExists(options),

      createWrapperModule(options),

      updateFeatureModule(options),
      removeLibraryDynamicImport(options),
```

```
      updateWrapperModule(options, options.markerModuleName),
      updateWrapperModule(options, options.featureModuleName),
    ]);
  };
}
```

5.9 addSSR

core-libs/schematics/src/add-ssr/index.ts:L442-L463

```ts
    return chain([
      addPackageJsonDependencies(prepareDependencies(), packageJson),
      externalSchematic(ANGULAR_SSR, 'ng-add', {
        project: options.project,
      }),
      removeOutputModeSupportedOnlyInNewSsrApi(options),
      removeServerRoutesFile(options),
      addBuildSsrScript(options),
      createAppServerModule(options),
      updateAppConfigServer(options),
      modifyAppServerModuleFile(),
      modifyIndexHtmlFile(options),
      updateAppConfigInSsr(options),
      branchAndMerge(
        chain([mergeWith(serverTemplate, MergeStrategy.Overwrite)]),
        MergeStrategy.Overwrite
      ),

      disableSsrAndPrerenderingInAngularJson(options),

      installPackageJsonDependencies(),
    ])(tree, context);
```

5.10 Wrapper per versione + dato generato

core-libs/schematics/src/migrations/221121_7/constructor-deprecations/constructor-deprecations.ts:L7-L19

```ts
import { Rule, SchematicContext, Tree } from '@angular-devkit/schematics';
import { migrateConstructorDeprecation } from '../../mechanism/constructor-deprecations/constructor-deprecations';
import { GENERATED_CONSTRUCTOR_MIGRATIONS } from './data/generated-constructor.migration';

export function migrate(): Rule {
  return (tree: Tree, context: SchematicContext) => {
    return migrateConstructorDeprecation(
      tree,
```

```
      context,
      GENERATED_CONSTRUCTOR_MIGRATIONS
    );
  };
}
```

core-libs/schematics/src/migrations/221121_7/constructor-deprecations/data/generated-constructor.migration.ts:L7-L29

```ts
//
// Generated file, don't update manually.
//
import { ConstructorDeprecation } from '../../../../shared/utils/file-utils';

export const GENERATED_CONSTRUCTOR_MIGRATIONS: ConstructorDeprecation[] = [
  {
    class: 'EntitySuccessAction',
    importPath: '@spartacus/core',
    deprecatedParams: [
      {
        className: 'string',
        importPath: '',
      },
      // ...
      {
        className: 'any | undefined',
        importPath: '',
      },
    ],
```

5.11 Mechanism costruttori

core-libs/schematics/src/migrations/mechanism/constructor-deprecations/constructor-deprecations.ts:L29-L58

```ts
  const project = getSourceRoot(tree, {});
  const sourceFiles = getAllTsSourceFiles(tree, project);
  for (const originalSource of sourceFiles) {
    const sourcePath = originalSource.fileName;

    for (const constructorDeprecation of constructorDeprecations) {
      if (
        !isCandidateForConstructorDeprecation(
          originalSource,
          constructorDeprecation
        )
      ) {
        continue;
```

```
}

for (const newConstructorParam of constructorDeprecation.addParams ||
  []) {
  // 'source' has to be reloaded after each committed change
  const source = getTsSourceFile(tree, sourcePath);
  const nodes = getSourceNodes(source);
  const constructorNode = findConstructor(nodes);

  const changes = addConstructorParam(
    source,
    sourcePath,
    constructorNode,
    newConstructorParam
  );
  commitChanges(tree, sourcePath, changes, InsertDirection.RIGHT);
}
```

5.12 Mechanism rename symbol (ts-morph)

core-libs/schematics/src/migrations/mechanism/rename-symbol/rename-symbol.ts:L40-L80

```ts
        if (renamedSymbol) {
          const newNodeName = renamedSymbol.newNode
            ? renamedSymbol.newNode
            : importName;

          const oldAlias = namedImport.getAliasNode()?.getText();
          let importPath = renamedSymbol.previousImportPath;

          if (renamedSymbol.newImportPath) {
            importPath = renamedSymbol.newImportPath;
          }

          if (!oldAlias && renamedSymbol.newNode) {
            namedImport.renameAlias(newNodeName);
          }
          // ...
          if ((id.getImportClause()?.getNamedImports()?.length || 0) > 1) {
            namedImport.remove();
          } else {
            id.remove();
          }
        }
      });
});

if (importDeclarationStructures.length) {
```

```
        sourceFile.addImportDeclarations(importDeclarationStructures);

        // organizeImports will remove unused imports
        saveAndFormat(sourceFile);
      }
```

5.13 Living migration AI (ramo non interattivo)

core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L32-L48

```ts
export function migrate(): Rule {
  return async (tree: Tree, context: SchematicContext): Promise<void> => {
    if (!isInteractiveTerminal()) {
      const existingTools = detectInstalledTools(tree);
      if (existingTools.length > 0) {
        // CI / non-interactive: deterministically re-sync the skills that are
        // already present. The delete + copy + integrity verify is idempotent,
        // so this keeps automated `ng update` runs from silently drifting.
        scheduleSkillsDeleteAndCopy(tree, context, existingTools);
        context.logger.info('Re-syncing existing Spartacus AI skills…');
      } else {
        // Nothing installed yet — we can't guess a tool without a prompt, so
        // fall back to the manual-install notice.
        printSkillsNotice(context);
      }
      return;
    }
```

5.14 Copia e verifica delle skill

core-libs/schematics/src/ai-context/index.ts:L140-L157

```ts
function copySkill(
  tree: Tree,
  files: SkillFile[],
  target: AiTool,
  options?: CopyOptions
): void {
  const dest = SKILL_DEST[target];
  if (options?.deleteBeforeCopy) {
    deleteSkillDir(tree, dest);
  }
  writeSkillTree(tree, files, dest);
  verifySkillTree(tree, files, dest);
  if (options?.debug) {
    options.logger?.info(
      `✅ Verified ${files.length} skill file(s) at ${dest}.`
```

```
    );
  }
}
```

5.15 Fallback dei modernize

core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L31-L43

```ts
export function withFallbackDocsForStandaloneBootstrap(rule: Rule): Rule {
  return (tree: Tree, context: SchematicContext) => {
    try {
      return rule(tree, context);
    } catch (error) {
      printErrorWithDocsForStandaloneBootstrap(
        error instanceof Error ? error.message : 'Unknown error',
        context
      );
      return tree;
    }
  };
}
```

5.16 Schematics di una feature lib

feature-libs/checkout/schematics/add-checkout/index.ts:L25-L43

```ts
export function addCheckoutFeatures(options: SpartacusCheckoutOptions): Rule {
  return (tree: Tree, _context: SchematicContext): Rule => {
    const packageJson = readPackageJson(tree);
    validateSpartacusInstallation(packageJson);

    const features = analyzeCrossFeatureDependencies(
      options.features as string[]
    );

    return chain([
      analyzeApplication(options, features),

      addFeatures(options, features),
      addPackageJsonDependenciesForLibrary(peerDependencies, options),

      finalizeInstallation(options, features),
    ]);
  };
}
```

6. Configurazione

Opzioni degli schematics (path relativi alla root del repo):

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`features`	string[]	16 feature (ASM … User-Profile)	core-libs/schematics/src/add-spartacus/schema.json:L14-L34	core-libs/schematics/src/add-spartacus/index.ts:L533-L533
`lazy`	boolean	`true`	core-libs/schematics/src/add-spartacus/schema.json:L338-L342	core-libs/schematics/src/shared/utils/lib-utils.ts:L412-L412
`baseUrl`	string	`https://localhost:9002`	core-libs/schematics/src/add-spartacus/schema.json:L343-L347	core-libs/schematics/src/add-spartacus/configuration.ts:L211-L211, core-libs/schematics/src/add-spartacus/index.ts:L238-L238, core-libs/schematics/src/add-ssr/index.ts:L105-L105
`occPrefix`	string	—	core-libs/schematics/src/add-spartacus/schema.json:L348-L351	core-libs/schematics/src/add-spartacus/configuration.ts:L212-L214
`baseSite` / `currency` / `language` / `urlParameters`	CSV string	—	core-libs/schematics/src/add-spartacus/schema.json:L352-L367	core-libs/schematics/src/add-spartacus/configuration.ts:L124-L151
`useMetaTags`	boolean	—	core-libs/schematics/src/add-spartacus/schema.json:L368-L371	core-libs/schematics/src/add-spartacus/index.ts:L564-L564, core-libs/schematics/src/add-spartacus/configuration.ts:L217-L217
`featureLevel` (`@deprecated`)	string	`major.minor` dello schematics	core-libs/schematics/src/add-spartacus/schema.ts:L19-L22	core-libs/schematics/src/add-spartacus/configuration.ts:L236-L242, core-libs/schematics/src/add-spartacus/index.ts:L74-L76
`overwriteAppComponent`	boolean	`true`	core-libs/schematics/src/add-spartacus/schema.json:L376-L380	core-libs/schematics/src/add-spartacus/index.ts:L215-L220
`ssr`	boolean	`false`	core-libs/schematics/src/add-spartacus/schema.json:L381-L385	core-libs/schematics/src/ng-add/index.ts:L21-L21
`theme`	string	—	core-libs/schematics/src/add-spartacus/schema.json:L386-L389	core-libs/schematics/src/add-spartacus/index.ts:L170-L172

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
aiTools	('agents'|'claude')[]	["agents"]	core-libs/schematics/src/add-spartacus/schema.json:L390-L414	core-libs/schematics/src/add-spartacus/ai-context.ts:L29-L29, core-libs/schematics/src/ai-context/index.ts:L43-L43
debug	boolean	false	core-libs/schematics/src/add-spartacus/schema.json:L333-L337	ovunque (options.debug), es. core-libs/schematics/src/shared/utils/logger-utils.ts:L12-L18
deleteBeforeCopy	boolean (interno)	—	core-libs/schematics/src/ai-context/schema.ts:L12-L13	core-libs/schematics/src/ai-context/index.ts:L68-L68, core-libs/schematics/src/migrations/add-or-update-ai-skills/ai-context.utils.ts:L111-L111
internal.existingSpartacusApplication	boolean (interno)	—	core-libs/schematics/src/shared/utils/lib-utils.ts:L97-L102	core-libs/schematics/src/shared/utils/feature-utils.ts:L119-L121, core-libs/schematics/src/shared/utils/lib-utils.ts:L902-L902
internal.wrapperModulePath	string (interno)	—	core-libs/schematics/src/wrapper-module/schema.ts:L32-L37	core-libs/schematics/src/wrapper-module/index.ts:L139-L141, core-libs/schematics/src/wrapper-module/index.ts:L282-L282, core-libs/schematics/src/wrapper-module/index.ts:L478-L478
cmsComponentDataModelPath	string	@spartacus/core (JSON)	core-libs/schematics/src/add-cms-component/schema.json:L23-L28	core-libs/schematics/src/add-cms-component/index.ts:L217-L217
env SPARTACUS_SKILLS_DIR	path	—	—	core-libs/schematics/src/ai-context/index.ts:L85-L88
env SERVER_REQUEST_ORIGIN	string	—	—	generato in core-libs/schematics/src/add-ssr/index.ts:L86-L88
env SSR_ALLOWED_ORIGINS, PORT	string	PORT = 4000	—	core-libs/schematics/src/add-ssr/files/server.__typescriptExt__:L30-L32, core-libs/schematics/src/add-ssr/files/server.__typescriptExt__:L66-L66
ng-add.save	string	devDependencies	core-libs/schematics/package.json:L43-L45	Angular CLI: manifest['ng-add']?.save, dev dependency se 'devDependencies' (deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L426-L426, deps:devkit/angular-cli-21.2.23/package/src/commands/add/cli.js:L473-L473)
ng-update.migrations	path	./src/migrations/migrations.json	core-libs/schematics/package.json:L46-L47	Angular CLI: packageNode['ng-update'].migrations (deps:devkit/angular-cli-21.2.23/package/src/commands/update/cli.js:L252-L253)

7. Commenti e note notevoli dal sorgente

- "it exports utils for other Spartacus libs' schematics, but is not intended as a public API for external use. It may change without deprecation." — core-libs/schematics/index.ts:L9-L10.
- `@deprecated - features level is no longer used` — core-libs/schematics/src/add-spartacus/schema.ts:L19-L22, core-libs/schematics/src/add-spartacus/configuration.ts:L233-L235.
- Motivazione ^→~: "since 2211.19 we agreed to break SemVer and release breaking changes in 'minor' (x.Y.z) versions" — core-libs/schematics/src/add-spartacus/index.ts:L482-L489.
- "Force installing versions of dependencies used by Spartacus. E.g. ng13 uses rxjs 7, but Spartacus uses rxjs 6." — commento datato, core-libs/schematics/src/add-spartacus/index.ts:L578-L581.
- "Since v17 Angular `ng new` command by default creates a local `AppRoutingModule` file in the project. So we have to replace it." — core-libs/schematics/src/add-spartacus/update-app-module.ts:L75-L78.
- `TODO: CXSPA-6466 Can be removed if Model T adjust their build process to not require this script.` — core-libs/schematics/src/add-ssr/index.ts:L158-L158, core-libs/schematics/src/migrations/221121_7/ssr/build-ssr-script/index.ts:L22-L22.
- "This could be removed when we migrate to new SSR API." — core-libs/schematics/src/add-ssr/index.ts:L365-L365.
- "Http Transfer Cache is temporarily disabled; https://jira.tools.sap/browse/CXSPA-10430" — core-libs/schematics/src/add-ssr/update-app-config-in-ssr.ts:L48-L48.
- `CXSPA-447: Eventually we should remove all the @import directives … and drop the usage of Bootstrap 4` — core-libs/schematics/src/shared/utils/workspace-utils.ts:L223-L224.
- `TODO:#schematics - fix the interactivity for the CDS / ASM, etc.` — core-libs/schematics/src/shared/utils/feature-utils.ts:L134-L134.
- Workaround ts-morph per file nuovi non referenziati — core-libs/schematics/src/wrapper-module/index.ts:L196-L203.
- "Angular CLI auto-kebab-cases camelCase schema props (aiTools → ai-tools) and refuses comma-separated values for array flags — repeat the flag once per value." — core-libs/schematics/src/ai-context/index.ts:L37-L39.
- "Living migration that runs on every `ng update @spartacus/schematics`…" — core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L22-L31.
- `// NOSONAR typescript:S4158` su un loop sopra un array vuoto — core-libs/schematics/src/migrations/2211_19/missing-packages/missing-packages.ts:L17-L17.
- `as any` per non introdurre la dipendenza da `@angular-devkit/build-angular` — core-libs/schematics/src/migrations/221121_7/angular-config/index.ts:L34-L35.
- "*Must* be updated when adding a new schematics library or a feature." — core-libs/schematics/src/shared/schematics-config-mappings.ts:L99-L103.
- "Generated file, don't update manually." — es. core-libs/schematics/src/migrations/221121_7/constructor-deprecations/data/generated-constructor.migration.ts:L7-L9.
- `NEW_VERSION = '221121_7'; // Increment this number to match the current release for which the script is used.` — tools/breaking-changes/common.ts:L12-L12.
- La JSDoc di `isCandidateForConstructorDeprecation` cita parametri (`inheritedClass`, `parameterClassTypes`) non più presenti nella firma — core-libs/schematics/src/shared/utils/file-utils.ts:L352-L368.
- La `description` di `01-migration-v2211_36-ssr` dice `index.server.ts`, ma il codice scrive `index.server.html` — core-libs/schematics/src/migrations/migrations.json:L58-L62, core-libs/schematics/src/migrations/2211_36/ssr/update-ssr/update-server-files.ts:L15-L149.

8. Extension point

Tutto ciò che segue è esportato dal barrel di `@spartacus/schematics` (core-libs/schematics/index.ts:L12-L13 → core-libs/schematics/src/shared/index.ts:L7-L10), che però si dichiara **non stabile**. Usarlo dentro schematics propri è legittimo, ma va trattato come API interna in un upgrade (vedi 54 — API pubblica vs interna e upgrade).

EXTENSION POINT	USO	RIFERIMENTO
Schematic `add-spartacus` / `ng-add` con opzioni (`features`, `lazy`, `baseUrl`, `ssr`, `theme`, `aiTools`…)	installazione parametrica	core-libs/schematics/src/add-spartacus/schema.ts:L9-L27
`ng add @spartacus/<lib> --features=…`	installare feature in un'app esistente	feature-libs/checkout/schematics/add-checkout/index.ts:L25-L43
`SchematicConfig` + `addLibraryFeature` / `addFeatures` / `analyzeApplication` / `finalizeInstallation` / `addPackageJsonDependenciesForLibrary`	scrivere lo schematic `ng add` di una **propria** feature lib nello stesso formato di Spartacus	core-libs/schematics/src/shared/utils/lib-utils.ts:L115-L273, core-libs/schematics/src/shared/utils/feature-utils.ts:L102-L154, core-libs/schematics/src/shared/utils/feature-utils.ts:L279-L370, core-libs/schematics/src/shared/utils/lib-utils.ts:L897-L909, core-libs/schematics/src/shared/utils/lib-utils.ts:L763-L777
`customConfig` (`AdditionalFeatureConfiguration` / `AdditionalProviders`)	provider extra nel feature module, override delle opzioni	core-libs/schematics/src/shared/utils/feature-utils.ts:L60-L80
`importAfter` + schematic `wrapper-module`	caricare una propria estensione nello stesso chunk di una feature Spartacus, dopo di essa	core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L119-L124, core-libs/schematics/src/wrapper-module/index.ts:L527-L541
Utility ts-morph: `addModuleImport`, `addModuleExport`, `addModuleProvider`, `ensureModuleExists`, `createImports`, `createProgram`, `saveAndFormat`	manipolare `@NgModule` e import in schematics propri	core-libs/schematics/src/shared/utils/new-module-utils.ts:L40-L176, core-libs/schematics/src/shared/utils/import-utils.ts:L134-L177, core-libs/schematics/src/shared/utils/program.ts:L11-L48
Tipi di deprecazione (`ConstructorDeprecation`, `MethodPropertyDeprecation`, `DeprecatedNode`, `ConfigDeprecation`, `RenamedSymbol`, `ComponentData`)	descrivere migrazioni di una propria libreria nello stesso formato	core-libs/schematics/src/shared/utils/file-utils.ts:L41-L117
Schematic `add-cms-component`	scaffolding di un componente CMS	core-libs/schematics/src/add-cms-component/index.ts:L398-L447
Schematic `ai-context` + env `SPARTACUS_SKILLS_DIR`	copiare una skill da una sorgente alternativa	core-libs/schematics/src/ai-context/index.ts:L84-L111
`generateDefaultWorkspace`	harness di test per gli spec degli schematics di feature lib	core-libs/schematics/src/shared/utils/generate-default-workspace.ts:L45-L80

Ricette pertinenti della Parte III:

- creare una feature lib lazy-loaded che segua lo stesso formato (`featureModules`, root module, i18n, stili) prodotto da `addLibraryFeature`: 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni;

- registrare componenti CMS (alternativa moderna a `ConfigModule.withConfig` generato da `add-cms-component`): 51 — Ricette: componenti CMS, outlet, route e layout;
- `provideConfig` e ordine dei provider nei moduli generati: 50 — Ricette: configurazione, DI, interceptor, eventi;
- cosa si rompe negli upgrade, ruolo di `ng update` e dei commenti `TODO:Spartacus`: 54 — API pubblica vs interna, upgrade, anti-pattern, checklist.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `ng add` **cancella il contenuto di** `app.component.html` (default `overwriteAppComponent: true`) e **sostituisce l'array** `imports` di `AppComponent` (`removeOldImports: true`), perdendo ad esempio `RouterOutlet` (core-libs/schematics/src/add-spartacus/index.ts:L215-L217, core-libs/schematics/src/add-spartacus/add-storefront-component-to-app-component.ts:L46-L48).
2. **Tutti i feature toggle a** `true`**.** Le app nuove non partono dai default di core ma con ogni toggle attivo (core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L77-L93). Vedi 42.
3. **Solo app standalone con naming 2016.** Se tra i sorgenti del primo tsconfig di build non c'è un file il cui path contiene `app.config.ts`, oppure uno che contiene `app.component.ts`, l'installazione si interrompe con il consiglio di rigenerare l'app (core-libs/schematics/src/add-spartacus/index.ts:L324-L369).
4. **Primo progetto di** `angular.json`**.** `getDefaultProjectNameFromWorkspace` restituisce `Object.keys(workspace.projects)[0]`; molte migrazioni chiamano `getSourceRoot(tree, {})` e quindi lavorano solo su quel progetto in un workspace multi-progetto (core-libs/schematics/src/shared/utils/workspace-utils.ts:L111-L115, core-libs/schematics/src/shared/utils/workspace-utils.ts:L33-L49). `getSourceRoot` inoltre **muta** `options.project` (core-libs/schematics/src/shared/utils/workspace-utils.ts:L39-L41).
5. **Migrazioni "a commento" non idempotenti.** Config, metodi/proprietà e API rimosse inseriscono il commento senza controllare se c'è già: rilanciarle duplica i `// TODO:Spartacus` (core-libs/schematics/src/migrations/mechanism/config-deprecations/config-deprecation.ts:L24-L35, core-libs/schematics/src/shared/utils/file-utils.ts:L1145-L1171); lo stesso vale per `insertCommentAboveIdentifier` (metodi/proprietà) e `insertCommentAboveImportIdentifier` (API rimosse) (core-libs/schematics/src/shared/utils/file-utils.ts:L1173-L1203, core-libs/schematics/src/migrations/mechanism/methods-and-properties-deprecations/methods-and-properties-deprecations.ts:L33-L42, core-libs/schematics/src/shared/utils/file-utils.ts:L1235-L1242, core-libs/schematics/src/migrations/mechanism/removed-public-api-deprecations/removed-public-api-deprecation.ts:L34-L44).
6. **Doppio prefisso nei commenti delle API rimosse.** Il generatore mette già `// TODO:Spartacus -` nel `comment` dei dati, e il mechanism lo avvolge ancora con `// TODO:Spartacus :` con `TODO_SPARTACUS = 'TODO:Spartacus -'` il risultato atteso è `// TODO:Spartacus - // TODO:Spartacus - Variable …` (tools/breaking-changes/generate-deleted.ts:L58-L58, core-libs/schematics/src/migrations/221121_7/removed-public-api-deprecations/data/generated-removed-public-api.migration.ts:L15-L16, core-libs/schematics/src/migrations/mechanism/removed-public-api-deprecations/removed-public-api-deprecation.ts:L39-L53, core-libs/schematics/src/shared/constants.ts:L1145-L1145). DA VERIFICARE: inferito dalla lettura del codice, nessuno spec esegue il mechanism sui dati generati.
7. **Match di import per sottostringa.** `insertCommentAboveImportIdentifier` filtra con `.includes(importPath)`: `@spartacus/cart` corrisponde anche a `@spartacus/cart/base` (core-libs/schematics/src/shared/utils/file-utils.ts:L1205-L1217).
8. **Due install task.** Con `aiTools` valorizzato `add-spartacus` accoda un `NodePackageInstallTask` in `installPackageJsonDependencies()` e un altro in `scheduleAiContext` (core-libs/schematics/src/add-

spartacus/index.ts:L583-L587, core-libs/schematics/src/add-spartacus/ai-context.ts:L61-L61). Il devkit non li deduplica: `TaskScheduler.schedule` accoda ogni task con un id nuovo ed `executePostTasks` li esegue tutti in sequenza, quindi l'install gira due volte (deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L117-L130, deps:devkit/angular-devkit-schematics-21.2.23/package/src/engine/engine.js:L283-L297).

9. `ai-context` **manuale dopo un** `npm update`**.** Senza `deleteBeforeCopy` (opzione interna, non nello schema JSON) la destinazione non viene svuotata, ma `verifySkillTree` rifiuta comunque file "extra": se la nuova versione di `@spartacus/skills` ha rimosso un file, `ng g @spartacus/schematics:ai-context` fallisce con "unexpected file(s)", proprio nel caso d'uso suggerito dal log di `ng add` ("Re-run 'ng generate @spartacus/schematics:ai-context' anytime to refresh them after updating the package") (core-libs/schematics/src/ai-context/index.ts:L146-L151, core-libs/schematics/src/ai-context/index.ts:L241-L253, core-libs/schematics/src/add-spartacus/ai-context.ts:L71-L74). Il "copy step" del README è invece il `cp -r` manuale, che allo stesso modo non rimuove i file obsoleti (core-libs/skills/README.md:L50-L55, core-libs/skills/README.md:L58-L61). La via robusta è `ng update`, che passa `deleteBeforeCopy: true` (core-libs/schematics/src/migrations/add-or-update-ai-skills/ai-context.utils.ts:L108-L114).
10. **JSDoc contraddittorie sull'AI context.** L'esempio mostra `--ai-tools=cursor`, ma lo schema accetta solo `agents`/`claude`: da CLI il comando viene rifiutato dalla validazione dello schema (`InvalidInputOptions`) prima di arrivare alla factory; `normalize` scarta valori non supportati solo nelle invocazioni programmatiche (`RunSchematicTask`, `add-spartacus`) (core-libs/schematics/src/ai-context/index.ts:L35-L35, core-libs/schematics/src/ai-context/schema.json:L7-L13, deps:devkit/angular-devkit-schematics-21.2.23/package/tools/schema-option-transform.js:L27-L31, core-libs/schematics/src/ai-context/index.ts:L256-L267); la living migration dice che in CI "falls back to a printed notice", ma se trova skill installate le ri-sincronizza (core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L28-L31, core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L34-L47).
11. `add-ssr` **sovrascrive** `server.ts` con `MergeStrategy.Overwrite`, compreso quello appena generato da `@angular/ssr` (core-libs/schematics/src/add-ssr/index.ts:L455-L458); il template riceve `browserDistDirectory` ma non lo usa (core-libs/schematics/src/add-ssr/index.ts:L119-L129, core-libs/schematics/src/add-ssr/files/server.__typescriptExt__:L19-L20).
12. `"undefined,noSsr"`**.** `useNoSsrConfigurationInNgServe` concatena `${buildTarget},noSsr` anche se `buildTarget` è indefinito (core-libs/schematics/src/add-ssr/index.ts:L336-L343).
13. `modifyIndexHtmlFile` **legge** `src/index.html` **fisso** ma scrive su `getIndexHtmlPath(tree)`: con un `sourceRoot` diverso da `src` il controllo fallisce in silenzio (core-libs/schematics/src/add-ssr/index.ts:L98-L117).
14. **Nome invertito.** `move-zone-change-detection-to-main` sposta `provideZoneChangeDetection` *da* `main.ts` *ad* `app.module.ts` (core-libs/schematics/src/migrations/migrations.json:L93-L97, core-libs/schematics/src/migrations/221121_7/move-zone-change-detection-to-main/index.ts:L11-L16).
15. **Check di duplicato testuali.** `angular-providers` e la migrazione zone controllano con `getFullText().includes(...)`: basta un commento con quel nome per saltare l'aggiunta (core-libs/schematics/src/migrations/221121_7/angular-providers/index.ts:L75-L78, core-libs/schematics/src/migrations/221121_7/move-zone-change-detection-to-main/add-zone-change-detection-to-app-module/index.ts:L86-L90).
16. **Side effect fuori dal Tree.** `2211_36 bootstrap` esegue davvero `npm/yarn/pnpm remove bootstrap` con `spawn(…, { shell: true })` dentro la rule e risolve la promise anche in caso di errore (core-libs/schematics/src/migrations/2211_36/bootstrap/bootstrap.ts:L26-L72). Anche `ai-context` legge le skill dal file system reale e non dal `Tree` (core-libs/schematics/src/ai-context/index.ts:L119-L132): il risultato dipende da ciò che è davvero installato in `node_modules`. Con `--dry-run` il workflow non aggiunge l'`HostSink` e salta i post-task, quindi install e `ai-context` schedulati come task non partono (deps:devkit/

angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L97-L98, deps:devkit/angular-devkit-schematics-21.2.23/package/src/workflow/base.js:L126-L128); invece la rule di uninstallBootstrap restituisce una funzione async che callRuleAsync invoca e attende durante la trasformazione del tree, quindi lo spawn di remove bootstrap gira **anche** con --dry-run (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/call.js:L76-L81).

17. **Path hard-coded.** src/styles.scss, src/main.ts, src/app/app.config.ts in bootstrap e modernize (core-libs/schematics/src/migrations/2211_36/bootstrap/bootstrap.ts:L76-L76, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/update-main-ts.ts:L23-L23, core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/csr-and-ssr/create-app-config.ts:L18-L18).
18. withFallbackDocs* **cattura solo eccezioni sincrone** della rule avvolta (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L31-L43). chain restituisce una funzione async che costruisce osservabili lazy (callRule usa defer) e ritorna () => intermediateTree: se la rule avvolta è una chain, rule(tree, context) restituisce una Promise e gli errori delle rule figlie emergono dopo, fuori dal try sincrono; anche una Promise rifiutata non viene catturata (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/base.js:L46-L61, deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/call.js:L68-L75).
19. **Factory non invocate in** chain**.** updateServerFiles (2211_19) passa removeServer, modifyServerImports, … senza chiamarle (core-libs/schematics/src/migrations/2211_19/update-ssr/update-ssr-files.ts:L25-L32). Funziona comunque: callRuleAsync continua a invocare il risultato finché è una funzione, quindi removeServer(tree, ctx) restituisce la Rule interna, che viene poi eseguita (deps:devkit/angular-devkit-schematics-21.2.23/package/src/rules/call.js:L77-L81). Il codice è fuorviante ma non è un bug.
20. **Bug sottili nell'ordinamento.** groupFeatures usa if (!lastExistingIndex), quindi un indice 0 è trattato come "non trovato" (core-libs/schematics/src/shared/utils/graph-utils.ts:L58-L66); calculateSortInternal con entrambe le voci assenti produce Infinity - Infinity = NaN (core-libs/schematics/src/shared/utils/lib-utils.ts:L890-L890).
21. break **invece di** continue**.** In addToFeatureModule, se un feature module è già presente (es. in un wrapper) il ciclo esce, saltando anche i moduli successivi dell'array (core-libs/schematics/src/shared/utils/lib-utils.ts:L405-L408).
22. isStaticallyImported confronta l'ultimo segmento dopo .: per X.forRoot() diventa forRoot() e non corrisponde al nome del modulo (core-libs/schematics/src/shared/utils/feature-utils.ts:L424-L433).
23. getMajorVersionNumber usa solo il primo carattere: per 221121.22.0 restituisce 2 (core-libs/schematics/src/shared/utils/package-utils.ts:L134-L137).
24. **Dati di deprecazione solo fino a 221121_7.** NEW_VERSION dei generatori è 221121_7; per la 221121.22.0 l'unica migrazione è quella delle skill AI (tools/breaking-changes/common.ts:L12-L12, core-libs/schematics/src/migrations/migrations.json:L123-L127).
25. **Barrel pubblico pesante e incompleto.** generateDefaultWorkspace e test-utils sono esportati, quindi importare @spartacus/schematics carica @angular-devkit/schematics/testing (core-libs/schematics/src/shared/utils/index.ts:L21-L23, core-libs/schematics/src/shared/utils/generate-default-workspace.ts:L7-L10); per contro lib-configs/index.ts non riesporta alcune config (customer-ticketing, estimated-delivery-date, pdf-invoices, pickup-in-store, product-multi-dimensional, requested-delivery-date) e schematics-config-mappings non è esportato (core-libs/schematics/src/shared/lib-configs/index.ts:L7-L21, core-libs/schematics/src/shared/index.ts:L7-L10).
26. add-cms-component **genera API vecchia** (ConfigModule.withConfig) e ha un default cmsComponentDataModelPath discordante fra JSDoc (@spartacus/storefront) e JSON (@spartacus/core) (core-libs/schematics/src/add-cms-component/index.ts:L118-L124, core-libs/schematics/src/add-cms-component/schema.ts:L29-L33).

27. **Versioni e licenze diverse.** `@angular/ssr` è `~21.2.23` nel `package.json` root (quindi in `dependencies.json`) ma `^21.2.23` tra le deps dello schematics (`package.json:L177-L177`, `core-libs/schematics/package.json:L21-L21`); `@spartacus/schematics` è MIT, `@spartacus/skills` e la root Apache-2.0 (`core-libs/schematics/package.json:L11-L11`, `core-libs/skills/package.json:L14-L14`, `package.json:L4-L4`).
28. `prepareDependencies` **calcolato due volte** in `addSpartacus` (`core-libs/schematics/src/add-spartacus/index.ts:L534-L534`, `core-libs/schematics/src/add-spartacus/index.ts:L574-L577`).

---

10. Come fare debug

- `--debug`: quasi ogni rule logga passi e risultati se `options.debug` è `true`, tipicamente con `debugLogRule` (`core-libs/schematics/src/shared/utils/logger-utils.ts:L12-L18`); `addFeatures` stampa anche il "Cross feature graph" calcolato (`core-libs/schematics/src/shared/utils/feature-utils.ts:L107-L113`). Utile per capire perché una feature viene installata (closure delle `dependencyFeatures`).
- **Ordine/dipendenze sbagliati**: controllare `dependencyFeatures` e `importAfter` del `SchematicConfig` e che la config sia in `SCHEMATICS_CONFIGS`; in caso contrario `addFeatures` lancia "[Internal] No feature config found for ..." (`core-libs/schematics/src/shared/utils/feature-utils.ts:L127-L132`). Cicli producono `Circular dependency detected.` (`core-libs/schematics/src/shared/utils/graph-utils.ts:L130-L132`).
- **Feature "dipendente" in un'app esistente**: se manca il marker module `analyzeApplication` lancia con il comando da eseguire, `ng add @spartacus/schematics --features=<marker>` (`core-libs/schematics/src/shared/utils/feature-utils.ts:L279-L370`).
- **Versioni**: le versioni installate vengono da `core-libs/schematics/src/dependencies.json`; dopo una modifica ai `package.json` delle librerie rigenerarlo con `npm run generate:deps` (`package.json:L64-L64`). I log `Upgrading/Downgrading` vengono da `updatePackageJsonDependencies` (`core-libs/schematics/src/shared/utils/package-utils.ts:L296-L335`).
- **Migrazioni che "non fanno nulla"**: molte sono condizionate da `isSsrUsed` / `isUsingLegacyServerBuilder` e loggano lo skip, es. "SSR is not configured - skipping Express 5 migration" (`core-libs/schematics/src/migrations/221121_7/ssr/express-v5/index.ts:L247-L261`, `core-libs/schematics/src/shared/utils/package-utils.ts:L155-L228`). Verificare il builder (`@angular/build:application` o `@angular-devkit/build-angular:application`), `build.options.server` e `ssr.entry`.
- **Mechanism di deprecazione**: un costruttore non migrato di solito non è "candidato" (import della classe da un path diverso, tipi dei parametri non importati, `super()` diverso): rileggere i criteri in `core-libs/schematics/src/shared/utils/file-utils.ts:L352-L412`. Le migrazioni lavorano sul sourceRoot del primo progetto (§9.4).
- **AI skills**: se la copia non avviene cercare il warning "Could not find '@spartacus/skills' in node_modules" (`core-libs/schematics/src/ai-context/index.ts:L51-L59`); per provare una skill locale impostare `SPARTACUS_SKILLS_DIR` (`core-libs/schematics/src/ai-context/index.ts:L85-L88`). Gli errori "Skill integrity check failed" indicano file mancanti/diversi/extra (`core-libs/schematics/src/ai-context/index.ts:L207-L254`).
- **Test**: gli spec sono Jest (`*_spec.ts`, soglie 90/74/90/90) (`core-libs/schematics/jest.schematics.config.js:L8-L13`, `core-libs/schematics/jest.schematics.config.js:L28-L35`). Comandi: `cd core-libs/schematics && npm run test`, `nx run <library-name>:test-jest --testPathPatterns="<spec>"`, `npm run test:all-schematics` (che esegue anche `-u` sugli snapshot) (`.claude/skills/schematics/SKILL.md:L20-L35`, `package.json:L146-L146`); `test-jest-update` aggiorna gli snapshot (`core-libs/schematics/project.json:L20-L26`). Gli spec delle feature lib partono da `generateDefaultWorkspace`, che tiene in cache il tree tra i test (`core-libs/schematics/src/shared/utils/generate-default-workspace.ts:L43-L51`); quelli dei

mechanism usano la collection fittizia `migrations/test/migrations-test.json` (core-libs/schematics/src/migrations/test/migrations-test.json:L1-L49).

- **Modernize**: se uno step fallisce si vedono "NOTA <messaggio>" e il link alla guida manuale, e viene restituito lo stesso `tree`, senza annullare le modifiche parziali già fatte dallo step (core-libs/schematics/src/modernize-app-to-standalone-bootstrap-application/fallback-advice-to-follow-docs.ts:L20-L43).

---

11. Domande di autoverifica

1. **Perché `@spartacus/schematics` può generare `provideFeatureToggles` con tutte le chiavi senza dipendere da `@spartacus/core` a runtime?** Perché uno step di pre-build copia `feature-toggles.ts` in `src/feature-toggles.copied-from-core-lib.ts`, che viene compilato con lo schematics e letto con `require` (core-libs/schematics/copy-feature-toggles.js:L5-L10, core-libs/schematics/src/add-spartacus/spartacus-feature-toggles.ts:L95-L113).
2. **Quali schematics esegue `ng add @spartacus/schematics --ssr`?** `add-spartacus` e poi `add-ssr`, con le stesse opzioni (core-libs/schematics/src/ng-add/index.ts:L17-L24).
3. **Dove finisce un feature module installato in modalità lazy e in che forma?** In `app/spartacus/features/<folder>/<module>-feature.module.ts`, come `provideConfig(<CmsConfig>{ featureModules: { [CHUNK]: { module: () => import(...).then(...) } } })` (core-libs/schematics/src/shared/utils/lib-utils.ts:L293-L326, core-libs/schematics/src/shared/utils/lib-utils.ts:L397-L446).
4. **A cosa serve `importAfter` e cosa produce?** A importare una feature (es. `CheckoutB2BModule`) dopo un'altra (`CheckoutModule`) nello stesso chunk: `addFeatures` lancia `wrapper-module`, che crea `<name>-wrapper.module.ts`, ripunta il dynamic import al wrapper e vi importa staticamente i due moduli in ordine (core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L119-L124, core-libs/schematics/src/shared/utils/feature-utils.ts:L140-L149, core-libs/schematics/src/wrapper-module/index.ts:L527-L541).
5. **Perché le dipendenze `@spartacus/*` vengono scritte con `~` invece di `^`?** Perché dalla 2211.19 i breaking change possono arrivare nelle versioni "minor"; `replaceCaretWithTildeForSpartacusDependencies` li blocca alla minor (core-libs/schematics/src/add-spartacus/index.ts:L482-L529).
6. **Come è fatta una migrazione di deprecazione costruttori per una nuova release?** `tools/breaking-changes/generate-constructors.ts` scrive `data/generated-constructor.migration.ts` sotto `migrations/<NEW_VERSION>`; un wrapper `migrate()` passa i dati a `migrateConstructorDeprecation`; `migrations.json` registra la voce con la `version` (tools/breaking-changes/common.ts:L12-L19, core-libs/schematics/src/migrations/221121_7/constructor-deprecations/constructor-deprecations.ts:L7-L19, core-libs/schematics/src/migrations/migrations.json:L103-L107).
7. **Perché la migrazione `add-or-update-ai-skills` gira a ogni `ng update`?** Perché è una "living migration": la sua `version` in `migrations.json` viene aggiornata a ogni release (core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L22-L31). La CLI esegue le voci con `version` nel range `>from <=to`, quindi la migrazione gira in ogni `ng update` che parte da una versione inferiore a quella del pacchetto, non se `from` è già pari (deps:devkit/angular-cli-21.2.23/package/src/commands/update/utilities/migration.js:L107-L118).
8. **Cosa succede in CI (senza TTY) durante `ng update` se il progetto ha già `.claude/skills/spartacus-developer/SKILL.md`?** Le skill vengono ri-sincronizzate: `@spartacus/skills` con `overwrite: true`, install, poi `ai-context` con `deleteBeforeCopy: true` (core-libs/schematics/src/migrations/add-or-update-ai-skills/index.ts:L33-L48, core-libs/schematics/src/migrations/add-or-update-ai-skills/ai-context.utils.ts:L95-L115).

9. **Una migrazione Express 5 non ha modificato** `server.ts`**. Prima ipotesi?** L'app non è riconosciuta come SSR da `isSsrUsed` (builder, `build.options.server`, `ssr.entry`, file server non vuoto) e la migrazione esce con il log di skip (`core-libs/schematics/src/migrations/221121_7/ssr/express-v5/index.ts:L247-L261`, `core-libs/schematics/src/shared/utils/package-utils.ts:L193-L228`).
10. **Perché in un workspace con più applicazioni le migrazioni potrebbero non toccare la tua app?** Perché molte usano `getSourceRoot(tree, {})` / `getDefaultProjectNameFromWorkspace`, che sceglie il primo progetto di `angular.json` (`core-libs/schematics/src/shared/utils/workspace-utils.ts:L33-L49`, `core-libs/schematics/src/shared/utils/workspace-utils.ts:L111-L115`).

TAVOLA 43.1 **4.1 ng add @spartacus/schematics (chain di add-spartacus)**

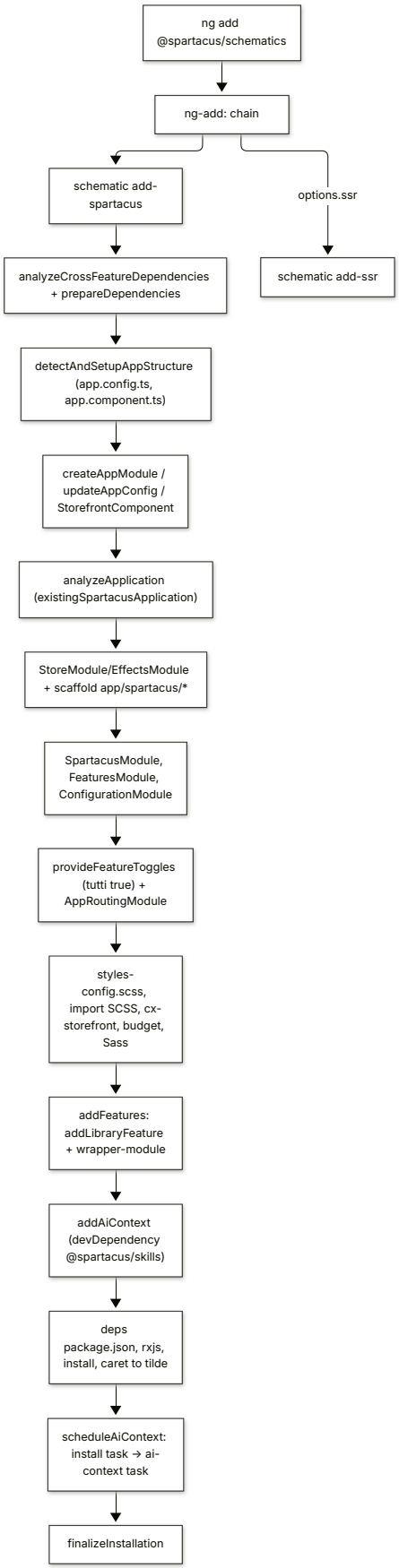
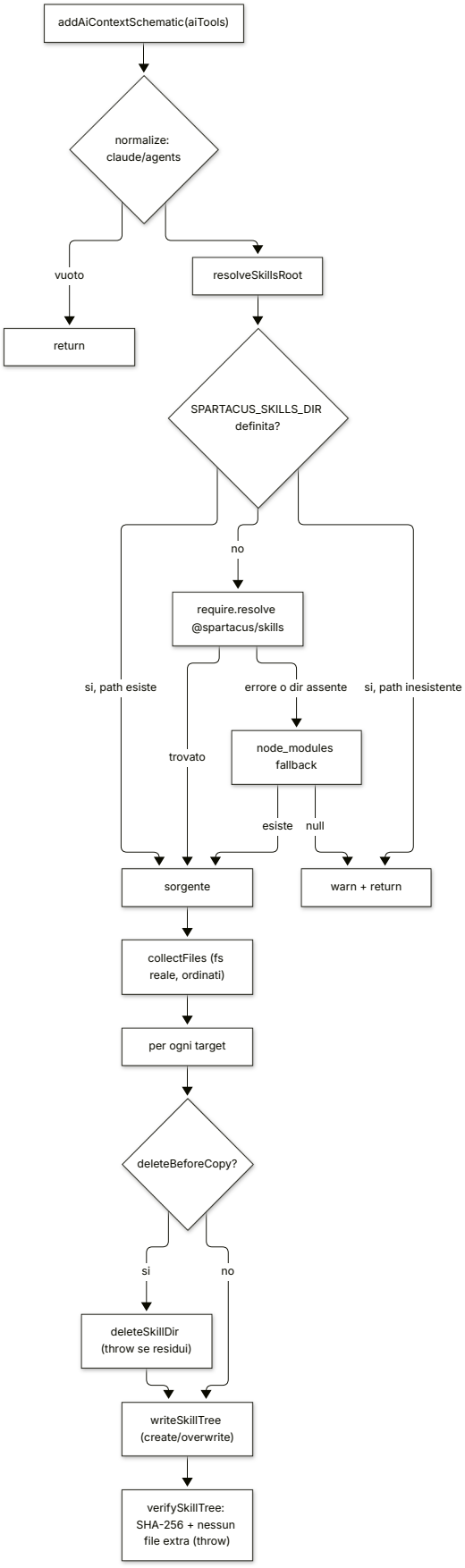


TAVOLA 43.2 **4.4 Schematic ai-context**



44

Strategia di test e tooling del monorepo

Tre runner di unit test, utility di test, E2E SSR e Cypress, validazioni statiche, tool interni.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Spartacus è un monorepo Nx con npm workspaces su `feature-libs/*`, `integration-libs/*` e `core-libs/*` (`package.json:L6-L10`). Le librerie sono decine, ognuna pubblicata come pacchetto npm a sé. Per tenerle coerenti servono test a più livelli e un tooling che li orchestri in CI senza far esplodere i tempi. Questo capitolo descrive quel tooling come si presenta allo SHA, che è uno stato **di transizione**:

- **tre runner di unit test convivono**. Vitest per quasi tutte le librerie, Karma/Jasmine per pochi progetti residui, Jest per gli schematics, per `@spartacus/setup` e per gli stili SCSS. Lo script CI li tratta come tre gruppi distinti (`ci-scripts/unit-tests.sh:L5-L12`, `ci-scripts/unit-tests.sh:L88-L106`);
- **le utility di test** non sono pubblicate. Sono mock condivisi in `core-libs/storefront/shared/test/` e cartelle `testing/` dentro `core-libs/core/src/...`, raggiungibili con alias di path definiti solo per i test (`core-libs/storefront/vitest.config.ts:L14-L34`, `core-libs/storefront/tsconfig.spec.json:L16-L35`);
- **gli E2E SSR** (`projects/ssr-tests`) sono una suite Jest su Node. Avvia il server SSR dell'app demo come processo figlio e mette un proxy HTTP tra l'app e il backend OCC, così da poter iniettare errori e ritardi (`projects/ssr-tests/src/utils/ssr.utils.ts:L47-L61`, `projects/ssr-tests/src/utils/proxy.utils.ts:L58-L115`);
- **gli E2E browser** usano Cypress (`projects/storefrontapp-e2e-cypress`). Qui interessano solo struttura, convenzioni di nome e modo di lancio in CI (`ci-scripts/e2e-cypress.sh:L7-L15`);
- **le validazioni statiche** sono script bash con grep anti-`fdescribe`, ESLint con regole custom del workspace, stylelint con un plugin custom, prettier e i18n-lint (`ci-scripts/validate-lint.sh:L32-L109`);
- **i tool interni** sotto `tools/` sono i builder ng-packagr custom, la coerenza di dipendenze e tsconfig (`config:check`), la pipeline API Extractor per i dati delle migrazioni e un registry Verdaccio per provare gli schematics (`tools/build-lib/builders.json:L1-L14`, `tools/config/index.ts:L129-L136`, `tools/breaking-changes/package.json:L7-L27`, `tools/schematics/testing.ts:L92-L109`).

Il capitolo non insegna a scrivere test Angular in generale. Mostra **come questo repository decide cosa eseguire, con quale runner, con quali mock, e dove si rompe**. È utile a chi contribuisce a Spartacus e a chi, in un progetto cliente, vuole riusarne gli schemi (proxy SSR, cookie di config E2E, sharding pesato).

Capitoli collegati: l'engine SSR verificato dagli E2E SSR è in 39 — SSR ottimizzato e, per la parte Angular, in 07 — SSR, hydration e TransferState. L'app demo e il suo `server.ts` sono in 47 — L'app demo storefrontapp. I feature toggle, e quindi `provideMockFeatureToggles` e `MockFeatureDirective`, sono in 42 — Feature toggle e deprecazioni. Il mixin `forFeature`/`forVersion` testato con sass-true è in 38 — Stili e theming. Gli schematics, con i loro test `*_spec.ts` e i dati generati da `tools/breaking-changes`, sono in 43 — Schematics e migrazioni. Builder e schematics dal lato Angular sono in 08 — Schematics e builder.

> Convenzione dei file (dal dossier): `*.spec.ts` sono test Vitest, Karma o Jest unit; `*_spec.ts` sono test Jest degli schematics; `*.spec.scss` sono test sass-true; `*.cy.ts` sono spec Cypress.

2. Mappa dei file

PATH	RUOLO
`ci-scripts/unit-tests.sh`	dispatcher dei tre runner unit (`$TEST_RUNNER`, `$UNIT_TEST_GROUP_PROJECTS`, modalità affected)
`ci-scripts/compute-unit-shards.sh`	costruisce la matrice GitHub Actions degli shard (Karma/Jest/Vitest)
`ci-scripts/shard-lib.sh`	funzioni condivise: enumerazione progetti via `nx show projects`, peso per righe di spec, partizione LPT
`.github/actions/prepare-unit-shards/action.yml`	action composita che pubblica la matrice come output
`.github/workflows/ci-continuous-integration.yml`, `.github/workflows/ci-merge-checks.yml`, `.github/workflows/ci.yml`	workflow PR (`ci-continuous-integration.yml`: shard, unit, lint), merge check su PR pronta per la review (`ci-merge-checks.yml`: build ed e2e, con gating di `validate-e2e-execution.sh`), push/dispatch (`ci.yml`: pipeline completa, inclusi ssr-tests) (`.github/workflows/ci-merge-checks.yml:L1-L10`, `.github/workflows/ci.yml:L1-L9`)
`nx.json`	`targetDefaults` (cache, `dependsOn`), inferenza disattivata
`core-libs/core/project.json`, `core-libs/storefront/project.json`	target `build` (builder custom), `test-vitest`, `lint`
`feature-libs/cart/project.json`	modello feature lib: `test-vitest` + `test-jest` (schematics)
`integration-libs/opf/project.json`, `integration-libs/opf/karma.conf.js`, `integration-libs/opf/test.ts`	esempio di libreria rimasta su Karma/Jasmine
`core-libs/core/vitest.config.ts`	config Vitest di core: Analog plugin, jsdom, `forks`, coverage v8, junit
`core-libs/storefront/vitest.config.ts`	come sopra più alias ordinati per gli entry di test e i sorgenti core/storefront
`feature-libs/asm/vitest.config.ts`	esempio di config Vitest di feature lib (senza `restoreMocks`)
`testing/setup-vitest.ts`	setup Vitest condiviso: zone.js, compiler JIT, `initTestEnvironment` browser
`testing/patch-object-define-property.ts`	patch globale di `Object.defineProperty` per `spyOnProperty` (Karma)
`vitest-globals.d.ts`	riferimento ambient a `vitest/globals` per l'editor
`core-libs/core/src/test.ts`	entry Karma legacy di core (orfano: core non ha più target Karma)
`core-libs/core/tsconfig.spec.json`, `core-libs/storefront/tsconfig.spec.json`, `feature-libs/cart/tsconfig.spec.json`	types Vitest e `paths` degli entry di test
`core-libs/storefront/shared/test/*`	`MockFeatureDirective`, `MockFeatureLevelDirective`, `MockFeatureDirectivesModule`, `MockMediaComponent`, `MockWinRef`
`core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts`	`provideMockFeatureToggles` / `MockFeatureTogglesController`
`core-libs/setup/jest.config.js`, `core-libs/setup/setup-jest.ts`	Jest di `@spartacus/setup` con `ServerTestingModule`
`core-libs/styles/jest.schematics.config.js`, `core-libs/styles/test-jest.ts`, `core-libs/styles/test/scss.spec.js`	Jest che esegue i `.spec.scss` tramite sass-true

PATH	RUOLO
core-libs/styles/test/*.spec.scss, core-libs/styles/test/versioning-test-util.scss	test di forFeature, forVersion, compareVersion
core-libs/schematics/jest.schematics.config.js, feature-libs/cart/jest.schematics.config.js, jest-esm-matchers.js	Jest per gli schematics (*_spec.ts)
projects/ssr-tests/{project.json,package.json,jest.config.js,tsconfig.json,eslint.config.mjs,README.md}	progetto E2E SSR
projects/ssr-tests/validate-ssr-build.ts	globalSetup Jest: build prod, marker proxy, porta libera
projects/ssr-tests/src/environments/custom-test-environment.ts	environment Jest che allega i log SSR agli errori
projects/ssr-tests/src/utils/{ssr,proxy,http,log}.utils.ts	avvio/arresto server SSR, proxy backend su 9002, client HTTP, parsing .ssr.log
projects/ssr-tests/src/ssr-testing.spec.ts	scenari SSR (200, 404, 500, timeout backend, cache)
projects/storefrontapp-e2e-cypress/{package.json,cypress.config.ts,cypress.config.ci.ts}	script cy:* e config Cypress locale/CI
projects/storefrontapp-e2e-cypress/cypress/support/cx-config.commands.ts	comando cy.cxConfig() (stesso cookie degli E2E SSR)
ci-scripts/e2e-cypress.sh, ci-scripts/e2e-a11y-helpers.sh, ci-scripts/test-distribution.sh, ci-scripts/test-distribution.config.sh	esecuzione e distribuzione degli e2e Cypress
ci-scripts/validate-e2e-execution.sh	decide se servono gli e2e in base ai file cambiati
ci-scripts/validate-lint.sh	validazioni statiche (focus test, tsconfig prod, stylelint, ESLint, prettier, i18n)
ci-scripts/npm-commands.sh	wrapper bash degli script npm (build CI con nx run-many)
eslint.config.mjs, projects/storefrontapp/eslint.config.mjs, tools/eslint-rules/*	flat config ESLint, module boundaries, 11 regole custom
.stylelintrc.json, tools/stylelint-rules/*	stylelint e plugin valid-for-feature-mixin-argument
tools/build-lib/*	builder augmented-types e declaration-merging
tools/config/*	config:check / config:update (dipendenze, tsconfig paths, lock file, versione migrazione AI)
tools/breaking-changes/*	API Extractor → confronto API → dati per le migrazioni
tools/schematics/testing.ts	Verdaccio locale per provare gli schematics pubblicati

3. Come funziona passo per passo

3.1 Chi usa quale runner

Il criterio pratico è il **nome del target nx**:

TARGET NX	RUNNER	EXECUTOR	CHI LO HA
`test-vitest`	Vitest	`nx:run-commands` → `npx vitest run --config vitest.config.ts` nella root del progetto	`core`, `storefrontlib` e le feature/integration lib migrate (`core-libs/core/project.json:L20-L26`, `core-libs/storefront/project.json:L21-L27`, `feature-libs/cart/project.json:L20-L26`)
`test`	Karma/Jasmine	`@angular-devkit/build-angular:karma` con `main: test.ts` e `karma.conf.js` locale	`opf`, `epd-visualization`, `punchout`, `storefrontapp` (`integration-libs/opf/project.json:L20-L28`); anche `eslint-rules`, ma con executor `@nx/jest:jest` (`tools/eslint-rules/project.json:L7-L14`)
`test-jest`	Jest	`nx:run-commands` → `npm run test` o `npm run test:schematics`	schematics delle lib (`feature-libs/cart/project.json:L27-L32`), `setup` (`core-libs/setup/project.json:L20-L26`), `storefrontstyles` (`core-libs/styles/project.json:L6-L12`), `ssr-tests` (`projects/ssr-tests/project.json:L1-L22`)

I numeri allo SHA: 32 `vitest.config.ts` (2 core-libs, 20 feature-libs, 10 integration-libs) e 4 `karma.conf.js` (`projects/storefrontapp`, `integration-libs/epd-visualization`, `integration-libs/opf`, `integration-libs/punchout`). La fonte è un glob sul working tree riportato nel dossier (F13, es. `integration-libs/opf/karma.conf.js:L1-L49`, `core-libs/core/vitest.config.ts:L1-L61`). Le devDependencies root contengono tutte e tre le toolchain: `vitest ^4.1.10`, `@analogjs/vite-plugin-angular`, `@nx/vite`, `jsdom`, `karma ~6.4.4`, `jasmine-core ~6.3.0`, `jest ^30.2.0`, `jest-preset-angular ^17.0.0`, `sass-true ^10.0.0` (`package.json:L200-L293`).

Una feature lib migrata ha quindi **due** target di test: `test-vitest` per gli unit e `test-jest` per gli schematics (`feature-libs/cart/project.json:L20-L32`). `core` e `storefrontlib` invece hanno solo `test-vitest` (`core-libs/core/project.json:L7-L36`).

3.2 Configurazione Vitest

`core-libs/core/vitest.config.ts` è il modello:

1. **Plugin**: `angular()` di Analog compila componenti e decorator Angular per Vite; `nxViteTsPaths()` risolve gli alias dei tsconfig. La root è la cartella della lib (`core-libs/core/vitest.config.ts:L7-L13`).
2. **Esecuzione**: `pool: 'forks'`, `watch: false`, `globals: true` (`describe`/`it`/`vi` senza import), `restoreMocks: true`, `environment: 'jsdom'`, setup `../../testing/setup-vitest.ts`, include `**/*.spec.ts` (`core-libs/core/vitest.config.ts:L14-L21`).
3. **Typecheck** con il `tsconfig.spec.json` della lib (`core-libs/core/vitest.config.ts:L22-L24`). Quel tsconfig ha `strict: false`, `module: preserve`, `moduleResolution: bundler` e i types `vitest/globals`, `vitest/importMeta`, `vite/client`, `node`, `vitest` (`core-libs/core/tsconfig.spec.json:L1-L19`).
4. **Coverage** sempre attiva: provider `v8`, reporter `text-summary`/`html`/`lcov`, output in `coverage/core` (`core-libs/core/vitest.config.ts:L25-L30`). Esclude spec, `public_api.ts`, `index.ts`, `*.module.ts`, asset, `testing/` e `schematics/` (`core-libs/core/vitest.config.ts:L31-L43`). Le soglie sono 85/85/65/85, identiche in storefront (`core-libs/core/vitest.config.ts:L44-L49`, `core-libs/storefront/vitest.config.ts:L64-L69`).
5. **Reporter** `default` + `junit` in `unit-tests-reports/unit-test-core.xml`, letto dalla CI (`core-libs/core/vitest.config.ts:L51-L59`).

Il setup condiviso `testing/setup-vitest.ts` importa `zone.js`, `zone.js/testing` e `@angular/compiler`, poi chiama `getTestBed().initTestEnvironment(BrowserTestingModule, platformBrowserTesting())` **senza opzioni** (testing/setup-vitest.ts:L7-L19). Non c'è quindi il `ZoneChangeDetectionModule` che serve nei `test.ts` di Karma (§3.3).

`core-libs/storefront/vitest.config.ts` aggiunge `resolve.alias`. Mappa i sotto-path di test (`@spartacus/core/testing/feature-toggles`, `@spartacus/storefront/testing/mock-feature-directive`, `@spartacus/core/testing/i18n`, `@spartacus/core/testing/url-testing`) e poi `@spartacus/core`/`@spartacus/storefront` sui `public_api.ts` sorgente (core-libs/storefront/vitest.config.ts:L14-L34). Il commento avverte che l'ordine conta, perché Vite applica il primo alias che corrisponde e un alias "nudo" cattura anche i deep path (core-libs/storefront/vitest.config.ts:L16-L19). I test di storefront girano quindi contro i **sorgenti** di core, non contro `dist`; il `tsconfig.spec.json` di storefront infatti include `../../core-libs/core/src/**/*.ts` (core-libs/storefront/tsconfig.spec.json:L16-L35).

Le feature lib ripetono lo schema con alias propri. `asm`, ad esempio, mappa anche `core-libs/storefront/shared/test/mock-window-ref` e non ha `restoreMocks` (feature-libs/asm/vitest.config.ts:L10-L28). Per l'editor, un file ambient alla root referenzia `vitest/globals`, perché il `tsconfig.json` root non lo include nei `types` (vitest-globals.d.ts:L7-L14).

3.3 Karma/Jasmine residuo e il `test.ts` legacy

Una libreria Karma (es. `opf`) usa l'executor `@angular-devkit/build-angular:karma` con `main: integration-libs/opf/test.ts`, polyfill `zone.js`/`zone.js/testing` e `karma.conf.js` locale (integration-libs/opf/project.json:L20-L28). Il `karma.conf.js` usa jasmine e karma-coverage, scrive un junit in `unit-tests-reports/unit-test-opf.xml` e fissa le soglie a 80/80/70/80 (integration-libs/opf/karma.conf.js:L7-L36).

Il `test.ts` di `opf` registra un `ZoneChangeDetectionModule` con `provideZoneChangeDetection()` e inizializza TestBed con `teardown: { destroyAfterEach: false }` (integration-libs/opf/test.ts:L9-L32). Il `test.ts` di **core** fa lo stesso. Il motivo è spiegato nel commento: Angular 21 causa errori NG0100 con un `test.ts` custom. Lo stesso commento annota "No longer necessary under vitest" (core-libs/core/src/test.ts:L21-L30, core-libs/core/src/test.ts:L33-L39). In più, il `test.ts` di core importa `testing/patch-object-define-property` "just after zone.js imports" (core-libs/core/src/test.ts:L7-L12).

La patch sostituisce globalmente `Object.defineProperty`. Ogni proprietà con getter non configurabile diventa `configurable: true`, così `spyOnProperty` può sovrascrivere export di terze parti come `isDevMode` (testing/patch-object-define-property.ts:L7-L36). Serve "since `zone.js@0.11.4`" e non va importata in produzione (testing/patch-object-define-property.ts:L11-L24).

Allo SHA il `test.ts` di core è **orfano**. `core` non ha né target `test` né `karma.conf.js`, e il file resta citato solo nell'`exclude` di `tsconfig.lib.json` (core-libs/core/project.json:L7-L36, core-libs/core/tsconfig.lib.json:L24-L24).

3.4 Jest: `@spartacus/setup`, stili, schematics

`@spartacus/setup` (che contiene l'engine SSR) si testa con Jest. Il target `test-jest` esegue `npm run test`, cioè `jest --config ./jest.config.js` (core-libs/setup/project.json:L20-L26, core-libs/setup/package.json:L15-L17). Il config usa il preset `jest-preset-angular` e `setupFilesAfterEnv: ['<rootDir>/setup-jest.ts']` (core-libs/setup/jest.config.js:L7-L26). Il setup inizializza TestBed con `ServerTestingModule`/`platformServerTesting()`: l'ambiente è server, non browser (core-libs/setup/setup-jest.ts:L7-L19). Il config rimappa inoltre due dipendenze interne di Angular per problemi ESM/CJS. `beasties` va verso `node_modules/beasties`, `bundled-domino.mjs` verso un wrapper CJS in `__mocks__` (core-libs/setup/jest.config.js:L12-L25). La coverage è spenta di default (`collectCoverage: false`) e ha soglie 90/80/90/90 (core-libs/setup/jest.config.js:L36-L46).

Stili (`storefrontstyles`): l'unico target è `test-jest` → `jest --config ./jest.schematics.config.js` (core-libs/styles/project.json:L6-L12, core-libs/styles/package.json:L13-L15). Il config cerca solo `**/*.spec.js` e usa `test-jest.ts` come setup (core-libs/styles/jest.schematics.config.js:L4-L20), che è vuoto: "empty, for now" (core-libs/styles/test-jest.ts:L7-L7). L'unico `*.spec.js`, `scss.spec.js`, raccoglie con `globSync` tutti i `test/**/*.spec.scss` relativi a `process.cwd()`. Per ciascuno chiama `sassTrue.runSass({ describe, it }, file)`, più un test sentinella che fallisce se non trova file (core-libs/styles/test/scss.spec.js:L1-L15). Ogni `.spec.scss` importa sass-true e il modulo da testare e usa i mixin `describe`/`it`/`assert-equal`/`assert`/`output`/`expect` di sass-true (core-libs/styles/test/features.spec.scss:L1-L11):

- `features.spec.scss` verifica `getFeatureCssClass` (prefisso `cxFeat_`) e `forFeature`. Senza secondo argomento la classe va sull'antenato: `.cxFeat_<flag> .test` (core-libs/styles/test/features.spec.scss:L14-L35). Con `'currentSelector'` va sul selettore corrente: `.test.cxFeat_<flag>` (core-libs/styles/test/features.spec.scss:L121-L141);
- `versioning-test-util.scss` definisce quattro mixin helper (`shouldDisplayForRange`, `shouldNOTDisplayForRange`, `shouldDisplayForVersion`, `shouldNOTDisplayForVersion`) che avvolgono `it` + `assert` sul mixin `forVersion` (core-libs/styles/test/versioning-test-util.scss:L1-L72);
- i tre `versioning-forVersion-0N.spec.scss` fissano le globali `$_fullVersion`, `$_majorVersion`, `$styleVersion`, `$useLatestStyles` prima di chiamare gli helper. Il file 01 usa `3.2` (core-libs/styles/test/versioning-forVersion-01.spec.scss:L1-L22), lo 02 versioni a tre cifre `5.1.3` con `// prettier-ignore` (core-libs/styles/test/versioning-forVersion-02.spec.scss:L5-L19), lo 03 i confini dei range con `5.1` (core-libs/styles/test/versioning-forVersion-03.spec.scss:L5-L20);
- `versioning-functions.spec.scss` testa `compareVersion()` con ritorni 0/1/-1, anche con argomenti stringa (core-libs/styles/test/versioning-functions.spec.scss:L5-L47).

Schematics (`*_spec.ts`): i config Jest di core e delle feature lib usano `jest-preset-angular`, `testMatch: **/*_spec.ts`, `tsconfig.schematics.json` e un `transformIgnorePatterns` costruito da `jest-esm-matchers` (core-libs/schematics/jest.schematics.config.js:L1-L23). Quel file elenca i pacchetti solo-ESM che Jest deve trasformare, come `.mjs`, i locale Angular, `ora`, `chalk` (jest-esm-matchers.js:L7-L30). Le soglie sono 90/74/90/90 in core e 90/90/90/90 in cart (core-libs/schematics/jest.schematics.config.js:L28-L35, feature-libs/cart/jest.schematics.config.js:L28-L35). Lo script `test:schematics` di una feature lib pulisce gli output compilati di core-schematics e della lib prima di lanciare Jest (feature-libs/cart/package.json:L27-L27). Lo script root `test:all-schematics` li esegue tutti con `-u`, cioè aggiornando gli snapshot (package.json:L146-L146).

3.5 Utility di test condivise

Storefront (`core-libs/storefront/shared/test/`): il barrel esporta cinque mock (core-libs/storefront/shared/test/index.ts:L7-L11).

- `MockFeatureDirective` (`[cxFeature]`) renderizza il template solo se l'argomento **non** contiene `!`. Così i rami "deprecati" (`*cxFeature="'!toggle'"`) non compaiono nei test (core-libs/storefront/shared/test/mock-feature-directive.ts:L9-L22).
- `MockFeatureLevelDirective` (`[cxFeatureLevel]`) renderizza sempre (core-libs/storefront/shared/test/mock-feature-level-directive.ts:L9-L18).
- `MockFeatureDirectivesModule` importa, senza dichiararle, le due direttive standalone (core-libs/storefront/shared/test/mock-feature-directives.module.ts:L11-L14).
- `MockMediaComponent` (`cx-media`) rende un `<img>` solo se ricava una `src` da `container.product.url` o `container.url` (core-libs/storefront/shared/test/mock-media.component.ts:L9-L25).
- `MockWinRef` importa `vi` da `vitest`. Fornisce `localStorage`/`sessionStorage` fatti di `vi.fn()`, `location.href` vuoto e `isBrowser()` sempre `true` (core-libs/storefront/shared/test/mock-window-ref.ts:L7-L23).

Gli spec li importano in tre modi: con l'alias `@spartacus/storefront/testing/...`, con un path relativo, oppure con un path dalla root del repo (core-libs/storefront/cms-components/content/banner/banner.component.spec.ts:L23-L23, core-libs/storefront/cms-components/product/product-list/container/product-list.component.spec.ts:L33-L33, feature-libs/asm/root/services/asm-enabler.service.spec.ts:L12-L12). Alcune feature lib (`asm`, `cart`, `checkout`, `customer-ticketing`, `order`, `product`, `user`) dichiarano gli alias nei `compilerOptions.paths` del proprio `tsconfig.spec.json` (feature-libs/cart/tsconfig.spec.json:L35-L40); le altre feature lib con `tsconfig.spec.json` non hanno `paths`. `tools/config/manage-dependencies.ts` sa che questi file sono "da test". Classifica come uso in spec gli import provenienti da `test.ts`, `src/test.ts`, `setup-jest.ts`, `vitest.config.ts` e da qualunque file sotto `/shared/test/`, così le loro dipendenze non finiscono tra le dipendenze di produzione (tools/config/manage-dependencies.ts:L527-L538).

Core: le utility di test stanno nelle cartelle `testing/` dei moduli, escluse dalla coverage (core-libs/core/vitest.config.ts:L38-L38). L'esempio principale è `provideMockFeatureToggles(initial)` (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L150-L158). Fa puntare `FeatureToggles`, `RootFeatureToggles`, `DefaultFeatureToggles` e il multi-provider `FeatureTogglesChunk` allo stesso `MockFeatureTogglesController`, creato nuovo a ogni setup di TestBed (core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L161-L184). Il meccanismo dei toggle è spiegato in 42.

3.6 Pattern ricorrenti negli spec

Un campione di sei spec, uno per tipo, mostra gli idiomi correnti:

- **Componente (Vitest)**: mock come classi locali, `TestBed.configureTestingModule` con `useClass`/`useValue`, `overrideComponent` per sostituire `MediaComponent`/`FeatureDirective` con i mock standalone, query con `By.css`/`By.directive` (core-libs/storefront/cms-components/content/banner/banner.component.spec.ts:L58-L74, core-libs/storefront/cms-components/content/banner/banner.component.spec.ts:L97-L129). Lo stato pilotato da un `BehaviorSubject` cambia in `beforeEach` annidati, prima di `fixture.detectChanges()` (core-libs/storefront/cms-components/content/banner/banner.component.spec.ts:L142-L155).
- **Facade NgRx (Vitest)**: store reale (`StoreModule.forRoot`, `EffectsModule.forRoot`, store module di feature), `vi.spyOn(store, 'dispatch')` e `provideMockFeatureToggles` (core-libs/core/src/site-context/facade/currency.service.spec.ts:L53-L72). Le asserzioni asincrone usano `await firstValueFrom(service.getAll())` dopo un dispatch di successo (core-libs/core/src/site-context/facade/currency.service.spec.ts:L101-L130).
- **Selettori (Vitest)**: `vi.mock('@ngrx/store', ...)` con `importOriginal`, per sostituire solo `select` (core-libs/core/src/product/store/selectors/product-reviews.selectors.spec.ts:L4-L7).
- **Adapter OCC (Vitest)**: `provideHttpClient(withInterceptorsFromDi()) + provideHttpClientTesting()`, `HttpTestingController.expectOne/flush` e `httpMock.verify()` in `afterEach` (core-libs/core/src/occ/adapters/product/occ-product-reviews.adapter.spec.ts:L29-L83). Le dipendenze reali come il `ConverterService` sono osservate con `vi.spyOn` senza cambiarne l'implementazione (core-libs/core/src/occ/adapters/product/occ-product-reviews.adapter.spec.ts:L56-L58).
- **Karma/Jasmine**: `jasmine.createSpyObj`, `.and.returnValue(...)`, callback `done` e `fail()` (integration-libs/opf/cta/core/services/opf-static-cta.service.spec.ts:L10-L79).
- **Schematics (Jest)**: `SchematicTestRunner.runExternalSchematic('@schematics/angular', ...)` per creare workspace e app, poi `add-spartacus` e `add-ssr` in `beforeAll`, infine `toMatchSnapshot()` sui file generati (core-libs/schematics/src/add-ssr/index_spec.ts:L17-L102).

Le istruzioni di progetto chiedono di preferire `firstValueFrom` alle subscription e di non condividere stato mutabile tra test (.claude/CLAUDE.md:L48-L50). Lo spec della facade segue la prima regola.

3.7 Unit test in CI: sharding pesato e dispatch

1. Il workflow PR calcola `affected` (vero per PR non `epic/*`) e invoca l'action `prepare-unit-shards` (`.github/workflows/ci-continuous-integration.yml:L51-L57`). L'action esegue `compute-unit-shards.sh` e pubblica `matrix` come output (`.github/actions/prepare-unit-shards/action.yml:L39-L51`). Il workflow di push `ci.yml` usa `affected: 'false'` (`.github/workflows/ci.yml:L64-L80`).
2. `compute-unit-shards.sh` fissa i default `KARMA_SHARDS=2`, `JEST_SHARDS=4`, `VITEST_SHARDS=4`, `AFFECTED=false` (`ci-scripts/compute-unit-shards.sh:L9-L12`).
3. Enumera i progetti **per target**. Karma = progetti con `test` meno app e solo-Jest; Jest = progetti con `test-jest` meno app; Vitest = tutti quelli con `test-vitest` (`ci-scripts/compute-unit-shards.sh:L19-L31`). L'enumerazione usa `nx show projects --with-target=<t> --json`, con `--affected` opzionale, e fallisce rumorosamente se nx fallisce (`ci-scripts/shard-lib.sh:L13-L28`).
4. Ogni progetto riceve un peso: il numero di righe "significative" dei suoi `*.spec.ts`, cioè escludendo import, commenti `//` e righe vuote, con minimo 1 (`ci-scripts/shard-lib.sh:L30-L48`).
5. `shard_partition` applica un greedy LPT (longest processing time): ordina per peso decrescente e assegna ogni progetto al bucket meno carico (`ci-scripts/shard-lib.sh:L56-L71`).
6. `matrix_legs` produce voci `{test-runner, shard-name: "<runner>-i/n", projects}` ed elimina i bucket vuoti (`ci-scripts/compute-unit-shards.sh:L34-L47`). In modalità run-all una matrice vuota è un errore (`ci-scripts/compute-unit-shards.sh:L57-L61`).
7. Ogni leg della matrice esegue `ci-scripts/unit-tests.sh` con `$TEST_RUNNER` e `$UNIT_TEST_GROUP_PROJECTS` impostati (`.github/workflows/ci-continuous-integration.yml:L59-L81`).
8. `unit-tests.sh`, con la variabile valorizzata, chiama `run_all_unit_tests` (`ci-scripts/unit-tests.sh:L108-L109`), che smista sul runner (`ci-scripts/unit-tests.sh:L88-L106`):
   - `karma` → `nx run-many --target=test ... -- --no-watch --source-map --code-coverage --browsers ChromeHeadless` (`ci-scripts/unit-tests.sh:L30-L42`);
   - `jest` → `nx run-many --target=test-jest ... -- --coverage --runInBand` (`ci-scripts/unit-tests.sh:L44-L56`);
   - `vitest` → `nx run-many --target=test-vitest ... -- --coverage` (`ci-scripts/unit-tests.sh:L58-L68`).
9. Il ramo Vitest arriva a `npx vitest run --config vitest.config.ts` nella cartella della lib (`core-libs/core/project.json:L20-L26`). Da lì valgono setup, jsdom, coverage e junit descritti in §3.2.

`shard-lib.sh` è condiviso anche con una pipeline Azure/piper tramite `split-unit-test-projects.sh`, che non è nel repo (`ci-scripts/shard-lib.sh:L3-L7`).

3.8 E2E SSR (`projects/ssr-tests`)

Il progetto nx `ssr-tests` (`projectType: library`) ha i target `lint` e `test-jest` (`npm run test --verbose`) (`projects/ssr-tests/project.json:L1-L22`). Lo script `test` del package esegue `jest --config ./jest.config.js`; le peerDependencies sono `http-proxy`, `jest-circus`, `jest-environment-node` (`projects/ssr-tests/package.json:L13-L24`). Il config trasforma con `ts-jest`, cerca `*.spec.ts`, usa un `testEnvironment` custom e un `globalSetup` (`projects/ssr-tests/jest.config.js:L9-L26`).

Sequenza completa:

1. **Build prerequisita**: `npm run build:libs && npm run build:ssr:local-http-backend` (`projects/ssr-tests/README.md:L7-L9`). L'env `local-http` imposta `CX_BASE_URL=http://localhost:9002`, cioè la porta del proxy di test (`package.json:L57-L57`, `.env-cmdrc:L11-L13`).
2. **Lancio**: `npm run test:ssr` / `test:ssr:ci` caricano l'env `dev`/`ci` con `env-cmd` e invocano `nx test ssr-tests` (`package.json:L156-L157`). Gli env `dev` e `ci` puntano `CX_BASE_URL` a backend remoti (`.env-cmdrc:L2-L4`, .

env-cmdrc:L14-L16), e lo spec usa quel valore come target del proxy (projects/ssr-tests/src/ssr-testing.spec.ts:L6-L6). La **build** punta dunque al proxy, mentre il **proxy** punta al backend reale.

3. globalSetup validateSsrBuild() controlla tre precondizioni dichiarate nella JSDoc: build prod, base OCC sul proxy locale, porta SSR libera (projects/ssr-tests/validate-ssr-build.ts:L7-L31). Lancia un errore se manca dist/storefrontapp/server/server.mjs, se nessun file di dist/storefrontapp/server contiene CX_BASE_URL:"http://localhost:9002", se un file server o browser contiene ngDevMode, o se la porta 4000 è occupata (projects/ssr-tests/validate-ssr-build.ts:L40-L58, projects/ssr-tests/validate-ssr-build.ts:L83-L141). Per la porta tenta un listen: EADDRINUSE significa occupata, qualunque altro errore conta come libera (projects/ssr-tests/validate-ssr-build.ts:L147-L165).
4. beforeEach → startSsrServer() svuota .ssr.log e lancia npm run serve:ssr della root in una shell detached. Passa SSR_CACHE, SSR_TIMEOUT, PORT e NODE_TLS_REJECT_UNAUTHORIZED=0, ridirige stdout e stderr nel log e attende il testo Node Express server listening on (projects/ssr-tests/src/utils/ssr.utils.ts:L47-L61). serve:ssr è NG_ALLOWED_HOSTS=localhost node dist/storefrontapp/server/server.mjs (package.json:L152-L152). Il server.ts demo legge SSR_TIMEOUT e SSR_CACHE per le SsrOptimizationOptions e stampa il messaggio atteso (projects/storefrontapp/src/server.ts:L22-L27, projects/storefrontapp/src/server.ts:L93-L93).
5. **Nel test** → startBackendProxyServer({ target, responseInterceptor? }) crea un http-proxy con secure: false e selfHandleResponse solo se c'è un interceptor, e ascolta sempre sulla porta 9002 (projects/ssr-tests/src/utils/proxy.utils.ts:L58-L115). Riscrive l'header host con il target (projects/ssr-tests/src/utils/proxy.utils.ts:L75-L77). Con interceptor bufferizza il body, lo decomprime (gzip/deflate) e lo passa all'interceptor, che deve chiamare res.end (projects/ssr-tests/src/utils/proxy.utils.ts:L79-L105, projects/ssr-tests/src/utils/proxy.utils.ts:L118-L127).
6. **Richiesta** → sendRequestToSsrServer({ path, testConfig? }) fa una GET su localhost:4000 con un header Cookie costruito da testConfig (projects/ssr-tests/src/utils/http.utils.ts:L13-L16, projects/ssr-tests/src/utils/http.utils.ts:L42-L54). Il cookie si chiama cxConfigE2E e contiene JSON URI-encoded; lo legge il TestConfigModule dell'app demo (projects/ssr-tests/src/utils/http.utils.ts:L75-L89, projects/storefrontapp/src/app/private/private.providers.ts:L54-L54). In questo modo un'app buildata una volta sola si può testare con varianti di config iniettate a runtime (projects/ssr-tests/src/utils/http.utils.ts:L27-L37).
7. **Asserzioni** su status code e messaggi di log. getLogsObjects legge ./.ssr.log e scarta le righe non JSON, getLogsMessages estrae message (projects/ssr-tests/src/utils/log.utils.ts:L18-L18, projects/ssr-tests/src/utils/log.utils.ts:L59-L80).
8. **Su fallimento**, CustomTestEnvironment (che estende jest-environment-node) intercetta test_done e allega i log SSR pretty-printed come cause sia dell'errore originale sia di quello asincrono, ad esempio un timeout (projects/ssr-tests/src/environments/custom-test-environment.ts:L11-L43). Usa cause perché è l'unica proprietà, oltre a message, che Jest stampa (projects/ssr-tests/src/environments/custom-test-environment.ts:L26-L29).
9. afterEach: backendProxy.close() e killSsrServer(), che invia il segnale al process group (-child.pid) e risolve all'exit (projects/ssr-tests/src/ssr-testing.spec.ts:L14-L17, projects/ssr-tests/src/utils/ssr.utils.ts:L67-L74).

Gli scenari verificati sono questi (projects/ssr-tests/src/ssr-testing.spec.ts:L26-L82):

- 200, con i log "Rendering started" e "Request is waiting...";
- 404 su pagina inesistente;
- 404 se cms/pages risponde 404;
- 500 se fallisce un'altra API (cms/components);
- 500 se /languages è ritardata oltre un backend.timeout.server di 4000 ms iniettato via testConfig. Il test verifica anche il log "exceeded expected time of 4000ms and was aborted" (projects/ssr-tests/src/

ssr-testing.spec.ts:L92-L129);

- con cache attiva, la seconda richiesta deve loggare `Render from cache (...)`. Se però il primo render è fallito con 404, la seconda **non** deve venire dalla cache (projects/ssr-tests/src/ssr-testing.spec.ts:L133-L196).

Il timeout Jest è `DEFAULT_SSR_TIMEOUT` (20000 ms) e raddoppia nei test che chiamano il server due volte (projects/ssr-tests/src/ssr-testing.spec.ts:L6-L12, projects/ssr-tests/src/ssr-testing.spec.ts:L164-L165). La pagina usata è `/contact`, "less busy" della home (projects/ssr-tests/src/ssr-testing.spec.ts:L12-L12). Il comportamento dell'engine sotto test (fallback CSR, cache, timeout) è descritto in 39.

3.9 E2E Cypress (struttura)

`cypress/e2e` contiene `a11y` (b2b, b2c, helpers), `b2b` (regression, vendor), `regression` (per dominio), `ssr` (2 spec) e `vendor` (per integrazione) (projects/storefrontapp-e2e-cypress/cypress/e2e/, listing nel dossier F165). Le convenzioni di nome separano `*.core-e2e.cy.ts` (sottoinsieme core), `*.e2e.cy.ts`, `*.e2e-flaky.cy.ts` e `*.a11y-e2e.cy.ts` (projects/storefrontapp-e2e-cypress/package.json:L18-L19, projects/storefrontapp-e2e-cypress/package.json:L61-L61). Gli script root `e2e:*` delegano ai `cy:*` del package con `npm --prefix` (package.json:L65-L124).

- `cypress.config.ts` (locale): baseUrl `http://localhost:4200`, `retries` 2 sia in run sia in open, env `BASE_SITE: 'electronics-spa'`, `OCC_PREFIX: '/occ/v2'` e `CLIENT_ID` derivato da `JDK_VERSION` (projects/storefrontapp-e2e-cypress/cypress.config.ts:L9-L47).
- `cypress.config.ci.ts`: aggiunge `projectId`, `video: true`, esclude `**/*.example-e2e.cy.ts` e abilita gli screenshot solo con `CYPRESS_ENABLE_SCREENSHOTS=true` (projects/storefrontapp-e2e-cypress/cypress.config.ci.ts:L18-L49).
- `cy:run:ci` registra sulla Cypress Dashboard con `--record --parallel --ci-build-id $BUILD_NUMBER --group B2C`, escludendo vendor/b2b/ssr, a11y e my-account-v2 (projects/storefrontapp-e2e-cypress/package.json:L24-L24).
- Il comando custom `cy.cxConfig(config)` imposta lo **stesso** cookie `cxConfigE2E` degli E2E SSR (projects/storefrontapp-e2e-cypress/cypress/support/cx-config.commands.ts:L26-L28).

In CI, `ci-scripts/e2e-cypress.sh` accetta `--suite/-s`, `--environment`, `--ssr`, `--skip-build` (ci-scripts/e2e-cypress.sh:L7-L15). Senza `--skip-build` esegue `npm ci` e `build_libs_ci`, fallisce se il log contiene `Warning: Can't resolve all parameters for`, poi esegue `build:csr` (ci-scripts/e2e-cypress.sh:L86-L112). In modalità SSR serve l'app con `serve:ssr:ci` e sceglie `e2e:run:ci:core:ssr` (PR non epic, commit di bot) oppure `e2e:run:ci:ssr` (ci-scripts/e2e-cypress.sh:L114-L166). In modalità CSR serve con `start:pwa` (http-server su 4200) e chiama `run_tests_for_suite` con scope `core` o `full` (ci-scripts/e2e-cypress.sh:L167-L197, package.json:L143-L143). Per `:a11y` carica `e2e-a11y-helpers.sh`; negli altri casi compone il nome dello script `e2e:run:ci[:core]<suite>` (ci-scripts/e2e-cypress.sh:L17-L30).

La distribuzione a11y senza recording è round-robin: file in ordine alfabetico, il file `i` va al container `(i % total) + 1` (ci-scripts/test-distribution.sh:L4-L26, ci-scripts/test-distribution.sh:L30-L75). Con `ENABLE_A11Y_RECORDING=true` si usano invece gli script `:record` della Dashboard; altrimenti si esporta `CYPRESS_SPEC_OVERRIDE` (ci-scripts/e2e-a11y-helpers.sh:L74-L125).

`validate-e2e-execution.sh` imposta `RUN_E2E=true` se il diff verso la base contiene file diversi da `*.md`, `docs/**`, `tools/**`, `*.spec.ts`, `**/schematics/**` (ci-scripts/validate-e2e-execution.sh:L5-L20, .github/workflows/ci.yml:L59-L63).

3.10 Lint e validazioni statiche

`validate-lint.sh` (job `linting`, con `LINT_PARALLEL=4` nel workflow PR: `.github/workflows/ci-continuous-integration.yml:L119-L124`) esegue in ordine:

1. verifica che `projects/storefrontapp/tsconfig.app.prod.json` **non** contenga `core-libs/storefront/public_api`, perché la build prod deve usare `dist` (`ci-scripts/validate-lint.sh:L11-L30`);
2. fallisce se trova `fdescribe(`, `describe.only(`, `fit(` o `it.only(` in un `*spec.ts` di `projects`, `feature-libs`, `integration-libs`, `core-libs` (`ci-scripts/validate-lint.sh:L32-L70`);
3. fallisce se storefrontlib importa se stessa (`from 'storefrontlib'`) (`ci-scripts/validate-lint.sh:L72-L80`);
4. stylelint → ESLint via nx (`affected` se `LINT_AFFECTED=true`, altrimenti `run-many --all`; `$LINT_PARALLEL` default 3; heap 3584 MB) → prettier → i18n-lint (`ci-scripts/validate-lint.sh:L82-L109`).

Gli script root corrispondenti sono `lint`, `lint:styles`, `prettier` (`--list-different`) e `i18n-lint`, che controlla gli attributi `alt,title,placeholder,aria-label` negli HTML (`package.json:L128-L131`).

ESLint. Il target `lint` di ogni lib usa `@angular-eslint/builder:lint` su `**/*.ts` e `**/*.html` (`core-libs/core/project.json:L27-L35`). La flat config root:

- ignora `dist`, `out-tsc`, l'intero progetto Cypress e i `vitest.config.ts` (`eslint.config.mjs:L18-L27`);
- impone selettori con prefisso `cx` e `prefer-standalone` come errore (`eslint.config.mjs:L51-L78`);
- applica `@nx/enforce-module-boundaries` sui tag (`eslint.config.mjs:L10-L16`, `eslint.config.mjs:L129-L154`). Nessuno dipende da `type:app`; `type:feature` non dipende da `type:integration`; `type:ui` non da `type:feature`; `scope:core` non da `type:ui`/`type:feature`. `core` ha i tag `type:util` + `scope:core`, `storefrontlib` ha `type:ui` (`core-libs/core/project.json:L37-L37`, `core-libs/storefront/project.json:L38-L38`);
- con una regola `no-restricted-syntax` (warn) segnala gli `inject()` assegnati a campi `private`, che devono essere `protected` per l'estendibilità, salvo `FeatureConfigService`/`FeatureToggles` (`eslint.config.mjs:L179-L192`);
- attiva le regole del workspace (`@nx/workspace-*`) per glob di file. Le `use-provide-default-*` valgono solo sui `*.module.ts`, quelle NgRx sui `*.action*.ts`, le altre su tutti i `.ts` (`eslint.config.mjs:L231-L258`).

Le 11 regole custom sono registrate in `tools/eslint-rules/index.ts` (`tools/eslint-rules/index.ts:L76-L101`). Sono scritte con `ESLintUtils.RuleCreator`; ad esempio `no-const-enum` esiste perché `const enum` non è compatibile con `isolatedModules: true` (`tools/eslint-rules/rules/no-const-enum.ts:L9-L37`). Si testano con `RuleTester` di `@angular-eslint/test-utils` (`tools/eslint-rules/rules/no-const-enum.spec.ts:L1-L20`) dentro un progetto nx con target `test` ed executor `@nx/jest:jest` (`tools/eslint-rules/project.json:L7-L14`, `tools/eslint-rules/jest.config.ts:L8-L16`). La regola `no-storefrontapp-false-feature-toggles` è attiva solo nell'app demo (`projects/storefrontapp/eslint.config.mjs:L28-L28`).

Stylelint estende `stylelint-config-recommended-scss`, carica il plugin `./tools/stylelint-rules` con la regola `spartacus/valid-for-feature-mixin-argument` e vieta le proprietà fisiche left/right (`margin-left`, `padding-right`, `border-left`, …) e alcune shorthand logiche (`padding-inline`, `margin-block`, `border-inline`, …), oltre a `text-align: left|right` (`.stylelintrc.json:L1-L5`, `.stylelintrc.json:L22-L44`, `tools/stylelint-rules/index.js:L9-L11`).

3.11 Nx, build e tool di configurazione

- `nx.json`: `targetDefaults` con cache su `build`, `lint`, `test`, `e2e`; `build.dependsOn: ["^build"]`; `useInferencePlugins: false` e `analytics: false` (`nx.json:L1-L23`). DA VERIFICARE l'effetto di `useInferencePlugins: false` sui target ricavati dagli script npm: gli script CI escludono esplicitamente dal

target `test` progetti che hanno uno script npm `test` ma nessun target `test` nel `project.json` (ci-scripts/compute-unit-shards.sh:L19-L25, ci-scripts/unit-tests.sh:L5-L6), il che fa pensare che Nx li esponga comunque come target.

- **Build delle lib**: `core` e `storefrontlib` usano il builder custom `./tools/build-lib:declaration-merging`, `setup` usa `@angular-devkit/build-angular:ng-packagr` (core-libs/core/project.json:L8-L19, core-libs/setup/project.json:L8-L19). `tools/build-lib` registra `augmented-types`, che propaga gli export di tipi al `.d.ts` principale per la module augmentation, e `declaration-merging`, che corregge i path relativi nel `.d.ts` flattened (tools/build-lib/builders.json:L1-L14).
- `build:libs` valida i JSON di traduzione e poi builda in ordine core → storefrontlib → schematics/user → cart → … → setup → s4-service, con gruppi `concurrently --kill-others-on-fail` (package.json:L35-L35). In CI, `build_libs_ci` compila in parallelo gli schematics delle lib che hanno `build:schematics`. Poi lancia `nx run-many --target=build --configuration=production --parallel=N`, escludendo app e Cypress e sfruttando grafo e cache di nx (ci-scripts/npm-commands.sh:L14-L43). `npm-commands.sh` centralizza i wrapper bash degli script npm (`build_ssr_local_http`, `test_ssr`, …), così rinominare uno script npm non richiede di toccare più punti (ci-scripts/npm-commands.sh:L5-L7, ci-scripts/npm-commands.sh:L45-L60).
- `config:check` / `config:update` eseguono `tools/config/index.ts`, il secondo con `--fix`; `generate:deps` esegue `tools/schematics/dependency-collector.ts` (package.json:L62-L64). Il CLI accetta `--fix`, `--bump-versions`, `--generate-deps` (tools/config/index.ts:L129-L136). Il modello del repository nasce dai `package.json` in `{core-libs,feature-libs,integration-libs,projects}/*`, esclusi app demo e Cypress (tools/config/index.ts:L228-L263). La sequenza è: lock file (fail-fast), dipendenze, tsconfig paths, versione della migrazione AI, `generate:deps` (con `--compare` in modalità check); in fix aggiunge `npm install` e `prettier:fix`; se ci sono errori imposta `process.exitCode = 1` (tools/config/index.ts:L330-L371). `syncAiMigrationVersion` allinea la versione di `add-or-update-ai-skills` in `migrations.json` a quella di `@spartacus/schematics` (tools/config/index.ts:L280-L325). `tsconfig-paths.ts` gestisce i `compilerOptions.paths` di tutti i tsconfig secondo un elenco di assunzioni dichiarate, ad esempio "all dependencies to other spartacus libs are peerDependencies" (tools/config/tsconfig-paths.ts:L7-L21).
- **API Extractor** (`tools/breaking-changes`) è un pacchetto a sé con la pipeline `extract-*` → `parse-*` → `compare` → `gen-*` → `regenerate` (tools/breaking-changes/package.json:L7-L27). `extract.ts` trova tutti i `dist/**/public_api.d.ts` del clone e lancia API Extractor su ognuno (tools/breaking-changes/extract.ts:L29-L44). La versione target sta in `common.ts` (`NEW_VERSION`) e determina anche la cartella delle migrazioni schematics in cui finiscono i dati (tools/breaking-changes/common.ts:L12-L18). I generatori `gen-const`, `gen-deleted`, `gen-moved`, `gen-methods-props` producono i dati per gli schematics di migrazione (tools/breaking-changes/readme.md:L57-L66); vedi 43.
- **Verdaccio**: `tools/schematics/testing.ts` avvia un registry locale su `http://localhost:4873/`, punta `@spartacus:registry` lì e pubblica i pacchetti per provare gli schematics a mano (tools/schematics/testing.ts:L92-L109, tools/schematics/testing.ts:L150-L156).

3.12 Pipeline CI completa

In `ci.yml` (push su `develop`, `develop-*`, `release/*` e dispatch manuale: .github/workflows/ci.yml:L1-L8) i job sono, in ordine (.github/workflows/ci.yml:L28-L358):

- `detect_changes`, `no_retries`, `validate_e2e_execution`;
- `prepare_unit_shards`, `unit_tests`, `linting`;
- `build_libs`, `build_apps` (matrice B2C CSR, B2B CSR, B2C SSR);
- `b2c_e2e_tests` (5 container), `b2c_ssr_e2e_tests`, `b2b_e2e_tests` (4), `a11y_e2e_tests` (3), `ssr_tests`;
- `build_conclusion`, `send_slack_message`.

`build_libs` usa l'action `build-integrity-check` (fasi `snapshot`/`verify`) per verificare che la build non modifichi i sorgenti, e mette in cache `dist` (`.github/workflows/ci.yml:L132-L161`). `ssr_tests` ripristina le lib buildate, builda con `build_ssr_local_http` e lancia `test_ssr` (`.github/workflows/ci.yml:L299-L322`). Gli e2e partono con `e2e-cypress.sh --skip-build`, a cui si aggiunge `--ssr`, `-s b2b` o `-s a11y` (`.github/workflows/ci.yml:L232-L298`).

---

4. Diagrammi

4.1 Unit test in CI: dalla matrice al runner

■ **TAVOLA 44.1** diagramma a tutta pagina, in fondo al capitolo

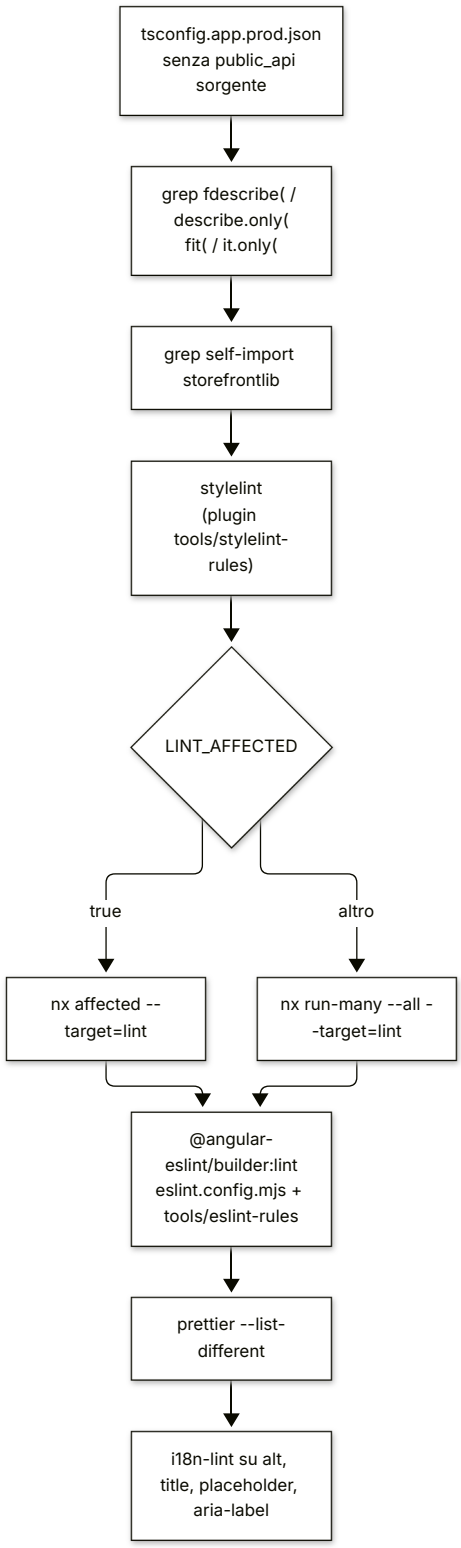
Riferimenti dei nodi: `.github/workflows/ci-continuous-integration.yml:L51-L81`, `.github/actions/prepare-unit-shards/action.yml:L39-L51`, `ci-scripts/compute-unit-shards.sh:L19-L55`, `ci-scripts/shard-lib.sh:L13-L71`, `ci-scripts/unit-tests.sh:L30-L68`, `ci-scripts/unit-tests.sh:L88-L106`, `core-libs/core/project.json:L20-L26`, `core-libs/core/vitest.config.ts:L14-L59`.

4.2 Un test E2E SSR con proxy e interceptor

■ **TAVOLA 44.2** diagramma a tutta pagina, in fondo al capitolo

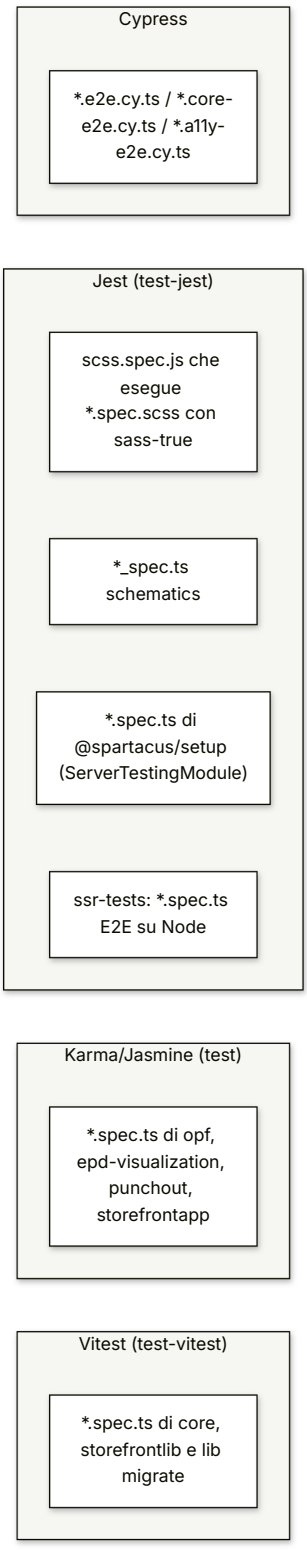
Riferimenti: `projects/ssr-tests/validate-ssr-build.ts:L83-L141`, `projects/ssr-tests/src/utils/ssr.utils.ts:L47-L74`, `projects/ssr-tests/src/utils/log.utils.ts:L82-L99`, `projects/ssr-tests/src/utils/proxy.utils.ts:L58-L127`, `projects/ssr-tests/src/utils/http.utils.ts:L42-L89`, `projects/ssr-tests/src/environments/custom-test-environment.ts:L19-L43`, `projects/ssr-tests/src/ssr-testing.spec.ts:L14-L82`, `.env-cmdrc:L2-L16`.

4.3 Validazioni statiche (`validate-lint.sh`)



Riferimenti: `ci-scripts/validate-lint.sh:L11-L109`, `core-libs/core/project.json:L27-L35`, `eslint.config.mjs:L18-L258`, `tools/eslint-rules/index.ts:L76-L101`, `.stylelintrc.json:L1-L44`, `package.json:L128-L131`.

4.4 Mappa dei runner per tipo di file



Riferimenti: core-libs/core/project.json:L20-L26, integration-libs/opf/project.json:L20-L28, core-libs/schematics/jest.schematics.config.js:L1-L23, core-libs/setup/setup-jest.ts:L7-L19, core-libs/styles/test/scss.spec.js:L1-L15, projects/ssr-tests/jest.config.js:L9-L26, projects/storefrontapp-e2e-cypress/package.json:L18-L24.

5. Estratti di codice reale

5.1 Dispatch del runner — ci-scripts/unit-tests.sh:L88-L118

BASH

```bash
function run_all_unit_tests {
    case "$TEST_RUNNER" in
        karma)
            run_karma_group_tests
            ;;
        jest)
            run_jest_group_tests
            ;;
        vitest)
            run_vitest_group_tests
            ;;
        *)
            run_karma_group_tests
            run_jest_group_tests
            run_vitest_group_tests
            ;;
    esac
    return 0
}

if [[ -n "${UNIT_TEST_GROUP_PROJECTS:-}" ]]; then
    run_all_unit_tests
elif [[ "${GITHUB_EVENT_NAME}" == "pull_request" ]]; then
    if [[ "${GITHUB_HEAD_REF}" == epic/* ]]; then
        run_all_unit_tests
    else
        run_affected_unit_tests
    fi
else
    run_all_unit_tests
fi
```

5.2 Partizione LPT — ci-scripts/shard-lib.sh:L56-L71

BASH

```bash
# Greedy longest-processing-time partition of a weighted list into N buckets.
# $1 = JSON array of {name,weight}, $2 = number of buckets.
# Emits a JSON array of buckets, each a JSON array of project names (sorted).
shard_partition() {
    local weighted="$1" n="$2"
    jq -nc --argjson list "$weighted" --argjson n "$n" '
        ($list | sort_by(-.weight)) as $sorted
        | reduce $sorted[] as $p ({buckets: [range(0;$n) | {items:[], load:0}]};
```

```
            (.buckets | to_entries | min_by(.value.load).key) as $idx
            | .buckets[$idx].items += [$p.name]
            | .buckets[$idx].load  += $p.weight
          )
        | .buckets | map(.items | sort)
    '
    return 0
}
```

5.3 Peso di un progetto — `ci-scripts/shard-lib.sh:L41-L48`

BASH

```
        if [[ -n "$root" && -d "$root" ]]; then
            # Count only "meaningful" spec lines: skip imports, comment-only and blank lines,
            # so the weight better reflects test volume than raw line count.
            weight=$(find "$root" -name '*.spec.ts' -exec cat {} + 2>/dev/null | grep -cvE '^\s*
(import|//|$)' | tr -d ' ')
        else
            weight=1
        fi
        [[ -z "$weight" || "$weight" -eq 0 ]] && weight=1
```

5.4 Config Vitest di core — `core-libs/core/vitest.config.ts:L11-L30`

TS

```
export default defineConfig({
  root: import.meta.dirname,
  plugins: [angular(), nxViteTsPaths()],
  test: {
    pool: 'forks',
    watch: false,
    globals: true,
    restoreMocks: true,
    environment: 'jsdom',
    setupFiles: ['../../testing/setup-vitest.ts'],
    include: ['**/*.spec.ts'],
    typecheck: {
      tsconfig: `${import.meta.dirname}/tsconfig.spec.json`,
    },
    coverage: {
      enabled: true,
      provider: 'v8',
      reporter: ['text-summary', 'html', 'lcov'],
      reportsDirectory: `${import.meta.dirname}/../../coverage/core`,
      include: ['**/*.ts'],
```

5.5 Setup Vitest condiviso — testing/setup-vitest.ts:L7-L19

```ts
import 'zone.js';
import 'zone.js/testing';
import '@angular/compiler';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserTestingModule,
  platformBrowserTesting,
} from '@angular/platform-browser/testing';

getTestBed().initTestEnvironment(
  BrowserTestingModule,
  platformBrowserTesting()
);
```

5.6 Alias ordinati — core-libs/storefront/vitest.config.ts:L14-L34

```ts
  resolve: {
    alias: {
      // NOTE: order matters. Vite matches string aliases in order (first match
      // wins), and a bare alias like '@spartacus/storefront' also matches any
      // '@spartacus/storefront/...' import. The specific deep-path aliases must
      // come BEFORE the general package aliases, otherwise they get shadowed.
      '@spartacus/core/testing/feature-toggles': `${import.meta.dirname}/../../core-
libs/core/src/features-config/feature-toggles/testing/index.ts`,
      '@spartacus/storefront/testing/mock-feature-directive': `${import.meta.dirname}/../../core-
libs/storefront/shared/test/mock-feature-directive.ts`,
      // ...
      '@spartacus/core': path.resolve(
        import.meta.dirname,
        '../core/public_api.ts'
      ),
      '@spartacus/storefront': path.resolve(
        import.meta.dirname,
        './public_api.ts'
      ),
    },
  },
```

5.7 test.ts legacy Karma di core — core-libs/core/src/test.ts:L21-L39

```ts
// Angular 21 introduced a change that causes NG0100 errors in Karma tests.
// See: https://github.com/angular/angular-cli/issues/32047
// Angular fixed this for built-in test.ts: https://github.com/angular/angular-cli/pull/32049
// Since we use a custom test.ts, we must manually provide zone change detection.
```

```
//No longer necessary under vitest
@NgModule({
  providers: [provideZoneChangeDetection()],
})
class ZoneChangeDetectionModule {}

// First, initialize the Angular testing environment.
getTestBed().initTestEnvironment(
  [BrowserTestingModule, ZoneChangeDetectionModule],
  platformBrowserTesting(),
  {
    teardown: { destroyAfterEach: false },
  }
);
```

5.8 Patch di `Object.defineProperty` — testing/patch-object-define-property.ts:L27-L36

TS

```
const { defineProperty } = Object;
Object.defineProperty = (object, name, meta) => {
  if (meta.get && !meta.configurable) {
    return defineProperty(object, name, {
      ...meta,
      configurable: true,
    });
  }
  return defineProperty(object, name, meta);
};
```

5.9 Mock di `*cxFeature` — core-libs/storefront/shared/test/mock-feature-directive.ts:L9-L23

TS

```
@Directive({ selector: '[cxFeature]' })
export class MockFeatureDirective {
  constructor(
    protected templateRef: TemplateRef<any>,
    protected viewContainer: ViewContainerRef
  ) {}

  @Input() set cxFeature(_feature: string) {
    // ensure the deprecated DOM changes are not rendered during tests

    if (!_feature.toString().includes('!')) {
      this.viewContainer.createEmbeddedView(this.templateRef);
    }
  }
}
```

5.10 provideMockFeatureToggles — core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L150-L166

```ts
export function provideMockFeatureToggles(
  initial: Partial<FeatureTogglesInterface> = {}
): Provider[] {
  return [
    // The controller is created fresh by the injector on each TestBed setup,
    // so mutations via .set() from one test cannot leak into the next.
    {
      provide: MockFeatureTogglesController,
      useFactory: () => createController(initial),
    },

    // New, preferred path: `inject(FeatureToggles)`.
    {
      provide: FeatureToggles,
      useFactory: (c: MockFeatureTogglesController) => c,
      deps: [MockFeatureTogglesController],
    },
```

5.11 Runner sass-true — core-libs/styles/test/scss.spec.js:L1-L15

```js
const path = require('path');
const sassTrue = require('sass-true');
const { globSync } = require('glob');

const testFiles = globSync(path.resolve(process.cwd(), `test/**/*.spec.scss`));

describe('Scss Styles test runner', () => {
  it('should discover scss spec files to run through sass-true', () => {
    expect(testFiles.length).toBeGreaterThan(0);
  });
});

describe('Scss Styles', () => {
  testFiles.forEach((file) => sassTrue.runSass({ describe, it }, file));
});
```

5.12 Un test SCSS — core-libs/styles/test/features.spec.scss:L14-L35

```scss
  @include describe('forFeature') {
    @include it('should prepend the feature class to the ancestor selector') {
      @include assert {
        @include output(false) {
          .test {
            color: yellow;
```

```
            @include forFeature('someFeatureFlag') {
              color: green;
            }
          }
        }
        @include expect(false) {
          .test {
            color: yellow;
          }
          .cxFeat_someFeatureFlag .test {
            color: green;
          }
        }
      }
    }
```

5.13 Avvio del server SSR — `projects/ssr-tests/src/utils/ssr.utils.ts:L47-L74`

TS

```ts
export async function startSsrServer({
  port = 4000,
  cache = false,
  timeout = DEFAULT_SSR_TIMEOUT,
}: SsrServerOptions = {}) {
  LogUtils.clearSsrLogFile();

  child = childProcess.spawn(
    // `2>&1` - redirect stderr to stdout, so also `console.error` and `console.warn` messages are captured in the log file
    `NODE_TLS_REJECT_UNAUTHORIZED=0 SSR_CACHE=${cache} SSR_TIMEOUT=${timeout} PORT=${port} npm run serve:ssr --prefix ../../> .ssr.log 2>&1`,
    { detached: true, shell: true }
  );

  await LogUtils.waitUntilLogContainsText(`Node Express server listening on `);
}

// ...
export function killSsrServer() {
  return new Promise((resolve) => {
    process.kill(-child.pid);
    child.on('exit', () => {
      resolve(true);
    });
  });
}
```

5.14 Proxy con interceptor — projects/ssr-tests/src/utils/proxy.utils.ts:L58-L115

TS

```ts
export async function startBackendProxyServer(
  options: ProxyOptions
): Promise<http.Server> {
  const proxyOptions: httpProxy.ServerOptions = {
    secure: false,
    selfHandleResponse: !!options.responseInterceptor,
  };
  // ...
  const proxy = httpProxy.createProxyServer(proxyOptions);

  proxy.on('proxyReq', (proxyReq) => {
    proxyReq.setHeader('host', extractUrlWithoutProtocol(options.target));
  });

  if (options.responseInterceptor) {
    proxy.on('proxyRes', (proxyRes, req, res) => {
      // We have to buffer the response body before passing it to the interceptor
      const bodyBuffer: Buffer[] = [];
      proxyRes.on('data', (chunk) => {
        bodyBuffer.push(chunk);
      });
      proxyRes.on('end', () => {
        const body = unzipResponseBody(
          Buffer.concat(bodyBuffer),
          proxyRes.headers['content-encoding']
        );
        // ...
      });
    });
  }

  return new Promise((resolve) => {
    const server = http.createServer((req, res) => {
      proxy.web(req, res, { target: options.target });
    });

    server.listen(9002, () => {
      resolve(server);
    });
  });
}
```

5.15 Cookie di config E2E — projects/ssr-tests/src/utils/http.utils.ts:L75-L89

```ts
/**
 * Builds a cookie string with key 'cxConfigE2E' and the given `cxConfig`
 * object as a value that is stringified and URI-encoded.
 *
 * Note: `TestConfigModule` of Spartacus will read this cookie
 *       and inject the passed config chunk into Spartacus global config.
 */
function buildCxConfigE2ECookie(cxConfig?: object): string {
  if (!cxConfig) {
    return '{}';
  }
  const cookieKey = 'cxConfigE2E';
  const cookieValue = encodeURIComponent(JSON.stringify(cxConfig));
  return `${cookieKey}=${cookieValue}`;
}
```

5.16 Scenario timeout backend — projects/ssr-tests/src/ssr-testing.spec.ts:L98-L128

```ts
          const BACKEND_TIMEOUT_TIME_LIMIT = 4000;
          const API_DELAY = BACKEND_TIMEOUT_TIME_LIMIT + 1;

          backendProxy = await ProxyUtils.startBackendProxyServer({
            target: BACKEND_BASE_URL,
            responseInterceptor: ({ res, req, body }) => {
              // Delay the response from Backend API, but only for for the /languages endpoint.
              // We want Spartacus to consider this Backend API request as Timeouted (therefore failed).
              if (req.url?.includes('languages')) {
                setTimeout(() => res.end(body), API_DELAY);
              } else {
                res.end(body);
              }
            },
          });

          const response = await HttpUtils.sendRequestToSsrServer({
            path: REQUEST_PATH,
            testConfig: {
              backend: { timeout: { server: BACKEND_TIMEOUT_TIME_LIMIT } },
            },
          });

          expect(response.statusCode).toEqual(500);

          const logsMessages = LogUtils.getLogsMessages();
          expect(logsMessages.join('\n')).toMatch(
            new RegExp(
```

```
            `Error: Request to URL '[^']*\/languages' exceeded expected time of
${BACKEND_TIMEOUT_TIME_LIMIT}ms and was aborted`
          )
        );
```

5.17 Log SSR come cause — projects/ssr-tests/src/environments/custom-test-environment.ts:L16-L44

```ts
export default class CustomTestEnvironment extends NodeEnvironment {
  // ...
  async handleTestEvent(event: Event, _state: State) {
    if (event.name === 'test_done') {
      event.test.errors = event.test.errors.map((errors) => {
        if (Array.isArray(errors)) {
          const [originalError, asyncError] = errors;

          const ssrLogs = getSsrLogs();
          // Error's `cause` property is the only property printed by Jest
          // besides `message` that we can utilize for attaching logs.
          // No other custom properties are printed by Jest.
          // ...
          addCauseToError({
            error: originalError, // e.g. error when an expect() fails
            cause: ssrLogs,
          });
          addCauseToError({
            error: asyncError, // e.g. error when a test timeout happens
            cause: ssrLogs,
          });
        }
        return errors;
      });
    }
  }
}
```

5.18 Module boundaries — eslint.config.mjs:L129-L154

```js
      '@nx/enforce-module-boundaries': [
        'error',
        {
          allow: [],
          allowCircularSelfDependency: true,
          depConstraints: [
            {
              sourceTag: '*',
```

```
            onlyDependOnLibsWithTags: ['*'],
            notDependOnLibsWithTags: [TAG.APP],
          },
          {
            sourceTag: TAG.FEATURE,
            notDependOnLibsWithTags: [TAG.INTEGRATION],
          },
          {
            sourceTag: TAG.UI,
            notDependOnLibsWithTags: [TAG.FEATURE],
          },
          {
            sourceTag: TAG.CORE,
            notDependOnLibsWithTags: [TAG.UI, TAG.FEATURE],
          },
        ],
      },
    ],
```

5.19 Sequenza di `config:check` — `tools/config/index.ts:L327-L347`

TS

```ts
// Run the cheap lock-file check first so a PR that forgot to update its
// `package-lock.json` fails fast, without waiting for the slower dependency,
// tsconfig and `generate:deps` steps below.
checkLockFiles(options);
if (!options.fix && errorsCount > 0) {
  console.log(chalk.red(`\nErrors: ${errorsCount}\n`));
  process.exit(1);
}

manageDependencies(repository, options);
// Keep it after dependencies, because fixes from deps might might result in different tsconfig files
manageTsConfigs(repository, options);
syncAiMigrationVersion();

if (options.generateDeps) {
  // re-generate dependencies.json file.
  execSync(`npm run generate:deps`);
} else {
  // collect and generate dependencies.json file.
  execSync(`npm run generate:deps -- --compare`);
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`$TEST_RUNNER` (env)	`karma|jest|vitest|""`	`""` (tutti e tre)	`ci-scripts/unit-tests.sh:L12-L12`; valorizzata dalla matrice in `.github/workflows/ci-continuous-integration.yml:L67-L69`	`ci-scripts/unit-tests.sh:L89-L104`
`$UNIT_TEST_GROUP_PROJECTS` (env)	lista CSV	vuota	matrice in `.github/workflows/ci-continuous-integration.yml:L67-L69`	`ci-scripts/unit-tests.sh:L31-L31`, `ci-scripts/unit-tests.sh:L108-L108`
`$KARMA_SHARDS` / `$JEST_SHARDS` / `$VITEST_SHARDS`	int	2 / 4 / 4	`ci-scripts/compute-unit-shards.sh:L9-L11`, `.github/actions/prepare-unit-shards/action.yml:L11-L22`	`ci-scripts/compute-unit-shards.sh:L49-L51`
`$AFFECTED`	`"true"|"false"`	`false`	`ci-scripts/compute-unit-shards.sh:L12-L12`	`ci-scripts/compute-unit-shards.sh:L14-L17`, `ci-scripts/compute-unit-shards.sh:L58-L58`
`$LINT_AFFECTED`	stringa bool	vuota	`.github/workflows/ci-continuous-integration.yml:L121-L121`	`ci-scripts/validate-lint.sh:L87-L87`
`$LINT_PARALLEL`	int	3 (PR CI: 4)	`ci-scripts/validate-lint.sh:L85-L85`, `.github/workflows/ci-continuous-integration.yml:L122-L122`	`ci-scripts/validate-lint.sh:L87-L93`
`$NX_BUILD_PARALLEL`	int	3	`ci-scripts/npm-commands.sh:L15-L15`	`ci-scripts/npm-commands.sh:L34-L37`
`$ENABLE_A11Y_RECORDING`	stringa bool	`false` (CI: `true`)	`ci-scripts/e2e-a11y-helpers.sh:L37-L37`, `.github/workflows/ci.yml:L297-L297`	`ci-scripts/e2e-a11y-helpers.sh:L78-L78`
`CYPRESS_SPEC_OVERRIDE`	glob/CSV	glob a11y	`ci-scripts/e2e-a11y-helpers.sh:L105-L105`	`projects/storefrontapp-e2e-cypress/package.json:L26-L27`
Vitest `test.pool` / `environment` / `globals`	string/bool	`forks` / `jsdom` / `true`	`core-libs/core/vitest.config.ts:L15-L19`	Vitest
Vitest `test.restoreMocks`	bool	`true` in core, storefront, user, cdc; assente altrove (es. asm)	`core-libs/core/vitest.config.ts:L18-L18`, `core-libs/storefront/vitest.config.ts:L39-L39`	Vitest
Vitest `coverage.thresholds`	numeri	85/85/65/85	`core-libs/core/vitest.config.ts:L44-L49`	Vitest
Jest `coverageThreshold` (setup)	numeri	90/80/90/90	`core-libs/setup/jest.config.js:L39-L46`	Jest
Jest `coverageThreshold` (schematics core / cart)	numeri	90/74/90/90 e 90/90/90/90	`core-libs/schematics/jest.schematics.config.js:L28-L35`, `feature-libs/cart/jest.schematics.config.js:L28-L35`	Jest
Karma `coverageReporter.check` (opf)	numeri	80/80/70/80	`integration-libs/opf/karma.conf.js:L28-L35`	Karma

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`SSR_PORT`	number	4000	`projects/ssr-tests/validate-ssr-build.ts:L40-L40`	`projects/ssr-tests/validate-ssr-build.ts:L134-L134`
`SsrServerOptions.port/cache/timeout`	number/bool/number	4000 / `false` / 20000	`projects/ssr-tests/src/utils/ssr.utils.ts:L47-L51`	`projects/ssr-tests/src/utils/ssr.utils.ts:L56-L56` → `projects/storefrontapp/src/server.ts:L22-L27`
`SSR_LOG_PATH`	string	`./.ssr.log`	`projects/ssr-tests/src/utils/log.utils.ts:L18-L18`	`projects/ssr-tests/src/utils/log.utils.ts:L24-L24`, `projects/ssr-tests/src/utils/log.utils.ts:L33-L33`
Porta proxy backend	number	9002 (hardcoded)	`projects/ssr-tests/src/utils/proxy.utils.ts:L112-L112`	—
`CX_BASE_URL` (env `local-http`)	URL	`http://localhost:9002`	`.env-cmdrc:L11-L13`	build SSR; marker in `projects/ssr-tests/validate-ssr-build.ts:L50-L50`
`CX_BASE_URL` (env `dev` / `ci`)	URL	backend remoto	`.env-cmdrc:L2-L4`, `.env-cmdrc:L14-L16`	target del proxy in `projects/ssr-tests/src/ssr-testing.spec.ts:L6-L6`
Cookie `cxConfigE2E`	JSON URI-encoded	—	`projects/ssr-tests/src/utils/http.utils.ts:L86-L88`, `projects/storefrontapp-e2e-cypress/cypress/support/cx-config.commands.ts:L27-L27`	`projects/storefrontapp/src/app/private/private.providers.ts:L54-L54`
`IS_CCV2_SERVER`	const bool	`false`	`projects/ssr-tests/src/utils/proxy.utils.ts:L17-L17`	`projects/ssr-tests/src/utils/proxy.utils.ts:L67-L67`
nx `targetDefaults.*.cache`	bool	`true` per build/lint/test/e2e	`nx.json:L6-L21`	Nx
`--fix` / `--bump-versions` / `--generate-deps`	flag	off	`tools/config/index.ts:L132-L134`	`tools/config/index.ts:L341-L357` (`--generate-deps`, `--fix`), `tools/config/manage-dependencies.ts:L1231-L1231` (`--bump-versions`)
`NEW_VERSION` (breaking-changes)	string	`'221121_7'`	`tools/breaking-changes/common.ts:L12-L12`	`tools/breaking-changes/common.ts:L13-L18`

7. Commenti e note notevoli dal sorgente

- "This file is required by karma.conf.js and loads recursively all the .spec and framework files" (`core-libs/core/src/test.ts:L7-L7`). Allo SHA è falso per core, che non ha più `karma.conf.js` (trappola 1).
- "Should be used with caution, and only if there is no other way to mock stuff (eg. by DI)" (`core-libs/core/src/test.ts:L10-L10`): la patch di `defineProperty` è dichiarata un'ultima risorsa.

- "//No longer necessary under vitest" (`core-libs/core/src/test.ts:L26-L26`): una nota di migrazione lasciata nel codice.
- "This file should NOT be imported in the production code, but only in unit tests!" (`testing/patch-object-define-property.ts:L11-L11`).
- "NOTE: order matters. Vite matches string aliases in order (first match wins) ..." (`core-libs/storefront/vitest.config.ts:L16-L19`).
- "// ensure the deprecated DOM changes are not rendered during tests" (`core-libs/storefront/shared/test/mock-feature-directive.ts:L17-L17`): il mock assume una convenzione (`!` = ramo deprecato) e non legge i toggle.
- "// empty, for now" (`core-libs/styles/test-jest.ts:L7-L7`).
- "mapping required to use `beasties` from node modules ..." e "Jest treats .mjs as ESM and double-wraps the default export ..." (`core-libs/setup/jest.config.js:L12-L24`): due workaround di interoperabilità ESM/CJS legati a issue Angular CLI linkate nel commento.
- La JSDoc del `globalSetup` elenca le tre precondizioni e contiene il refuso "The app us built" (`projects/ssr-tests/validate-ssr-build.ts:L7-L31`). La JSDoc del marker proxy è corrotta da un replace: "String that always USING_DEV_MODE_MARKER in the `main.js` ..." (`projects/ssr-tests/validate-ssr-build.ts:L48-L48`).
- "Error's `cause` property is the only property printed by Jest besides `message` ..." (`projects/ssr-tests/src/environments/custom-test-environment.ts:L26-L29`).
- "Note: `TestConfigModule` of Spartacus will read this cookie ..." (`projects/ssr-tests/src/utils/http.utils.ts:L79-L80`).
- "path to the page that is less "busy" than the homepage" (`projects/ssr-tests/src/ssr-testing.spec.ts:L12-L12`).
- "Shared unit-test sharding helpers, sourced by both CI pipelines" (`ci-scripts/shard-lib.sh:L3-L7`).
- "--json is passed explicitly: `nx show projects` output is environment-dependent (JSON array locally, newline plaintext on non-TTY CI runners)" (`ci-scripts/shard-lib.sh:L11-L12`).
- "Weight = total lines across the project's *.spec.ts files (a better runtime proxy than file count)" (`ci-scripts/shard-lib.sh:L30-L32`).
- "Run the cheap lock-file check first so a PR that forgot to update its `package-lock.json` fails fast" (`tools/config/index.ts:L327-L329`).
- "To make sure we get an accurate diff, do not update the dependencies of this tool ..." (`tools/breaking-changes/readme.md:L3-L3`).
- "#Force run vendor tests." è un blocco commentato in `e2e-cypress.sh` (`ci-scripts/e2e-cypress.sh:L199-L204`).
- "We've imported your old cypress plugins here. You may want to clean this up later" (`projects/storefrontapp-e2e-cypress/cypress.config.ts:L41-L42`): residuo della migrazione automatica a Cypress 10+.

---

8. Extension point

Questo capitolo riguarda tooling interno: nessuno dei simboli qui sotto è nel public API delle librerie pubblicate. Sono punti di estensione **per chi contribuisce al monorepo** o per chi copia gli schemi nel proprio progetto.

PUNTO	TIPO	DEFINIZIONE	NOTE
`MockFeatureDirective`, `MockFeatureLevelDirective`, `MockFeatureDirectivesModule`, `MockMediaComponent`, `MockWinRef`	mock di test	`core-libs/storefront/shared/test/index.ts:L7-L11`	non esportati da `core-libs/storefront/public_api.ts`; si raggiungono con gli alias `@spartacus/storefront/testing/...` (`core-libs/storefront/tsconfig.spec.json:L16-L32`)
`provideMockFeatureToggles(initial)`, `MockFeatureTogglesController`	provider di test	`core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L150-L184`	alias `@spartacus/core/testing/feature-toggles` (`core-libs/storefront/vitest.config.ts:L20-L20`). Non esiste un entry point `testing` pubblicato: l'unico `ng-package.json` di core è `core-libs/core/ng-package.json` e `core-libs/core/public_api.ts` non esporta `testing`; l'alias vive solo nelle config di test, e `**/testing/**` è esclusa dalla coverage (`core-libs/core/vitest.config.ts:L38-L38`)
Nuova lib testata con Vitest	convenzione	target `test-vitest` (`nx:run-commands`) + `vitest.config.ts` + `tsconfig.spec.json`, modello `core-libs/core/project.json:L20-L26`	viene inclusa da sola nello sharding, perché l'enumerazione è per target (`ci-scripts/compute-unit-shards.sh:L31-L31`)
Regola ESLint del workspace	file in `tools/eslint-rules/rules/`	registrazione in `tools/eslint-rules/index.ts:L61-L101`	attivazione in `eslint.config.mjs` con prefisso `@nx/workspace-` (`eslint.config.mjs:L250-L258`); test con `RuleTester`
Regola stylelint	plugin	`tools/stylelint-rules/index.js:L9-L11`	attivata in `.stylelintrc.json:L3-L5`
Builder ng-packagr custom	builder Angular	`tools/build-lib/builders.json:L1-L14`	referenziato come `./tools/build-lib:<name>` (`core-libs/core/project.json:L9-L9`)
`ProxyOptions.responseInterceptor`	callback	`projects/ssr-tests/src/utils/proxy.utils.ts:L27-L50`	simula status code, ritardi e body del backend negli E2E SSR
`TestRequest.testConfig`	config iniettata	`projects/ssr-tests/src/utils/http.utils.ts:L27-L37`	varia toggle o timeout senza ribuildare l'app
`cy.cxConfig(config)`	comando Cypress	`projects/storefrontapp-e2e-cypress/cypress/support/cx-config.commands.ts:L21-L28`	stesso cookie letto da `TestConfigModule` (`projects/storefrontapp/src/app/private/private.providers.ts:L54-L54`)

Come aggiungere uno scenario SSR (schema basato sui test esistenti):

```ts
// dentro un describe che in beforeEach chiama SsrUtils.startSsrServer(...)
it('should ... when <API> fails', async () => {
  backendProxy = await ProxyUtils.startBackendProxyServer({
    target: BACKEND_BASE_URL,
    responseInterceptor: ({ res, req, body }) => {
```

```
      if (req.url?.includes('<frammento URL OCC>')) {
        res.statusCode = 500;
      }
      res.end(body); // obbligatorio: l'interceptor gestisce la risposta
    },
  });
  const response = await HttpUtils.sendRequestToSsrServer({
    path: REQUEST_PATH,
    testConfig: { /* chunk di Config opzionale */ },
  });
  expect(response.statusCode).toEqual(500);
  expect(LogUtils.getLogsMessages()).toContain(`...`);
});
```

Ogni chiamata usata nello schema esiste in projects/ssr-tests/src/ssr-testing.spec.ts:L52-L82 (interceptor e sendRequestToSsrServer), projects/ssr-tests/src/ssr-testing.spec.ts:L114-L119 (testConfig) e projects/ssr-tests/src/ssr-testing.spec.ts:L35-L36 (LogUtils.getLogsMessages() + toContain). L'afterEach di file chiude già proxy e server (projects/ssr-tests/src/ssr-testing.spec.ts:L14-L17).

Ricette pertinenti della Parte III:

- 50 — Ricette: configurazione, DI, interceptor, eventi: ogni override via DI (service, facade, interceptor) si verifica con gli stessi strumenti del §3.6 (TestBed, provideHttpClientTesting, vi.spyOn); provideMockFeatureToggles serve per i toggle.
- 51 — Ricette: componenti CMS, outlet, route e layout: per testare un componente CMS sostituito o esteso si usa lo schema overrideComponent + mock (MockFeatureDirective, MockMediaComponent) del test del banner.
- 52 — Ricette: OCC, adapter, converter, endpoint, meta resolver: un adapter custom si testa come occ-product-reviews.adapter.spec.ts (HttpTestingController, ConverterService spiato). Il proxy degli E2E SSR mostra come simulare errori OCC end-to-end.
- 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni: per gli override SCSS con forFeature si può riusare l'approccio sass-true del §3.4.
- 54 — API pubblica vs interna, upgrade, anti-pattern, checklist: tools/breaking-changes (API Extractor) e le regole ESLint (private per FeatureToggles, protected per gli inject()) spiegano perché certe API restano stabili e altre no.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. test.ts **orfano in core.** core-libs/core/src/test.ts dichiara di servire karma.conf.js, ma core non ha più né target test né karma.conf.js. Il file sopravvive solo come esclusione in tsconfig.lib.json (core-libs/core/project.json:L7-L36, core-libs/core/tsconfig.lib.json:L24-L24). Dice anche "Has to be imported just after zone.js imports" senza importare zone.js (core-libs/core/src/test.ts:L11-L14).
2. **Il commento di** $TEST_RUNNER **mente.** Dice "karma → Jasmine/Karma (+ Jest schematics)", ma il case ha un ramo jest separato e il ramo karma non lancia Jest (ci-scripts/unit-tests.sh:L9-L9, ci-scripts/unit-tests.sh:L90-L95).

3. **Ramo affected irraggiungibile in CI.** La matrice valorizza sempre `$UNIT_TEST_GROUP_PROJECTS`, quindi `run_affected_unit_tests` di `unit-tests.sh` non viene mai eseguito: l'affected è gestito a monte in `compute-unit-shards.sh` (ci-scripts/unit-tests.sh:L108-L115, .github/workflows/ci-continuous-integration.yml:L69-L69). Fuori dalla CI lo script resta usabile da solo.
4. `eslint-rules` **nella shard "Karma".** Ha un target `test` (executor `@nx/jest:jest`), quindi rientra tra i progetti enumerati per `test` e riceve gli argomenti Karma (`--no-watch --browsers ChromeHeadless ...`) (tools/eslint-rules/project.json:L7-L14, ci-scripts/compute-unit-shards.sh:L22-L25). DA VERIFICARE l'effetto reale: senza `node_modules` non si può eseguire nx.
5. **Script npm e istruzioni obsoleti.** `test:libs` mescola `nx test-vitest` e `nx test <lib> --code-coverage` per lib già migrate (`cdc`, `s4-service`, `cdp`, `digital-payments`, `cpq-quote`), che non hanno target `test` (package.json:L147-L147, integration-libs/cdc/project.json:L20-L27). Per `setup` il target `test` deriva verosimilmente dallo script npm Jest (core-libs/setup/package.json:L16-L16, core-libs/setup/project.json:L20-L26), e infatti la CI lo tratta come JEST_ONLY (ci-scripts/compute-unit-shards.sh:L20-L20) (DA VERIFICARE, vedi trappola 6). Lo stesso vale per `test:storefront:lib` (`nx test storefrontlib`) (package.json:L148-L148, core-libs/storefront/project.json:L21-L27) e per le istruzioni di progetto, che citano ancora `nx run storefrontlib:test` (.claude/CLAUDE.md:L26-L41). Il comando giusto è `nx test-vitest <lib>` oppure `npx vitest run --config <lib>/vitest.config.ts <spec>`.
6. `nx test ssr-tests`**.** `test:ssr` invoca il target `test`, ma `ssr-tests` definisce solo `lint` e `test-jest` (package.json:L156-L157, projects/ssr-tests/project.json:L6-L20). `nx.json` ha un `targetDefaults.test`, ma contiene solo `cache: true`: non definisce un executor e da solo non crea il target (nx.json:L15-L17); inoltre `useInferencePlugins` è `false` (nx.json:L22-L22). `projects/ssr-tests/package.json` ha però lo script `test` → `jest --config ./jest.config.js` (projects/ssr-tests/package.json:L14-L14), e la CI esclude `ssr-tests` dall'enumerazione `--with-target=test` (ci-scripts/compute-unit-shards.sh:L19-L19, ci-scripts/unit-tests.sh:L5-L5): segno che Nx espone lo script npm come target `test`. Anche il job CI passa di lì: `test_ssr` → `npm run test:ssr:ci` → `nx test ssr-tests` (ci-scripts/npm-commands.sh:L45-L60, package.json:L157-L157). DA VERIFICARE che `nx test ssr-tests` si risolva sullo script npm `test`: non è eseguibile staticamente (`node_modules` assente).
7. **Cache nx solo per** `test`**.** `nx.json` mette in cache `test` ma non `test-vitest`/`test-jest`, che sono `nx:run-commands` senza `targetDefaults`. Salvo config per progetto (non trovata in core e storefront), i run Vitest/Jest non sono cacheati (nx.json:L15-L17, core-libs/core/project.json:L20-L26).
8. `restoreMocks` **non uniforme.** È presente solo in 4 dei 32 `vitest.config.ts` (core, storefront, user, cdc); manca ad esempio in `asm` (core-libs/core/vitest.config.ts:L18-L18, feature-libs/asm/vitest.config.ts:L21-L28). In una lib senza `restoreMocks` uno `vi.spyOn` che non viene ripristinato può influenzare i test successivi dello stesso file: di qui la regola di progetto sul reset in `afterEach` (.claude/CLAUDE.md:L48-L50).
9. **Esclusioni di coverage sbagliate in core.** La config esclude `'../../testing/setup-test.ts'`, che non esiste (il file reale è `setup-vitest.ts`), e `textfield/root/**`, cartella che non appartiene a core (core-libs/core/vitest.config.ts:L41-L42).
10. `MockWinRef` **dipende da Vitest.** Importa `vi` da `vitest`, quindi le lib ancora su Karma/Jasmine non possono usarlo (core-libs/storefront/shared/test/mock-window-ref.ts:L7-L7).
11. **Alias** `mock-media` **solo nel tsconfig.** `@spartacus/storefront/testing/mock-media` è nei `paths` di `tsconfig.spec.json` ma non tra gli alias di `storefront/vitest.config.ts`; a runtime la risoluzione dipende da `nxViteTsPaths()` (core-libs/storefront/tsconfig.spec.json:L23-L25, core-libs/storefront/vitest.config.ts:L14-L35). DA VERIFICARE quale tsconfig legga il plugin (root o `tsconfig.spec.json`): il codice di `@nx/vite` non è disponibile allo SHA.
12. `MockFeatureDirective` **non copre i rami** `!toggle`**.** Qualunque argomento con `!` non viene renderizzato, a prescindere dal valore reale dei toggle (core-libs/storefront/shared/test/mock-feature-directive.ts:L19-L21). Per testare il ramo legacy serve la direttiva reale con `provideMockFeatureToggles` (vedi 42).

13. **sass-true e path relativi.** I `.spec.scss` importano `'../../sass-true/sass/true'`, che rispetto a `core-libs/styles/test/` punterebbe a una `core-libs/sass-true/` inesistente (core-libs/styles/test/features.spec.scss:L1-L1). `DA VERIFICARE` come sass-true risolva i load path: il pacchetto `sass-true@10` non è in `/tmp/deps`.
14. `it` **annidati per errore.** In `features.spec.scss` due `it` (L61 e L94) sono annidati dentro un altro `it` aperto a L37 e chiuso a L119, per una parentesi messa male (core-libs/styles/test/features.spec.scss:L37-L119).
15. **Glob relativo alla cwd.** `scss.spec.js` risolve `test/**/*.spec.scss` rispetto a `process.cwd()`, quindi funziona solo se Jest parte da `core-libs/styles`, come fa il target nx (core-libs/styles/test/scss.spec.js:L5-L5, core-libs/styles/project.json:L10-L10). Lanciato da un'altra cartella, fallisce il test sentinella.
16. **Porte fisse e un server alla volta.** SSR su 4000 (validate, http.utils, default di ssr.utils) e proxy su 9002, entrambe hardcoded: i test SSR non si possono parallelizzare (projects/ssr-tests/validate-ssr-build.ts:L40-L40, projects/ssr-tests/src/utils/http.utils.ts:L13-L16, projects/ssr-tests/src/utils/proxy.utils.ts:L112-L112). Il processo figlio vive in una variabile di modulo, e `killSsrServer` fallisce se `startSsrServer` non è stato chiamato (projects/ssr-tests/src/utils/ssr.utils.ts:L40-L40, projects/ssr-tests/src/utils/ssr.utils.ts:L67-L69).
17. **Polling senza timeout proprio.** `waitUntilLogContainsText` termina solo quando il testo compare o quando scade il timeout Jest (projects/ssr-tests/src/utils/log.utils.ts:L82-L99). La ricorsione non ripassa `checkInterval`, quindi un intervallo custom vale solo per il primo controllo (projects/ssr-tests/src/utils/log.utils.ts:L94-L97). Se il server non parte (build mancante, errore di bootstrap), il test fallisce per timeout, non con un messaggio esplicito.
18. **Marker fragile.** Il controllo del proxy cerca la stringa esatta `CX_BASE_URL:"http://localhost:9002"` nei bundle minificati di primo livello: `getFiles` non è ricorsiva, e un cambio di minificazione o di naming rompe il controllo (projects/ssr-tests/validate-ssr-build.ts:L50-L50, projects/ssr-tests/validate-ssr-build.ts:L92-L108).
19. **Cookie "vuoto" malformato.** Senza `testConfig`, `buildCxConfigE2ECookie` restituisce `'{}'` come intero header `Cookie`, non una coppia `nome=valore` (projects/ssr-tests/src/utils/http.utils.ts:L82-L85).
20. **Timeout "lungo" uguale al default.** Il `describe` "With long SSR request timeout" passa `timeout: 20_000`, che coincide con `DEFAULT_SSR_TIMEOUT` (projects/ssr-tests/src/ssr-testing.spec.ts:L85-L90, projects/ssr-tests/src/utils/ssr.utils.ts:L17-L17).
21. **Consigli di build divergenti.** L'errore del `globalSetup` suggerisce `npm run build && ...`, il README `npm run build:libs && ...` (projects/ssr-tests/validate-ssr-build.ts:L63-L65, projects/ssr-tests/README.md:L7-L7).
22. `moduleNameMapper` **vuoto in ssr-tests.** `jest.config.js` legge `compilerOptions.paths` dal proprio `tsconfig.json`, che non li definisce: li erediterebbe via `extends`, ma `require` non risolve `extends` (projects/ssr-tests/jest.config.js:L1-L8, projects/ssr-tests/tsconfig.json:L1-L23).
23. **Falsi positivi e negativi in** `validate-lint.sh`**.** Il grep `'fit('` colpisce anche identificatori che finiscono in "fit(" come `benefit(` (ci-scripts/validate-lint.sh:L53-L53). Il check prettier fallisce solo se l'**ultima** riga di output contiene `projects`, quindi un file non formattato fuori da `projects/` in ultima posizione passa (ci-scripts/validate-lint.sh:L98-L99). Il messaggio di `validateTsConfigFile` usa `$LOCAL_ENV_LIB_PATH`, definita dopo la funzione: funziona solo per l'ordine di esecuzione (ci-scripts/validate-lint.sh:L15-L15, ci-scripts/validate-lint.sh:L28-L30).
24. **ESLint cieco su Cypress e Vitest config.** `projects/storefrontapp-e2e-cypress/**` e tutti i `vitest.config.ts` sono ignorati (eslint.config.mjs:L21-L26).
25. **Glob Cypress "full" senza i core.** Il pattern `--spec "...*.e2e.cy.ts"` (es. `cy:run:ci`) non corrisponde ai file `*.core-e2e.cy.ts`, perché prima di `e2e` c'è `-` invece di `.`. La suite "full" quindi non include gli spec core, mentre `cy:run:ci:b2b` usa `*e2e.cy.ts` e li include (projects/storefrontapp-e2e-cypress/package.

json:L24-L24, projects/storefrontapp-e2e-cypress/package.json:L30-L30, projects/storefrontapp-e2e-cypress/package.json:L36-L36). DA VERIFICARE se sia intenzionale: dipende dal processo del team.

26. **Script Cypress del layout < 10.** `cy:run:ci:cdc-b2b` e `cy:run:ci:vendor:cdc-b2b` usano `--config-file cypress.ci.json` e `cypress/integration/...` (projects/storefrontapp-e2e-cypress/package.json:L40-L40, projects/storefrontapp-e2e-cypress/package.json:L52-L52).
27. **a11y: recording contro round-robin.** Con `ENABLE_A11Y_RECORDING: true`, impostato in `ci.yml`, ogni container lancia l'intera suite a11y affidandosi a `--parallel` della Dashboard; `test-distribution.sh` si usa solo senza recording (.github/workflows/ci.yml:L297-L298, ci-scripts/e2e-a11y-helpers.sh:L78-L90). Anche con `--skip-build`, la parte B2B ribuilda l'app (ci-scripts/e2e-a11y-helpers.sh:L47-L52, ci-scripts/e2e-a11y-helpers.sh:L114-L115). Per la suite a11y lo script forza `SPA_ENV='ci,b2c'`, sovrascrivendo quello del job (ci-scripts/e2e-cypress.sh:L78-L80).
28. **`$RUN_E2E` solo su PR.** `validate-e2e-execution.sh` imposta `$RUN_E2E` solo se `$GITHUB_BASE_REF` è valorizzata; su `push` resta `false`, e i job e2e di `ci.yml` partono comunque grazie alla condizione `github.event_name == 'push'` (ci-scripts/validate-e2e-execution.sh:L5-L7, .github/workflows/ci.yml:L122-L122). Poiché `ci.yml` scatta solo su `push`/`workflow_dispatch` (.github/workflows/ci.yml:L1-L9), lì `$RUN_E2E` è sempre `false`: il gating vale davvero solo in `ci-merge-checks.yml`, che esegue lo script su PR `ready_for_review` e condiziona build ed e2e a `SHOULD_RUN_E2E == 'true'` (.github/workflows/ci-merge-checks.yml:L1-L7, .github/workflows/ci-merge-checks.yml:L59-L68).
29. **Versioni divergenti nei tool.** Il `package.json` root dice `2612.0.0`, `PUBLISHING_VERSION` dice `221121.22.0`, `tools/breaking-changes` usa `NEW_VERSION = '221121_7'` e il suo readme cita `'2211.19'` (package.json:L3-L3, tools/config/const.ts:L13-L13, tools/breaking-changes/common.ts:L12-L12, tools/breaking-changes/readme.md:L10-L10). Il commento di `tsconfig-paths.ts` cita inoltre `ts-node ./tools/tsconfig-paths/index.ts`, path inesistente: l'entry reale è `tools/config/index.ts` (tools/config/tsconfig-paths.ts:L15-L15, package.json:L63-L63).

---

10. Come fare debug

- **Un singolo spec Vitest.** `npx vitest run --config <lib>/vitest.config.ts <path-spec>` (.claude/CLAUDE.md:L37-L41), oppure `nx test-vitest <lib>`, che esegue lo stesso comando con cwd sulla lib (core-libs/core/project.json:L20-L26). La coverage è sempre attiva (core-libs/core/vitest.config.ts:L26-L26), quindi su un singolo file le soglie 85/85/65/85 possono fallire anche se i test passano. Le istruzioni di progetto suggeriscono anche `--sequence.shuffle` per scoprire dipendenze dall'ordine e `--detect-async-leaks` per gli async non risolti (.claude/CLAUDE.md:L44-L46).
- **Import non risolto sotto Vitest.** Controlla nell'ordine gli alias di `resolve.alias` (deep path prima di quelli generici: core-libs/storefront/vitest.config.ts:L16-L19) e poi i `paths` del `tsconfig.spec.json` della lib (feature-libs/cart/tsconfig.spec.json:L35-L40). Storefront mappa `@spartacus/core` sul sorgente (core-libs/storefront/vitest.config.ts:L25-L28); la motivazione non è dichiarata nel codice, ma verosimilmente serve a evitare due copie dello stesso token DI (una da `dist`, una dal sorgente), causa tipica di mock "ignorati".
- **NG0100 sotto Karma.** Verifica che il `test.ts` registri `ZoneChangeDetectionModule` come in integration-libs/opf/test.ts:L9-L32. Sotto Vitest non serve (core-libs/core/src/test.ts:L26-L26).
- **Quale shard ha eseguito un progetto.** Ogni gruppo stampa nel log CI l'elenco dei progetti preceduto da `[KARMA]/[JEST]/[VITEST] Group will run N project(s):` (ci-scripts/unit-tests.sh:L18-L28). In locale `ci-scripts/compute-unit-shards.sh` stampa la matrice JSON (ci-scripts/compute-unit-shards.sh:L53-L63). Servono `jq` e nx funzionante.

- **E2E SSR che non parte.** Leggi il messaggio del `globalSetup`: nomina la condizione violata e suggerisce il comando di build o quello per liberare la porta (projects/ssr-tests/validate-ssr-build.ts:L63-L81, projects/ssr-tests/validate-ssr-build.ts:L83-L141). Se il test va in timeout, apri `projects/ssr-tests/.ssr.log`, dove finiscono stdout e stderr del server (projects/ssr-tests/src/utils/ssr.utils.ts:L56-L56). Su asserzione fallita, i log sono già nella `cause` stampata da Jest, tra `--- SSR LOGS (with JSONs pretty-printed) ---` e `--- SSR LOGS END ---` (projects/ssr-tests/src/environments/custom-test-environment.ts:L58-L60). Se resta un processo orfano sulla 4000: `kill -9 $(lsof -t -i :4000)`, come suggerito da `KILL_PORT_ADVICE` (projects/ssr-tests/validate-ssr-build.ts:L70-L81).
- **Backend CCv2 dietro il proxy.** Imposta a mano `IS_CCV2_SERVER = true` per usare un `HttpsAgent` con SNI (projects/ssr-tests/src/utils/proxy.utils.ts:L13-L17, projects/ssr-tests/src/utils/proxy.utils.ts:L66-L71).
- **Config dinamica negli E2E.** In SSR passa `testConfig` (projects/ssr-tests/src/utils/http.utils.ts:L27-L37), in Cypress `cy.cxConfig(...)` (projects/storefrontapp-e2e-cypress/cypress/support/cx-config.commands.ts:L26-L28). Funzionano solo con un'app che importa `TestConfigModule.forRoot({ cookie: 'cxConfigE2E' })`, come quella demo (projects/storefrontapp/src/app/private/private.providers.ts:L54-L54).
- **Test SCSS.** Lancia sempre da `core-libs/styles` (`nx run storefrontstyles:test-jest`, core-libs/styles/project.json:L6-L12). Se il runner dice che non ci sono file, il problema è la cwd (trappola 15).
- **Lint rosso in CI ma non in locale.** Esegui `ci-scripts/validate-lint.sh` per intero: i grep anti-focus e il check su `tsconfig.app.prod.json` non fanno parte di `npm run lint` (ci-scripts/validate-lint.sh:L28-L80, package.json:L128-L131).
- **Dipendenze o tsconfig incoerenti.** `npm run config:check` riporta errori e warning; `npm run config:update` prova a correggerli, rigenera il lock file e formatta (package.json:L62-L64, tools/config/index.ts:L349-L371).

---

11. Domande di autoverifica

1. **Con quale runner si testano allo SHA `core` e `storefrontlib`, e con quale comando?** Con Vitest, tramite il target `test-vitest` → `npx vitest run --config vitest.config.ts` nella root della lib. Non hanno un target `test`. — core-libs/core/project.json:L20-L26, core-libs/storefront/project.json:L21-L27
2. **Perché una lib come `cart` ha due target di test?** `test-vitest` esegue gli unit test, `test-jest` esegue gli schematics (`npm run test:schematics`) con Jest. — feature-libs/cart/project.json:L20-L32, core-libs/schematics/jest.schematics.config.js:L1-L23
3. **Come decide la CI quanti e quali progetti mettere in ogni shard?** Enumera i progetti per target con `nx show projects --with-target`, pesa ognuno con le righe significative dei suoi `*.spec.ts` e li distribuisce con un greedy LPT sul bucket meno carico. — ci-scripts/compute-unit-shards.sh:L19-L51, ci-scripts/shard-lib.sh:L30-L71
4. **Perché il `test.ts` Karma registra `provideZoneChangeDetection()` e `setup-vitest.ts` no?** Con un `test.ts` custom Angular 21 produce errori NG0100 sotto Karma; il commento annota che sotto Vitest non serve più. — core-libs/core/src/test.ts:L21-L30, testing/setup-vitest.ts:L16-L19
5. **In un test di storefront, perché l'alias `@spartacus/storefront/testing/mock-feature-directive` deve stare prima di `@spartacus/storefront`?** Vite usa il primo alias che corrisponde, e quello "nudo" catturerebbe anche il deep path. — core-libs/storefront/vitest.config.ts:L16-L19
6. **Come fanno gli E2E SSR a simulare un errore del backend OCC senza modificare il backend?** L'app è buildata con `CX_BASE_URL=http://localhost:9002`. Su quella porta il test avvia un `http-proxy` verso il bac-

kend reale, il cui `responseInterceptor` può cambiare status code o ritardare la risposta. — `.env-cmdrc:L11-L13`, `projects/ssr-tests/src/utils/proxy.utils.ts:L58-L115`, `projects/ssr-tests/src/ssr-testing.spec.ts:L52-L82`

7. **Come si cambia la configurazione Spartacus di un'app SSR già buildata, per un singolo test?** Passando `testConfig` a `sendRequestToSsrServer`: diventa il cookie `cxConfigE2E`, che `TestConfigModule` dell'app inietta nella config globale. — `projects/ssr-tests/src/utils/http.utils.ts:L27-L37`, `projects/ssr-tests/src/utils/http.utils.ts:L75-L89`, `projects/storefrontapp/src/app/private/private.providers.ts:L54-L54`
8. **Dove si trovano i log del server SSR quando un test fallisce?** Nella `cause` dell'errore stampato da Jest, aggiunta da `CustomTestEnvironment` su `test_done`, e nel file `.ssr.log`. — `projects/ssr-tests/src/environments/custom-test-environment.ts:L19-L43`, `projects/ssr-tests/src/utils/log.utils.ts:L18-L18`
9. **Che cosa blocca `validate-lint.sh` che `npm run lint` non controlla?** I focus test (`fdescribe(`, `describe.only(`, `fit(`, `it.only()`), il self-import di storefrontlib e un `tsconfig.app.prod.json` che punta ai sorgenti invece che a `dist`. — `ci-scripts/validate-lint.sh:L11-L80`
10. **Come si testa un mixin SCSS in questo repo?** Con un file `test/*.spec.scss` che usa i mixin di sass-true (`describe`, `it`, `assert`, `output`, `expect`). `scss.spec.js` lo scopre con `globSync` e lo esegue sotto Jest con `sassTrue.runSass`. — `core-libs/styles/test/features.spec.scss:L14-L35`, `core-libs/styles/test/scss.spec.js:L1-L15`
11. **Perché `MockWinRef` non si può usare in `integration-libs/opf`?** Importa `vi` da `vitest`, mentre `opf` gira ancora su Karma/Jasmine. — `core-libs/storefront/shared/test/mock-window-ref.ts:L7-L7`, `integration-libs/opf/project.json:L20-L28`
12. **Una regola `@nx/enforce-module-boundaries` impedisce a `core` di importare da `storefrontlib`: su quali tag si basa?** `core` ha `scope:core`, che non può dipendere da `type:ui`/`type:feature`; `storefrontlib` ha `type:ui`. — `eslint.config.mjs:L148-L151`, `core-libs/core/project.json:L37-L37`, `core-libs/storefront/project.json:L38-L38`

TAVOLA 44.1 **4.1 Unit test in CI: dalla matrice al runner**

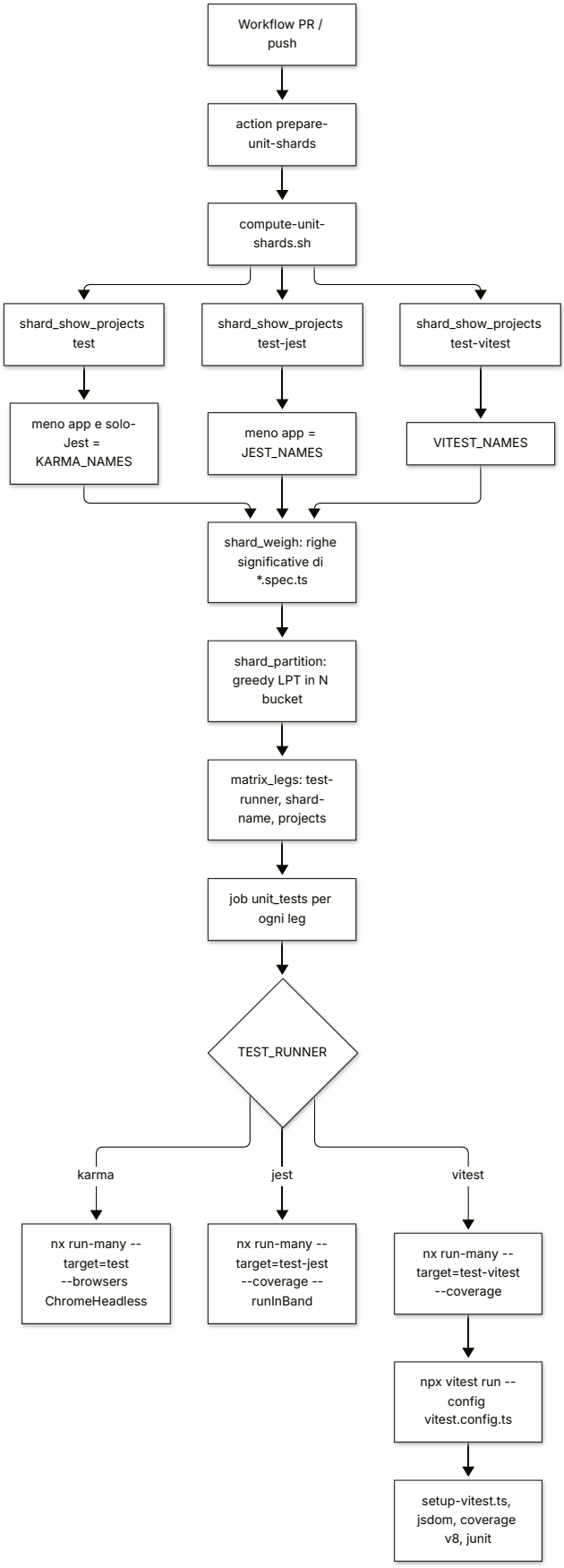
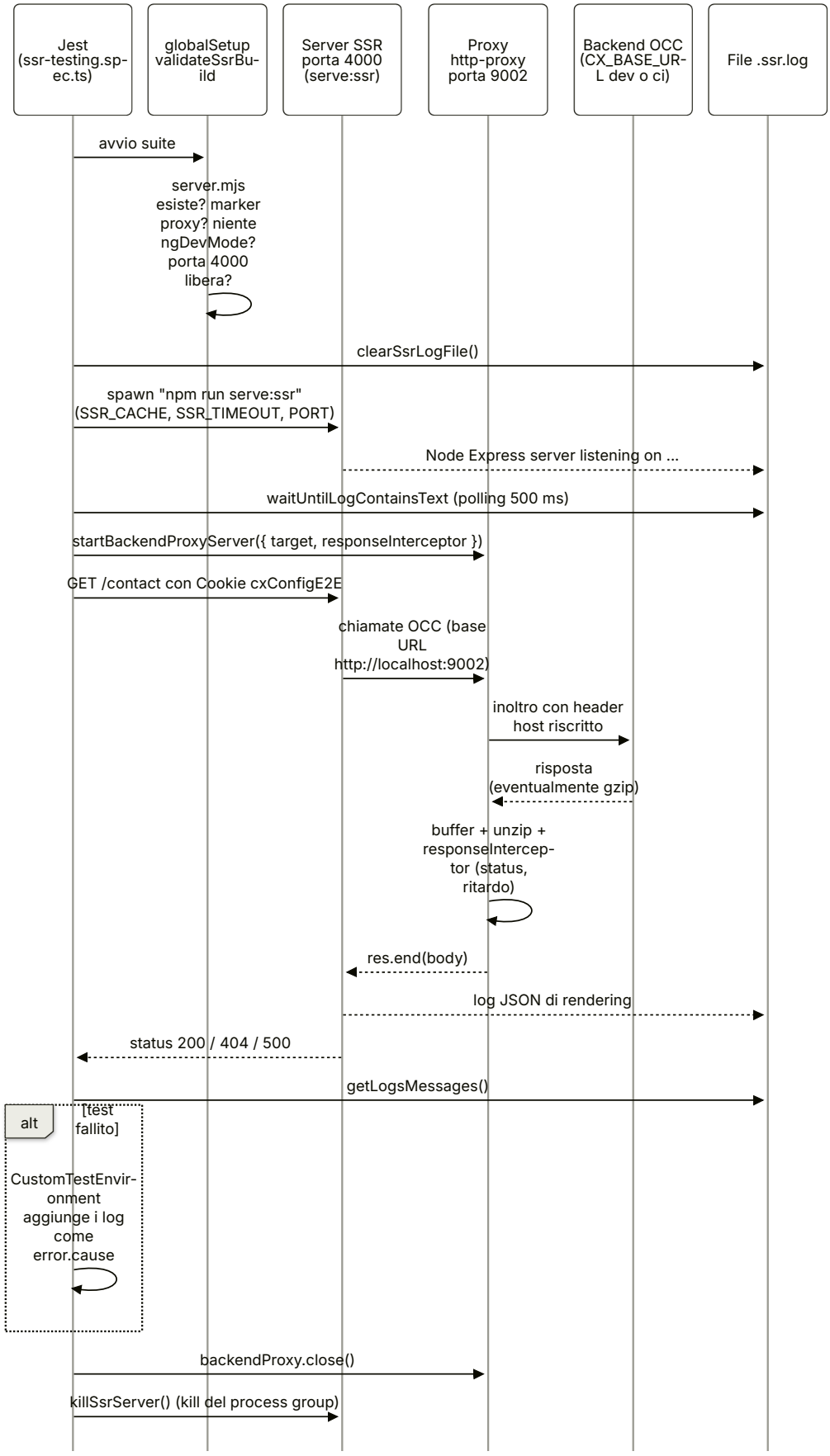


TAVOLA 44.2 **4.2 Un test E2E SSR con proxy e interceptor**



45

Componenti CMS e shared components di storefront

Componenti per i tipi CMS di base (dati, normalizzazione, link, a11y) e building block `shared/`.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

I capitoli 15 e 16 spiegano come una pagina CMS arriva nello store e come `cxComponentWrapper` trasforma un `typeCode` in un componente Angular. Questo capitolo guarda *dentro* i componenti: quelli CMS "generici" di `@spartacus/storefront` (content, navigation, misc) e la cassetta degli attrezzi condivisa (`shared/`: media, carousel, generic link, form errors, item counter, pagination, popover, direttive, pipe, servizi file e validator) che tutte le feature lib riusano.

Lo scopo non è un catalogo esaustivo dei 286 file assegnati: si sceglie un componente rappresentativo per famiglia (Banner per content, NavigationService + NavigationUIComponent per navigation, Icon per misc, Media/Carousel/GenericLink per shared) e lo si segue fino alle righe di codice, mentre gli altri sono descritti per differenza. La tabella completa `typeCode → componente → modulo` di tutto il monorepo è nell'appendice A03.

---

1. A cosa serve e perché esiste

Il backend SAP Commerce restituisce componenti CMS con un `typeCode` (es. `SimpleResponsiveBannerComponent`, `CMSParagraphComponent`, `NavigationComponent`) e un sacco di proprietà quasi sempre **stringhe**. Lo storefront deve fornire, per ogni tipo "di base" previsto dal modello CMS standard, un componente Angular che:

1. riceva i dati tramite `CmsComponentData` (creato per ogni istanza da `CmsInjectorService.getInjector`, core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50);
2. **normalizzi** i dati CMS, che arrivano spesso con booleani come stringa: `data.external === 'true' || data.external === true` nel banner (core-libs/storefront/cms-components/content/banner/banner.component.ts:L133-L135), `data.autoPlay === 'true'` nel video (core-libs/storefront/cms-components/content/video/video.component.ts:L92-L96), `isBool(...)` nel search box (core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L246-L266);
3. **risolva i link** CMS (URL assoluto, content page, prodotto, categoria) in qualcosa che il router Angular capisca, con la cascata `urlLink → contentPage → product → category` (core-libs/storefront/cms-components/content/banner/banner.component.ts:L137-L161);
4. renda il tutto accessibile (tastiera, ARIA) e compatibile SSR.

Accanto ai componenti CMS esiste la directory `shared/`: componenti *non* mappati su typeCode ma usati come building block da tutto il resto (`cx-media`, `cx-generic-link`, `cx-carousel`, `cx-item-counter`, `cx-form-errors`, `cx-pagination`, `[cxPopover]`, ...). `cms-components/index.ts` e `shared/index.ts` sono entrambi riesportati da `@spartacus/storefront` (core-libs/storefront/public_api.ts:L21-L21, core-libs/storefront/public_api.ts:L29-L29), quindi tutto ciò che vedete in questo capitolo è API pubblica salvo le eccezioni elencate nella sezione 9.

Il pattern architetturale ricorrente è: **componente standalone + NgModule "wrapper"** che importa il componente e registra il mapping con `provideDefaultConfig(<CmsConfig>{ cmsComponents: {...} })` (core-libs/storefront/cms-components/content/banner/banner.module.ts:L20-L47). Il modulo resta per retrocompatibilità e per portare con sé la configurazione di default; l'app demo li importa esplicitamente in `SpartacusFeaturesModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L184-L199).

2. Mappa dei file

Prefisso comune: `core-libs/storefront/`. I file sono raggruppati per cartella; dove una cartella contiene il trittico `.component.ts/.html/.module.ts` si indica solo la cartella.

2.1 `cms-components/` (content, navigation, misc)

PATH	RUOLO
`core-libs/storefront/cms-components/index.ts`	Barrel di tutti i gruppi CMS (anche myaccount/product/user, fuori capitolo)
`core-libs/storefront/cms-components/content/banner/`	`BannerComponent` per 3 typeCode, link risolto, LCP, frecce a11y
`core-libs/storefront/cms-components/content/banner-carousel/`	`BannerCarouselComponent` per `RotatingImagesComponent`: carousel di componenti CMS
`core-libs/storefront/cms-components/content/link/`	`LinkComponent` per `CMSLinkComponent` → `cx-generic-link`
`core-libs/storefront/cms-components/content/paragraph/`	`ParagraphComponent` per `CMSParagraphComponent`/`CMSTabParagraphComponent`: HTML trusted + click via router
`core-libs/storefront/cms-components/content/pdf/`	`PDFComponent` per `PDFDocumentComponent`
`core-libs/storefront/cms-components/content/video/`	`VideoComponent` CMS (sorgente, thumbnail, flag, link)
`core-libs/storefront/cms-components/content/tab/` (`tab.component.ts`, `tab.model.ts`, `tab.module.ts`, `tab.utils.ts`, `panel/`)	`cx-tab`: tab/accordion generico, **non** mappato su typeCode
`core-libs/storefront/cms-components/content/tab-paragraph-container/`	`TabParagraphContainerComponent` per `CMSTabParagraphContainer`
`core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts`	`NavigationService`: costruzione dell'albero `NavigationNode` dallo store CMS
`core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts` (+ `.html`)	`NavigationUIComponent` (`cx-navigation-ui`): flyout, hamburger, tastiera
`core-libs/storefront/cms-components/navigation/navigation/navigation-node.model.ts`	Modello di vista `NavigationNode`
`core-libs/storefront/cms-components/navigation/navigation/navigation.component.ts`	`NavigationComponent` CMS (usa `createNavigation`)
`core-libs/storefront/cms-components/navigation/category-navigation/`	`CategoryNavigationComponent`, unico mapping con `data` statici
`core-libs/storefront/cms-components/navigation/footer-navigation/`	`FooterNavigationComponent`, senza flyout
`core-libs/storefront/cms-components/navigation/breadcrumb/`	`BreadcrumbComponent` da `PageMetaService`
`core-libs/storefront/cms-components/navigation/page-header/` (+ `page-title.utils.ts`)	`PageTitleComponent` e util `getPageTitle`
`core-libs/storefront/cms-components/navigation/scroll-to-top/`	`ScrollToTopComponent`

PATH	RUOLO
core-libs/storefront/cms-components/navigation/search-box/	SearchBoxComponent, SearchBoxComponentService, eventi, outlet, HighlightPipe
core-libs/storefront/cms-components/misc/icon/	IconComponent, IconLoaderService, IconConfig, ICON_TYPE, config Font Awesome, IconTestingModule
core-libs/storefront/cms-components/misc/site-context-selector/	SiteContextSelectorComponent, LanguageCurrencyComponent, SiteContextComponentService
core-libs/storefront/cms-components/misc/site-theme-switcher/	SiteThemeSwitcherComponent + service
core-libs/storefront/cms-components/misc/global-message/	GlobalMessageComponent (nel layout, non CMS)
core-libs/storefront/cms-components/misc/message/	MessageComponent inline riutilizzabile
core-libs/storefront/cms-components/misc/promotions/	PromotionsComponent presentational

2.2 shared/

PATH	RUOLO
core-libs/storefront/shared/components/media/ (media.component.ts, media.service.ts, media.config.ts, media.model.ts, media.module.ts, media-sources.pipe.ts)	cx-media: immagini responsive img/picture, URL assoluti, lazy/eager, fetch priority
core-libs/storefront/shared/components/carousel/ (+ carousel.service.ts, focusable-carousel-item/)	cx-carousel a slide
core-libs/storefront/shared/components/carousel-scrolling/	carousel a scroll orizzontale
core-libs/storefront/shared/components/generic-link/ (+ generic-link-component.service.ts)	cx-generic-link: link interno (routerLink) o esterno (href)
core-libs/storefront/shared/components/item-counter/	cx-item-counter su FormControl esterno
core-libs/storefront/shared/components/form/ (form-errors, date-picker, file-upload, password-visibility-toggle, required asterisks/legend)	componenti form
core-libs/storefront/shared/components/list-navigation/ (pagination, sorting, total)	paginazione (PaginationBuilder), ordinamento ng-select, totale
core-libs/storefront/shared/components/popover/	[cxPopover], PopoverComponent, PopoverService, modello
core-libs/storefront/shared/services/positioning/positioning.service.ts	PositioningService: calcolo posizione popover
core-libs/storefront/shared/components/card/, spinner/, star-rating/, progress-button/, read-more/, truncate-text-popover/	presentational vari
core-libs/storefront/shared/components/table/, split-view/, chat-messaging/, captcha/, anonymous-consents-dialog/, assistive-technology-message/, ng-select-a11y/	componenti più complessi (mappati, non approfonditi qui)
core-libs/storefront/shared/directives/ (roving-tabindex/, horizontal-scrolling-position/)	direttive a11y e scroll
core-libs/storefront/shared/pipes/suplement-hash-anchors/	cxSupplementHashAnchors
core-libs/storefront/shared/services/file/	download file, CSV import/export, validator file

PATH	RUOLO
core-libs/storefront/shared/utils/ (validators/custom-form-validators.ts, forms/title-utils.ts)	validator e util form
core-libs/storefront/shared/config/ (form-config.ts, view-config.ts, default-form-config.ts)	config FormConfig / ViewConfig

3. Come funziona passo per passo

3.1 Il contratto comune: mapping, injector, data$

1. **Registrazione.** Ogni modulo wrapper registra uno o più typeCode. BannerModule mappa tre typeCode sullo stesso BannerComponent (core-libs/storefront/cms-components/content/banner/banner.module.ts:L20-L47); CmsParagraphModule mappa CMSParagraphComponent e CMSTabParagraphComponent su ParagraphComponent (core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L22-L31); LinkModule mappa CMSLinkComponent (core-libs/storefront/cms-components/content/link/link.module.ts:L17-L21); BannerCarouselModule mappa RotatingImagesComponent (core-libs/storefront/cms-components/content/banner-carousel/banner-carousel.module.ts:L23-L29); PDF, Video e TabParagraphContainer seguono lo stesso schema (core-libs/storefront/cms-components/content/pdf/pdf.module.ts:L16-L22, core-libs/storefront/cms-components/content/video/video.module.ts:L27-L33, core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.module.ts:L31-L37). I moduli navigation mappano 1:1 typeCode omonimi (core-libs/storefront/cms-components/navigation/navigation/navigation.module.ts:L33-L39, core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.module.ts:L29-L35, core-libs/storefront/cms-components/navigation/search-box/search-box.module.ts:L41-L47).
2. **Moduli senza mapping.** TabModule non registra nulla: cx-tab è un building block (core-libs/storefront/cms-components/content/tab/tab.module.ts:L13-L17). Anche GlobalMessageComponentModule, MessageComponentModule e PromotionsModule non hanno mapping; <cx-global-message> è scritto direttamente nel template di StorefrontComponent (core-libs/storefront/cms-components/misc/global-message/global-message.module.ts:L13-L17, core-libs/storefront/layout/main/storefront.component.html:L16-L16) e il modulo è importato da BaseStorefrontModule (core-libs/storefront/base-storefront.module.ts:L30-L30).
3. **Injector per istanza.** Quando il wrapper crea un componente, CmsInjectorService.getInjector(type, uid) costruisce un Injector con un provider CmsComponentData ({ uid, data$ }), provideLcpPresenceForCmsComponent() e i providers del mapping (core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50). CmsComponentData è il tipo con uid e data$ (core-libs/storefront/cms-structure/page/model/cms-component-data.ts:L10-L13).
4. **Dati statici + dati backend.** data$ è ComponentDataProvider.get(uid, type): se il mapping ha data, emette subito i dati statici (startWith) e poi {...static, ...backend}, quindi il backend sovrascrive (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L35-L59). Nel perimetro di questo capitolo solo CategoryNavigationComponent usa data (resetMenuOnClose: true, core-libs/storefront/cms-components/navigation/category-navigation/category-navigation.module.ts:L21-L30).
5. **Providers per componente.** SiteContextSelectorModule usa il campo providers del mapping per dare a ogni istanza CMS un proprio SiteContextComponentService costruito con CmsComponentData nelle deps, e fornisce *anche* il service a livello modulo (senza dati CMS) per l'uso standalone (core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.module.ts:L33-L52).

3.2 Content: Banner (rappresentativo)

1. `BannerComponent` è OnPush; `data$` deriva da `component.data$` con un `tap` che chiama `setRouterLink(data)` e copia `data.styleClasses` nella `@HostBinding('class')` (core-libs/storefront/cms-components/content/banner/banner.component.ts:L44-L57).
2. `setRouterLink` risolve in cascata (core-libs/storefront/cms-components/content/banner/banner.component.ts:L137-L161):
   - `urlLink` → usato così com'è;
   - `contentPage` → `cmsService.getPage({ id, type: PageType.CONTENT_PAGE }).pipe(take(1))` e poi `page?.label` (asincrono);
   - `product` → `urlService.transform({ cxRoute: 'product', params: { code } })`;
   - `category` → idem con route `category`.
3. `getTarget` restituisce `_blank` se `data.external` vale `'true'` o `true` (core-libs/storefront/cms-components/content/banner/banner.component.ts:L133-L135).
4. `getImage` distingue `Image` da `ImageGroup` con `'url' in data.media`; l'aria-label del link è `headline ?? altText` della prima immagine (core-libs/storefront/cms-components/content/banner/banner.component.ts:L163-L189).
5. Il template, dentro `*cxLcpContext`, ha due rami: con `routerLink` rende un `cx-generic-link` **vuoto** seguito da headline/media/content come fratelli; senza link rende un `div.no-link`. Headline e content usano `[innerHTML]`; `cx-media` riceve `elementType 'picture'` e `fetchPriority` dal contesto LCP (core-libs/storefront/cms-components/content/banner/banner.component.html:L1-L35). Il contesto LCP è trattato nel capitolo 16.
6. Con il toggle `a11yBannerArrowKeyNavigation` (default `false`, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1114-L1114) il banner gestisce le frecce: cerca i sibling `cx-banner` nel parent e per ↑/↓ usa come passo il numero di colonne della prima riga, calcolato confrontando `getBoundingClientRect().top` con tolleranza di 1px (core-libs/storefront/cms-components/content/banner/banner.component.ts:L65-L127).

3.3 Content: gli altri per differenza

- **Paragraph.** Il contenuto passa prima dalla pipe `cxSupplementHashAnchors` e poi da `DomSanitizer.bypassSecurityTrustHtml`: nessuna sanitizzazione Angular (core-libs/storefront/cms-components/content/paragraph/paragraph.component.html:L1-L4, core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L49-L51). Un `@HostListener('click')` intercetta i click su `<a>`: se l'host coincide con quello del documento fa `preventDefault` e `router.navigateByUrl` (navigazione SPA) (core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L29-L42). La pipe riscrive gli `href="#id"` prefissandoli con l'URL corrente senza fragment, usando un `<template>` creato via `Renderer2` (core-libs/storefront/shared/pipes/suplement-hash-anchors/supplement-hash-anchors.pipe.ts:L10-L50).
- **Link.** `LinkComponent` si sottoscrive a mano in `ngOnInit` per `styleClasses` e si disiscrive in `ngOnDestroy`; il template delega a `cx-generic-link` con `url`, `style` e `target` (core-libs/storefront/cms-components/content/link/link.component.ts:L35-L53, core-libs/storefront/cms-components/content/link/link.component.html:L1-L7).
- **BannerCarousel.** `data.banners` è una stringa di uid separati da spazi: viene splittata e ogni uid diventa un Observable `cmsService.getComponentData(code)`; la classe host è `${effect}-theme` (core-libs/storefront/cms-components/content/banner-carousel/banner-carousel.component.ts:L30-L47). Ogni item è reso con `[cxComponentWrapper]` dentro `cx-carousel` con `itemWidth="100%"` (core-libs/storefront/cms-

components/content/banner-carousel/banner-carousel.component.html:L1-L17): un componente CMS che istanzia altri componenti CMS.

- **PDF.** `url = MediaService.getMedia(data.pdfFile)?.src`; `addPdfExtension` aggiunge `.pdf` al titolo se manca (core-libs/storefront/cms-components/content/pdf/pdf.component.ts:L22-L43).
- **Video.** Flag stringa convertiti (`autoPlay`/`loop` → `=== 'true'`, `mute` → `'muted'|undefined`) (core-libs/storefront/cms-components/content/video/video.component.ts:L92-L96); thumbnail scelto in base a `containerBackground`/`thumbnailSelector` (core-libs/storefront/cms-components/content/video/video.component.ts:L67-L90); la cascata di routing replica quella del banner ma chiama `cd.markForCheck()` dopo la risoluzione asincrona della content page (core-libs/storefront/cms-components/content/video/video.component.ts:L98-L123).

3.4 Content: TabParagraphContainer e `cx-tab`

1. Config di default `{ openTabs: [0], breakpoint: BREAKPOINT.md }` (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L39-L42).
2. `components$` splitta `data.components`, carica ogni figlio con `cmsService.getComponentData`, usa `typeCode` come `flexType` se mancante, imposta `title = ${containerUid}.tabs.${tabUid}` (chiave i18n), scarta i nulli ed emette su `tabConfig$` la label `${uid}.tabPanelContainerRegionGroup` (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L99-L140).
3. Il template crea un `ng-template #tabRef` per figlio con `[cxOutlet]="component?.flexType" + [cxComponentWrapper]` e ascolta `(cxComponentRef)` (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.html:L1-L15).
4. In `ngAfterViewInit` `tabs$` combina `components$`, `tabRefs.changes` e `tabTitleParams$` per produrre i `Tab` (`headerKey`, `content` = TemplateRef, `id`, `headerParams`), poi `cdr.detectChanges()` (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L169-L200).
5. Quando un figlio viene creato, `tabCompLoaded` registra il suo `tabTitleParam$` una sola volta per uid, per evitare il loop `tabs$ → load → tabTitleParams$ → tabs$` (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L202-L213).
6. `cx-tab` decide la modalità: `config.mode` se presente, altrimenti `TAB` sopra il breakpoint e `ACCORDIAN` sotto (`BreakpointService.isUp`), altrimenti `TAB` (core-libs/storefront/cms-components/content/tab/tab.component.ts:L254-L268). In `TAB` `select()` apre solo il tab scelto, in `ACCORDIAN` fa toggle (core-libs/storefront/cms-components/content/tab/tab.component.ts:L96-L105, core-libs/storefront/cms-components/content/tab/tab.component.ts:L196-L205).
7. Tastiera: frecce, Home ed End spostano solo il focus con wrap circolare (`wrapIntoBounds`); `restrictDirectionKeys` limita Tab a ←/→ e Accordion a ↑/↓ (core-libs/storefront/cms-components/content/tab/tab.component.ts:L136-L182, core-libs/storefront/cms-components/content/tab/tab.utils.ts:L12-L20). ARIA: `role="tab"` in modalità TAB, `role="button"` + `aria-expanded` in ACCORDIAN (core-libs/storefront/cms-components/content/tab/tab.component.html:L4-L17).

3.5 Navigation: dall'entry CMS al menu

1. `NavigationComponent.node$ = service.createNavigation(componentData.data$)` (core-libs/storefront/cms-components/navigation/navigation/navigation.component.ts:L23-L40). `createNavigation` incapsula il nodo radice in `{ title: data.name, children: [nav] }` (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L27-L38). `CategoryNavigationComponent` e `FooterNavigationComponent` usano invece `getNavigationNode` direttamente, **senza** nodo wrapper (core-libs/storefront/cms-components/navigation/category-navigation/category-navigation.component.ts:L22-L33, core-libs/storefront/cms-components/navigation/footer-navigation/footer-navigation.component.ts:L23-L26).

2. `getNavigationNode` prende `data.navigationNode ?? data` e legge gli entry item dallo store con `cmsService.getNavigationEntryItems(uid)` (un `select` NgRx) (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L45-L57, core-libs/core/src/cms/facade/cms.service.ts:L171-L192).
3. Se gli item sono `undefined`, `loadNavigationEntryItems(navigation, true)` raccoglie ricorsivamente `{superType, id}` di entries e children e dispatcha **solo per la radice** (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L95-L118); `CmsService.loadNavigationItems` dispatcha `CmsActions.LoadCmsNavigationItems` (core-libs/core/src/cms/facade/cms.service.ts:L171-L192).
4. Se gli item ci sono ma ne mancano alcuni, carica solo i mancanti (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L63-L80).
5. `filter(Boolean)` e poi `populateNavigationNode` ricorsivo: titolo, `styleClasses`, `styleAttributes`, link dalla **prima** entry, figli filtrati; un nodo senza proprietà diventa `null` e viene scartato (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L125-L161).
6. `populateLink` considera solo `itemType === 'CMSLinkComponent'`, cerca l'item con chiave `${itemId}_${itemSuperType}` e imposta `url` solo se il nodo ha un titolo; `target='_blank'` se `item.target` è `'true'`/`true` (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L166-L193). `getLink` (protected) sceglie `url` → `contentPageLabelOrId` → route `category` → route `product` (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L201-L223).
7. `cx-navigation-ui` rende l'albero. Input: `node`, `wrapAfter`, `resetMenuOnClose`, `focusableNodeTitles`, `navAriaLabel`, `flyout` (default `true`, classe host `flyout`), `isOpen` (classe host `is-open`) (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L57-L88). La category navigation passa `wrapAfter` convertito con `+`, `resetMenuOnClose` e l'aria-label `navigation.categoryNavLabel` (core-libs/storefront/cms-components/navigation/category-navigation/category-navigation.component.html:L1-L8); il footer disattiva il flyout (core-libs/storefront/cms-components/navigation/footer-navigation/footer-navigation.component.html:L1-L7).
8. Apertura sottomenu: `toggleOpen` gestisce click e tastiera; con `a11yNavigationSpaceKeyOnKeyUp` ignora `keydown` e reagisce su `keyup`, altrimenti il contrario (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L197-L229, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1093-L1093). Lo stato vive in `openNodes: HTMLElement[]` ed è riflesso con `Renderer2` come `is-open` (ultimo) e `is-opened` (antenati) (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L381-L393).
9. A ogni `NavigationEnd` il costruttore chiama `clear()`, che azzera `openNodes` e la classe host `is-open` (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L118-L122, core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L303-L306, core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L381-L393): le classi `is-open`/`is-opened` già applicate ai `<li>` e gli `aria-expanded` restano, li rimuove solo `reinitializeMenu()` (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L166-L185). Su `window:resize` con `debounceTime(50)` riallinea i `.wrapper` che sbordano a destra (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L95-L128, core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L356-L372).
10. Con `resetMenuOnClose` il componente si sottoscrive a `HamburgerMenuService.isExpanded` e reinizializza il menu quando l'hamburger si apre (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L133-L185). Il numero di colonne di un sottomenu è 1 senza `wrapAfter` (o con `length <= 0`); altrimenti `floor(length / wrapAfter)`, più 1 se c'è resto e il quoziente è almeno 1, quindi con meno figli di `wrapAfter` il risultato è 0, non 1 (core-libs/storefront/cms-components/

navigation/navigation/navigation-ui.component.ts:L323-L335); getTabIndex dà -1 alle foglie in profondità > 0 solo con flyout (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L398-L403).

3.6 Navigation: breadcrumb, titolo, scroll-to-top, search box

- **Breadcrumb** estende PageTitleComponent; le crumb vengono da pageMetaService.getMeta().breadcrumbs, con fallback [{ label: 'common.home' tradotto, link: '/' }] (core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.component.ts:L32-L66). ariaLive$ emette a ogni NavigationEnd il booleano document.activeElement !== document.body (core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.component.ts:L37-L42). I meta di pagina sono descritti nel capitolo 40.
- **Page title**: getPageTitle restituisce meta.heading || meta.title (core-libs/storefront/cms-components/navigation/page-header/page-title.utils.ts:L11-L18, core-libs/storefront/cms-components/navigation/page-header/page-title.component.ts:L25-L45).
- **Scroll-to-top**: scrollBehavior e displayThreshold dai dati CMS con take(1); default SMOOTH e metà di innerHeight (400/2 se assente) (core-libs/storefront/cms-components/navigation/scroll-to-top/scroll-to-top.component.ts:L40-L42, core-libs/storefront/cms-components/navigation/scroll-to-top/scroll-to-top.component.ts:L102-L107). Visibile se scrollY > displayThreshold o se attivato da tastiera e ancora focalizzato (core-libs/storefront/cms-components/navigation/scroll-to-top/scroll-to-top.component.ts:L66-L123).
- **Search box**: config di default (minCharactersBeforeRequest: 1, max 5 prodotti/suggerimenti, recent e trending attivi) (core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L61-L72) fusa a tre livelli default → dati CMS → @Input() config, con conversione dei flag stringa 'false' e riscrittura del risultato in this.config (core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L246-L266). SearchBoxComponentService.search non chiama il backend sotto minCharactersBeforeRequest e usa searchWithCompletion/searchSuggestionsWithCompletion con pageSize = maxProducts/maxSuggestions (core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L70-L125); getResults toggla sul body la classe has-searchbox-results (core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L149-L166). Eventi SearchBoxSuggestionSelectedEvent/SearchBoxProductSelectedEvent via EventService (core-libs/storefront/cms-components/navigation/search-box/search-box.events.ts:L12-L32, core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L204-L230) e tre outlet SearchBoxOutlets (core-libs/storefront/cms-components/navigation/search-box/search-box-outlets.model.ts:L7-L11). La ricerca prodotti lato core è nel capitolo 24. DA VERIFICARE: la logica interna di recent/trending searches e della gestione focus a gruppi di SearchBoxComponent (951 righe) è stata solo mappata dall'analista, non letta per intero.

3.7 Misc: Icon (rappresentativo), site context, theme, messaggi

1. IconComponent ha selettore doppio cx-icon,[cxIcon]: come attributo non aggiunge un nodo DOM e il contenuto è proiettato (core-libs/storefront/cms-components/misc/icon/icon.component.ts:L22-L46).
2. I setter cxIcon e type chiamano setIcon, che ottiene l'HTML dal loader, aggiunge le classi, carica l'eventuale risorsa LINK e imposta il flip (core-libs/storefront/cms-components/misc/icon/icon.component.ts:L52-L99). addStyleClasses aggiunge sempre cx-icon, rimuove le classi del tipo precedente e aggiunge quelle del simbolo split per spazio (core-libs/storefront/cms-components/misc/icon/icon.component.ts:L113-L126).
3. IconLoaderService.getHtml: per SVG costruisce <svg><use xlink:href> con URL sanitizzato (SecurityContext.URL); per TEXT mette il simbolo in textContent; per i font (LINK) restituisce undefined e bastano le classi (core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L33-L55). Il

template rende `<i [outerHTML]="icon">` solo se c'è HTML e proietta `<ng-content />` (core-libs/storefront/cms-components/misc/icon/icon.component.html:L1-L4).

4. `addLinkResource` inietta una sola volta per URL un `<link rel="stylesheet">` nel `<head>`, cercando una risorsa specifica per tipo o una "one-size-fits-all" senza `types` (core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L118-L160).
5. Configurazione: `IconConfig` è una classe astratta `useExisting: Config` con `IconOptions` (`symbols`, `resources`, `flipDirection`) (core-libs/storefront/cms-components/misc/icon/icon.model.ts:L79-L121); `IconResourceType` vale `SVG`, `LINK`, `TEXT` (core-libs/storefront/cms-components/misc/icon/icon.model.ts:L133-L149). `IconModule` fornisce `defaultIconConfig` (flip RTL dei caret) e `fontawesomeIconConfig` (core-libs/storefront/cms-components/misc/icon/icon.module.ts:L14-L23, core-libs/storefront/cms-components/misc/icon/default-icon.config.ts:L10-L17); quest'ultimo mappa ogni `ICON_TYPE` su classi Font Awesome con una risorsa LINK a `url: ''`, quindi nessun CSS viene caricato dinamicamente (core-libs/storefront/cms-components/misc/icon/fontawesome-icon.config.ts:L9-L86). `ICON_TYPE` è un enum augmentabile riesportato nel blocco delimitato dal commento `AUGMENTABLE_TYPES_START` (core-libs/storefront/cms-components/misc/icon/icon.model.ts:L11-L77, core-libs/storefront/public_api.ts:L32-L36).
6. **Site context selector.** `SiteContextComponentService` risolve il contesto dall'`@Input context` o da `componentData.data$.context` (LANGUAGE/CURRENCY → id di contesto) (core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L91-L118), poi trova il service concreto con `injector.get(contextServiceMap[context])` (core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L120-L125). Label: `nativeName` per la lingua, `symbol + ' ' + isocode` per la valuta (core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L127-L136). Il `<select>` appare solo con più di un item e il `change` passa dal setter `active` (core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.component.html:L1-L4, core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.component.ts:L68-L70). `LanguageCurrencyComponent` affianca due selettori in un template inline (core-libs/storefront/cms-components/misc/site-context-selector/language-currency.component.ts:L11-L22). Il meccanismo dei contesti è nel capitolo 13.
7. **Theme switcher**: delega a `SiteThemeService` tramite `SiteThemeSwitcherComponentService`, fornito dal modulo e non root (core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.module.ts:L28-L36, core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.component.service.ts:L11-L26).
8. **Messaggi**: `GlobalMessageComponent` legge `globalMessageService.get()` e rimuove con `remove(type, index)` (core-libs/storefront/cms-components/misc/global-message/global-message.component.ts:L33-L51); `MessageComponent` mappa `GlobalMessageType` su classi e icone e focalizza il contenitore se ha testo accordion o bottone d'azione (core-libs/storefront/cms-components/misc/message/message.component.ts:L62-L89). Il servizio dei global message è nel capitolo 41.

3.8 Shared: Media

1. `cx-media` accetta `MediaContainer | Image | ImageGroup | ImageGroup[]`; con un array usa solo il primo elemento (core-libs/storefront/shared/components/media/media.component.ts:L62-L67, core-libs/storefront/shared/components/media/media.component.ts:L162-L174).
2. A ogni `ngOnChanges` ricalcola il `Media`; senza `src` entra in stato "missing" (`is-missing`, `is-initialized`) ed emette `loaded(false)` (core-libs/storefront/shared/components/media/media.component.ts:L155-L208). Le classi host `is-initialized`, `is-loading`, `is-missing` sono aggiornate da `loadHandler` sull'evento `(load)` (core-libs/storefront/shared/components/media/media.component.ts:L134-L148, core-libs/storefront/shared/components/media/media.component.ts:L179-L184).

3. `fetchPriority === HIGH` forza `loading="eager"`, altrimenti vale `loading` o la strategia globale (core-libs/storefront/shared/components/media/media.component.ts:L217-L222), che per default è `LAZY` (core-libs/storefront/shared/components/media/media.service.ts:L163-L168).
4. Template: `<picture>` con `<source>` se `elementType !== 'img'` e ci sono `sources`, altrimenti `<img>` con `srcset` (fallback `src`) e `sizes` (core-libs/storefront/shared/components/media/media.component.html:L1-L47).
5. `MediaService`: `sortedFormats` ordina `mediaFormats` per larghezza e lo memorizza (core-libs/storefront/shared/components/media/media.service.ts:L175-L190); `resolveFormat` usa il formato richiesto, poi il più grande disponibile, poi la prima chiave (core-libs/storefront/shared/components/media/media.service.ts:L255-L276); `resolveSrcSet` produce `"url widthw, ..."` (core-libs/storefront/shared/components/media/media.service.ts:L285-L309); i formati `<picture>` derivano da `media.pictureElementFormats` ordinati con `media.pictureFormatsOrder` (core-libs/storefront/shared/components/media/media.service.ts:L197-L231).
6. URL assoluti: `resolveAbsoluteUrl` lascia stare `http…` e `//…`, gli altri li prefissa con `getBaseUrl()` (core-libs/storefront/shared/components/media/media.service.ts:L383-L387), che usa `backend.media.baseUrl ?? backend.occ.baseUrl ?? ''` e, con `enableMediaPrefix` (default `true`), compone `new URL(backend.media.prefix, baseUrl)` con fallback e log in caso di errore (core-libs/storefront/shared/components/media/media.service.ts:L399-L428, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1032-L1032).
7. `MediaModule.forRoot()` registra un `APP_INITIALIZER` che aggiunge il link `preconnect` ed è importato da `BaseStorefrontModule` (core-libs/storefront/shared/components/media/media.module.ts:L14-L43, core-libs/storefront/base-storefront.module.ts:L42-L42).

3.9 Shared: Carousel, GenericLink, ItemCounter, FormErrors

- **Carousel.** Disattiva l'hydration con `host: { ngSkipHydration: 'true' }` (core-libs/storefront/shared/components/carousel/carousel.component.ts:L57-L65). Gli item sono Observable (caricamento lazy); cambiare `items` riporta `activeSlide` a 0 (core-libs/storefront/shared/components/carousel/carousel.component.ts:L89-L95). `template` è obbligatorio, `itemWidth` vale `'300px'` di default (core-libs/storefront/shared/components/carousel/carousel.component.ts:L101-L153). `CarouselService.getItemsPerSlide` ricalcola su `WindowRef.resize$` gli item per slide da `clientWidth` e `itemWidth` in px o % (core-libs/storefront/shared/components/carousel/carousel.service.ts:L29-L60, core-libs/core/src/window/window-ref.ts:L91-L107). Il template raggruppa gli item in `div.slide` (core-libs/storefront/shared/components/carousel/carousel.component.html:L39-L72); le frecce ←/→ spostano il focus tra `[cxFocusableCarouselItem]` cambiando slide (core-libs/storefront/shared/components/carousel/carousel.component.ts:L210-L287).
- **GenericLink.** `[href]` per URL esterni, `[routerLink]` + `queryParams` + `fragment` per quelli locali (core-libs/storefront/shared/components/generic-link/generic-link.component.html:L3-L38). Esterno = stringa che inizia con `http://`, `https://`, `mailto:` o `tel:` (core-libs/storefront/shared/components/generic-link/generic-link-component.service.ts:L12-L39). Per URL locali aggiunge lo slash iniziale e separa path/query/fragment (core-libs/storefront/shared/components/generic-link/generic-link.component.ts:L43-L127); `rel="noopener"` solo per i link esterni e solo con `target === '_blank'`; il ramo `routerLink` non imposta `rel` (core-libs/storefront/shared/components/generic-link/generic-link.component.ts:L65-L67, core-libs/storefront/shared/components/generic-link/generic-link.component.html:L3-L7, core-libs/storefront/shared/components/generic-link/generic-link.component.html:L19-L24).
- **ItemCounter.** Lavora su un `UntypedFormControl` del parent; input `min` (1), `max`, `step` (1), `allowZero`, `readonly` (core-libs/storefront/shared/components/item-counter/item-counter.component.ts:L42-L86). In `ngOnInit` corregge ogni valore entro `[min,max]` con `emitEvent: false` (core-libs/storefront/shared/

components/item-counter/item-counter.component.ts:L95-L146). I bottoni usano aria-disabled (core-libs/storefront/shared/components/item-counter/item-counter.component.html:L1-L54).

- **FormErrors.** errorsDetails$ nasce da control.statusChanges (core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L73-L88); la chiave di traduzione è un array [prefix.key, fallbackPrefix.key], accettato da TranslatePipe (core-libs/storefront/shared/components/form/form-errors/form-errors.component.html:L1-L10, core-libs/core/src/i18n/translate.pipe.ts:L42-L43); il componente è cx-visually-hidden finché il controllo non è invalid e touched/dirty, con aria-live di default polite (core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L126-L142). Poiché il cambio di touched non emette eventi, ngDoCheck lo confronta a mano (core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L96-L112).

3.10 Shared: Popover, Pagination, direttive

- **Popover.** [cxPopover] accetta stringa o TemplateRef (core-libs/storefront/shared/components/popover/popover.directive.ts:L32-L52). Tutti gli eventi passano da un unico eventSubject smistato in handlePopoverEvents (core-libs/storefront/shared/components/popover/popover.directive.ts:L117-L182). renderPopover crea PopoverComponent con ComponentFactoryResolver (deprecato in Angular, angular:packages/core/src/linker/component_factory_resolver.ts:L38-L41) e, con appendToBody, sposta il nodo nel primo elemento figlio del body (body.firstElementChild ?? body, di solito il root dell'app) (core-libs/storefront/shared/components/popover/popover.directive.ts:L187-L221, core-libs/storefront/shared/components/popover/popover.directive.ts:L210-L217). PopoverComponent si riposiziona su resize, opzionalmente su scroll e a ogni ngAfterViewChecked, e chiude su NavigationStart (core-libs/storefront/shared/components/popover/popover.component.ts:L211-L267); la posizione è calcolata da PositioningService.positionElements, che con auto prova 12 posizioni (core-libs/storefront/shared/services/positioning/positioning.service.ts:L285-L336).
- **Pagination.** PaginationBuilder.paginate non produce nulla con meno di 2 pagine e compone pagine → dots → first/last → navigazione (core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L65-L76); getStartOfRange centra la pagina corrente (core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L331-L346). PaginationComponent.getQueryParams imposta il query param e lo rimuove se uguale a defaultPage (core-libs/storefront/shared/components/list-navigation/pagination/pagination.component.ts:L199-L219). Con a11yPaginationKeyboardNavigation (default false) i link sono avvolti in uno span con cxRovingTabindex (core-libs/storefront/shared/components/list-navigation/pagination/pagination.component.html:L1-L38, core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1116-L1116).
- **Roving tabindex.** CxRovingTabindexDirective implementa il pattern APG (un solo item con tabindex 0, frecce, Home/End) (core-libs/storefront/shared/directives/roving-tabindex/roving-tabindex.directive.ts:L19-L181) e osserva solo childList con un MutationObserver creato solo se esiste (SSR) (core-libs/storefront/shared/directives/roving-tabindex/roving-tabindex.directive.ts:L121-L138). Per il quadro generale dell'accessibilità vedi il capitolo 18.
- **Horizontal scrolling position**: espone isScrollStart$, isScrollEnd$, isScrollNeeded$ (core-libs/storefront/shared/directives/horizontal-scrolling-position/horizontal-scrolling-position.directive.ts:L94-L116) calcolati con ResizeObserver/IntersectionObserver (core-libs/storefront/shared/directives/horizontal-scrolling-position/horizontal-scrolling-position.directive.ts:L199-L244) e non fa nulla fuori dal browser (core-libs/storefront/shared/directives/horizontal-scrolling-position/horizontal-scrolling-position.directive.ts:L122-L125).

3.11 Shared: servizi file, validator, config

- `FileDownloadService.download` crea, clicca e rimuove un `<a download>` (core-libs/storefront/shared/services/file/file-download.service.ts:L20-L29); `ExportCsvFileService.convert` quota solo le celle con il separatore e chiude le righe con `\r\n` (core-libs/storefront/shared/services/file/csv/export-csv-file.service.ts:L23-L52); `ImportCsvFileService` fa `split('\n')` + `split(separator)` (core-libs/storefront/shared/services/file/csv/import-csv-file.service.ts:L82-L88); `FilesFormValidators` offre `maxSize`, `maxEntries`, `allowedTypes` (core-libs/storefront/shared/services/file/files-form-validators.ts:L13-L86).
- `CustomFormValidators` raccoglie validator statici e gli array `passwordValidators`/`securePasswordValidators` (core-libs/storefront/shared/utils/validators/custom-form-validators.ts:L26-L218); `controlsMustMatch` non tocca il secondo controllo se ha già altri errori (core-libs/storefront/shared/utils/validators/custom-form-validators.ts:L444-L463); `sortTitles` ordina per `titleScores` (core-libs/storefront/shared/utils/forms/title-utils.ts:L9-L27).
- `FormConfig` e `ViewConfig` sono config augmentate `useExisting: Config` (core-libs/storefront/shared/config/form-config.ts:L10-L22, core-libs/storefront/shared/config/view-config.ts:L10-L30); `defaultFormConfig` è fornito da `PasswordVisibilityToggleModule` (core-libs/storefront/shared/config/default-form-config.ts:L9-L13, core-libs/storefront/shared/components/form/password-visibility-toggle/password-visibility-toggle.module.ts:L23-L23). `CaptchaModule` fornisce di default un renderer mock (core-libs/storefront/shared/components/captcha/captcha.module.ts:L15-L25); `TableModule` fornisce i componenti header/cella globali (core-libs/storefront/shared/components/table/table.module.ts:L28-L28, core-libs/storefront/shared/components/table/config/default-table.config.ts:L11-L16).

---

4. Diagrammi

4.1 Render di un banner CMS

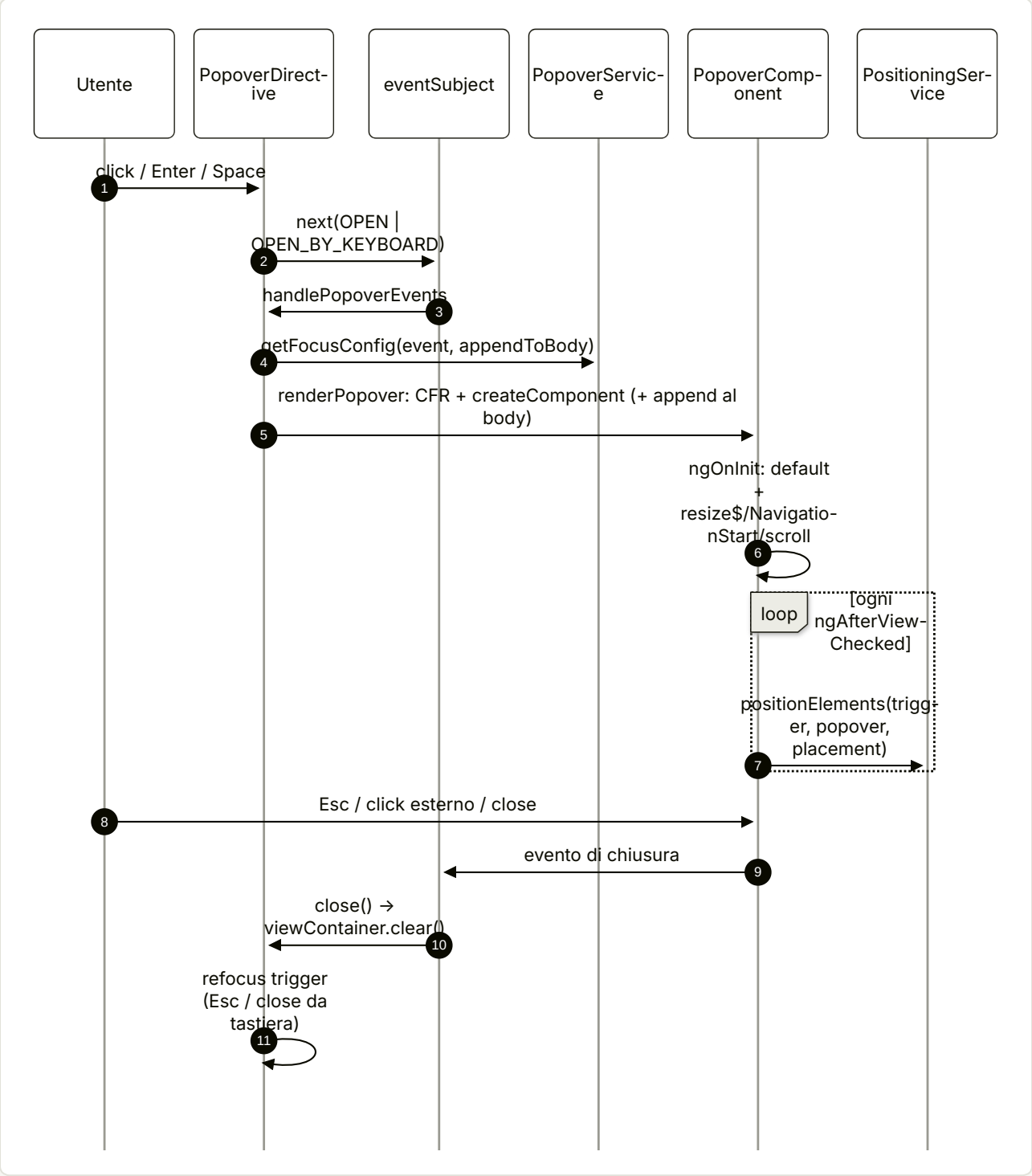
Ogni messaggio corrisponde ai passi 3.1–3.2 e 3.8–3.9.

■ **TAVOLA 45.1** diagramma a tutta pagina, in fondo al capitolo

4.2 Costruzione dell'albero di navigazione

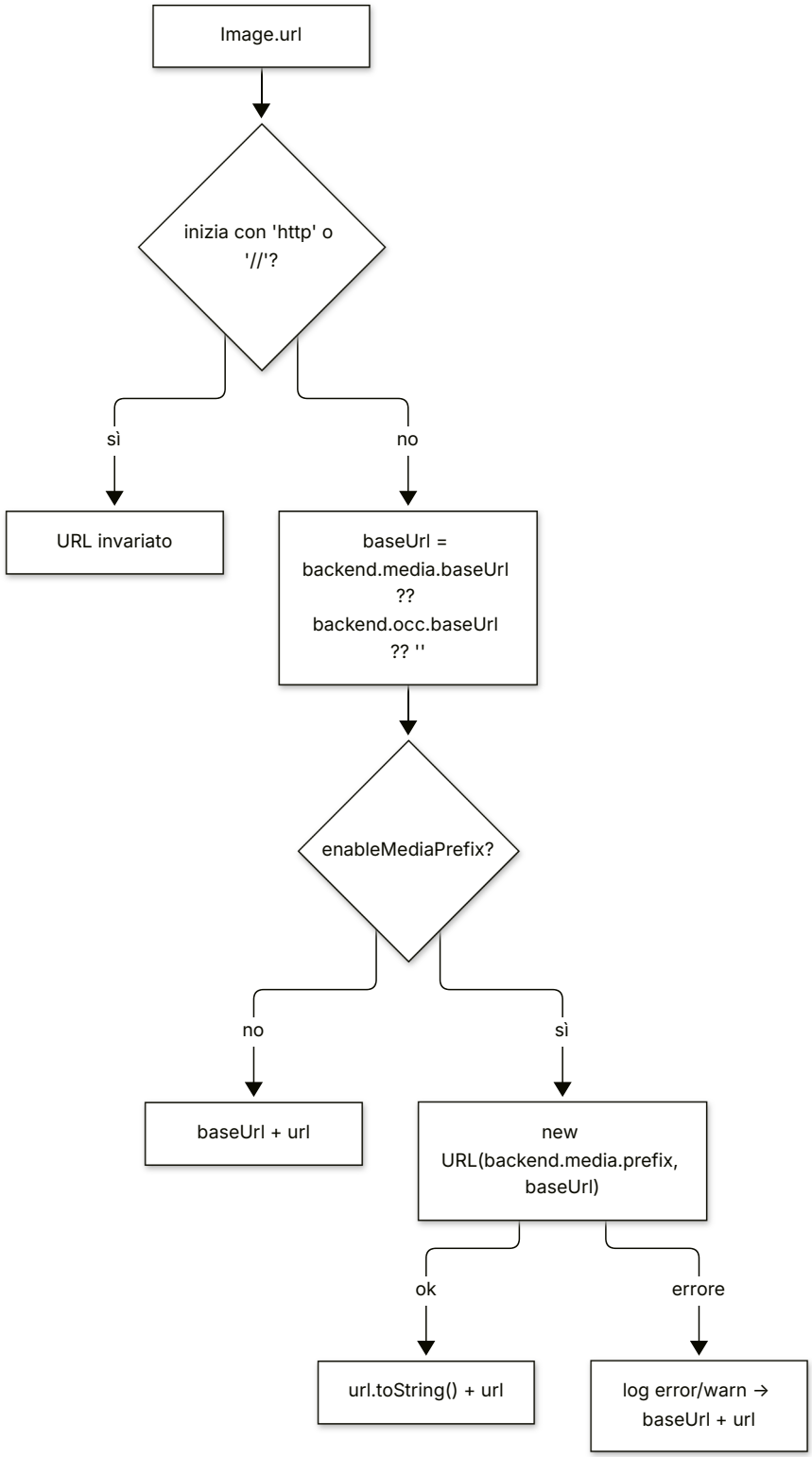
■ **TAVOLA 45.2** diagramma a tutta pagina, in fondo al capitolo

4.3 Popover: ciclo di vita degli eventi



Riferimenti per il diagramma 4.3: `core-libs/storefront/shared/components/popover/popover.directive.ts:L83-L115` (apertura), `core-libs/storefront/shared/components/popover/popover.directive.ts:L142-L221` (gestione e render), `core-libs/storefront/shared/components/popover/popover.component.ts:L226-L252` (init), `core-libs/storefront/shared/components/popover/popover.component.ts:L159-L192` (chiusura), `core-libs/storefront/shared/components/popover/popover.service.ts:L34-L38` (refocus).

4.4 Risoluzione dell'URL di un'immagine



5. Estratti di codice reale

5.1 Il modulo wrapper del banner

core-libs/storefront/cms-components/content/banner/banner.module.ts:L20-L47

```ts
@NgModule({
  imports: [
    CommonModule,
    RouterModule,
    GenericLinkModule,
    MediaModule,
    FeaturesConfigModule,
    LcpContextDirectiveModule,
    BannerComponent,
  ],
  providers: [
    provideDefaultConfig(<CmsConfig>{
      cmsComponents: {
        SimpleResponsiveBannerComponent: {
          component: BannerComponent,
        },
        BannerComponent: {
          component: BannerComponent,
        },
        SimpleBannerComponent: {
          component: BannerComponent,
        },
      },
    }),
  ],
  exports: [BannerComponent],
})
export class BannerModule {}
```

5.2 Injector per istanza e merge dei dati statici

core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50

```ts
  public getInjector(
    type: string,
    uid: string,
    parentInjector?: Injector
  ): Injector {
    const configProviders =
      this.cmsComponentsService.getMapping(type)?.providers ?? [];
    return Injector.create({
```

```
    providers: [
      {
        provide: CmsComponentData,
        useFactory: (dataProvider: ComponentDataProvider) => ({
          uid,
          data$: dataProvider.get(uid, type),
        }),
        deps: [ComponentDataProvider],
      },
      provideLcpPresenceForCmsComponent(),
      ...configProviders,
    ],
    parent: parentInjector ?? this.injector,
  });
}
```

core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L35-L59

```ts
get<T extends CmsComponent>(uid: string, type?: string): Observable<T> {
  return defer(() => {
    let staticComponentData: T | undefined;

    if (type) {
      staticComponentData = this.componentsService.getStaticData<T>(type);
    }

    if (uid) {
      if (staticComponentData) {
        return this.cmsService.getComponentData<T>(uid).pipe(
          map((data) => ({
            ...staticComponentData,
            ...data,
          })),
          startWith(staticComponentData)
        );
      } else {
        return this.cmsService.getComponentData<T>(uid);
      }
    } else {
      return staticComponentData ? of(staticComponentData) : EMPTY;
    }
  });
}
```

5.3 Banner: `data$` e cascata del link

core-libs/storefront/cms-components/content/banner/banner.component.ts:L44-L57

```ts
export class BannerComponent {
  routerLink: string | any[] | undefined;

  @HostBinding('class') styleClasses: string | undefined;

  private featureToggles = inject(FeatureToggles);
  protected el = inject(ElementRef);

  data$: Observable<CmsBannerComponent> = this.component.data$.pipe(
    tap((data) => {
      this.setRouterLink(data);
      this.styleClasses = data.styleClasses;
    })
  );
```

core-libs/storefront/cms-components/content/banner/banner.component.ts:L137-L161

```ts
  protected setRouterLink(data: CmsBannerComponent): void {
    if (data.urlLink) {
      this.routerLink = data.urlLink;
    } else if (data.contentPage) {
      this.cmsService
        .getPage({
          id: data.contentPage,
          type: PageType.CONTENT_PAGE,
        })
        .pipe(take(1))
        .subscribe((page) => {
          this.routerLink = page?.label;
        });
    } else if (data.product) {
      this.routerLink = this.urlService.transform({
        cxRoute: 'product',
        params: { code: data.product },
      });
    } else if (data.category) {
      this.routerLink = this.urlService.transform({
        cxRoute: 'category',
        params: { code: data.category },
      });
    }
  }
```

5.4 Paragraph: HTML fidato e click intercettati

core-libs/storefront/cms-components/content/paragraph/paragraph.component.html:L1-L4

```HTML
<div
  *ngIf="component.data$ | async as data"
  [innerHTML]="bypassSecurityTrustHtml(data.content | cxSupplementHashAnchors)"
></div>
```

core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L29-L42

```TS
  @HostListener('click', ['$event'])
  public handleClick(event: Event): void {
    if (event.target instanceof HTMLAnchorElement) {
      const element = event.target as HTMLAnchorElement;
      const href = element?.getAttribute('href');
      const documentUrlObject = new URL(element.ownerDocument.URL);

      // Use router for internal link navigation
      if (href && documentUrlObject.host === element.host) {
        event.preventDefault();
        this.router.navigateByUrl(href.replace(documentUrlObject.origin, ''));
      }
    }
  }
```

5.5 TabParagraphContainer: dal campo `components` ai tab

core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L99-L140

```TS
  components$: Observable<any[]> = this.componentData.data$.pipe(
    distinctUntilChanged((x, y) => x?.components === y?.components),
    tap((data: CMSTabParagraphContainer) => {
      this.ariaLabel = `${data?.uid}.tabPanelContainerRegion`;
    }),
    switchMap((data) =>
      combineLatest(
        (data?.components ?? '').split(' ').map((component) =>
          this.cmsService.getComponentData<any>(component).pipe(
            distinctUntilChanged(),
            map((tab) => {
              if (!tab) {
                return undefined;
              }

              if (!tab.flexType) {
                tab = {
                  ...tab,
                  flexType: tab.typeCode,
                };
              }
```

```
            return {
              ...tab,
              title: `${data.uid}.tabs.${tab.uid}`,
            };
          })
        )
      )
    ).pipe(
      // Remove tabs whose component data is null or undefined
      map((tabs) => tabs.filter((tab) => tab != null)),
      // Update tablist label with name from CMS
      tap(() => {
        this.tabConfig$.next({
          label: `${data?.uid}.tabPanelContainerRegionGroup`,
          ...defaultTabConfig,
        });
      })
    )
  )
);
```

core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L202-L213

```ts
tabCompLoaded(componentRef: any, componentId?: string): void {
  this.tabTitleParams.push(componentRef.instance.tabTitleParam$);
  if (componentId) {
    const currentParams = this.tabTitleParams$.getValue();
    // We need to check if the componentId is already in the map
    // to avoid infinite loop(tabs$ emit -> component load -> tabTitleParams$ emit -> tabs$ emit)
    if (!currentParams.has(componentId)) {
      currentParams.set(componentId, componentRef.instance.tabTitleParam$);
      this.tabTitleParams$.next(currentParams);
    }
  }
}
```

core-libs/storefront/cms-components/content/tab/tab.component.ts:L254-L268

```ts
protected getMode(): Observable<TAB_MODE> {
  if (this.config.mode) {
    return of<TAB_MODE>(this.config.mode);
  }

  if (this.config.breakpoint) {
    return this.breakpointService
      .isUp(this.config.breakpoint)
```

```
        .pipe(
          map((isUp: boolean) => (isUp ? TAB_MODE.TAB : TAB_MODE.ACCORDIAN))
        );
    }

    return of<TAB_MODE>(TAB_MODE.TAB);
  }
```

5.6 NavigationService: caricamento incrementale e link

core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L51-L84

```ts
    return data$.pipe(
      filter((data) => !!data),
      switchMap((data) => {
        const navigation = data.navigationNode ? data.navigationNode : data;
        return this.cmsService
          .getNavigationEntryItems(navigation.uid ?? '')
          .pipe(
            tap((items) => {
              if (items === undefined) {
                this.loadNavigationEntryItems(navigation, true);
                return;
              }
              // we should check whether the existing node items are what expected
              const expectedItems: {
                superType: string | undefined;
                id: string | undefined;
              }[] = [];
              this.loadNavigationEntryItems(navigation, false, expectedItems);
              const existingItems = Object.keys(items).map(
                (key) => items[key].uid ?? ''
              );
              const missingItems = expectedItems.filter(
                (it) => it.id && !existingItems.includes(it.id)
              );
              if (missingItems.length > 0) {
                this.cmsService.loadNavigationItems(
                  navigation.uid ?? '',
                  missingItems
                );
              }
            }),
            filter(Boolean),
            map((items) => this.populateNavigationNode(navigation, items) ?? {})
          );
```

core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L166-L183

```ts
private populateLink(node: NavigationNode, entry: any, items: any) {
  const item = items[`${entry.itemId}_${entry.itemSuperType}`];

  // now we only consider CMSLinkComponent
  if (item && entry.itemType === 'CMSLinkComponent') {
    if (!node.title) {
      node.title = item.linkName;
    }
    const url = this.getLink(item);
    // only populate the node link if we have a visible node
    if (node.title && url) {
      node.url = url;
      // the backend provide boolean value for the target
      // in case the link should be opened in a new window
      if (item.target === 'true' || item.target === true) {
        node.target = '_blank';
      }
    }
```

5.7 NavigationUIComponent: tastiera dipendente dal feature toggle

core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L197-L229

```ts
toggleOpen(event: UIEvent): void {
  const isKeyboard = event.type === 'keydown' || event.type === 'keyup';
  if (this.featureToggles?.a11yNavigationSpaceKeyOnKeyUp) {
    if (event.type === 'keydown') {
      event.preventDefault();
      return;
    }
  } else if (event.type === 'keyup') {
    return;
  }
  if (isKeyboard) {
    event.preventDefault();
  }
  this.ariaCollapseNodes();
  const node = <HTMLElement>event.currentTarget;
  const parentNode = <HTMLElement>node.parentNode;
  if (this.openNodes.includes(parentNode)) {
    if (isKeyboard) {
      this.back();
    } else {
      this.openNodes = this.openNodes.filter((n) => n !== parentNode);
      this.renderer.removeClass(parentNode, 'is-open');
    }
  } else {
    this.openNodes.push(parentNode);
```

```
      this.renderer.setAttribute(node, ARIA_EXPANDED_ATTR, 'true');
    }

    this.updateClasses();

    event.stopImmediatePropagation();
    event.stopPropagation();
  }
```

5.8 Site context: provider per componente e risoluzione del service

core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.module.ts:L33-L52

```ts
  providers: [
    provideDefaultConfig(<CmsConfig>{
      cmsComponents: {
        CMSSiteContextComponent: {
          component: SiteContextSelectorComponent,
          providers: [
            {
              provide: SiteContextComponentService,
              useClass: SiteContextComponentService,
              deps: [CmsComponentData, ContextServiceMap, Injector],
            },
          ],
        },
        LanguageCurrencyComponent: {
          component: LanguageCurrencyComponent,
        },
      },
    }),
    SiteContextComponentService,
  ],
```

core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L120-L125

```ts
  protected getInjectedService(context: string): SiteContext<any> {
    return this.injector.get<SiteContext<any>>(
      this.contextServiceMap[context],
      undefined
    );
  }
```

5.9 Icone: HTML sicuro e risorse LINK

core-libs/storefront/cms-components/misc/icon/icon.component.ts:L91-L99

```ts
protected setIcon(type: ICON_TYPE): void {
  if (!type || <string>type === '') {
    return;
  }
  this.icon = this.iconLoader.getHtml(type);
  this.addStyleClasses(type);
  this.iconLoader.addLinkResource(type);
  this.flipIcon(type);
}
```

core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L33-L55

```ts
getHtml(type: ICON_TYPE | string): SafeHtml | undefined {
  if (this.isResourceType(type, IconResourceType.SVG)) {
    const url = this.sanitizer.sanitize(
      SecurityContext.URL,
      this.getSvgPath(type) || null
    );
    if (url) {
      const useElement = this.winRef.document.createElement('use');
      useElement.setAttribute('xlink:href', url);
      const svgElement = this.winRef.document.createElement('svg');
      svgElement.appendChild(useElement);
      return this.sanitizer.bypassSecurityTrustHtml(svgElement.outerHTML);
    }
  }
  if (this.isResourceType(type, IconResourceType.TEXT)) {
    const symbol = this.getSymbol(type);
    if (symbol) {
      const helperDiv = this.winRef.document.createElement('div');
      helperDiv.textContent = symbol;
      return this.sanitizer.bypassSecurityTrustHtml(helperDiv.innerHTML);
    }
  }
}
```

core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L139-L160

```ts
private findResource(
  iconType: ICON_TYPE | string,
  resourceType: IconResourceType
): IconConfigResource | undefined {
  if (!this.config?.resources) {
    return;
  }

  let resource = this.config.resources.find(
    (res) =>
```

```
      res.type === resourceType && res.types && res.types.includes(iconType)
  );
  // no specific resource found, let's try to find a one-size-fits-all resource
  if (!resource) {
    resource = this.config.resources.find(
      (res) =>
        (res.type === resourceType && !res.types) ||
        (res.types && res.types.length === 0)
    );
  }
  return resource;
}
```

5.10 MediaService: base URL e srcset

core-libs/storefront/shared/components/media/media.service.ts:L399-L428

TS

```
public getBaseUrl(): string {
  const baseUrl =
    this.config.backend?.media?.baseUrl ??
    this.config.backend?.occ?.baseUrl ??
    '';

  if (!this.featureToggles.enableMediaPrefix) {
    return baseUrl;
  }

  try {
    const mediaPrefix = this.config?.backend?.media?.prefix ?? '';
    // Construct URL (with Validation)
    const url = new URL(mediaPrefix, baseUrl);
    return url.toString();
  } catch (error) {
    if (isDevMode()) {
      this.logger.error(
        'Invalid media/occ baseUrl and prefix configuration',
        error
      );
    } else {
      this.logger.warn(
        'Invalid media/occ baseUrl and prefix configuration',
        error
      );
    }
    return baseUrl;
  }
}
```

core-libs/storefront/shared/components/media/media.service.ts:L175-L190

```ts
protected get sortedFormats(): { code: string; size: MediaFormatSize }[] {
  const mediaFormats = this.config?.mediaFormats;

  if (!this._sortedFormats && mediaFormats) {
    this._sortedFormats = Object.keys(mediaFormats)
      .map((key) => ({
        code: key,
        size: mediaFormats[key],
      }))
      .sort((a, b) =>
        a.size.width && b.size.width && a.size.width > b.size.width ? 1 : -1
      );
  }

  return this._sortedFormats ?? [];
}
```

5.11 Carousel: item per slide

core-libs/storefront/shared/components/carousel/carousel.service.ts:L29-L60

```ts
getItemsPerSlide(
  nativeElement: HTMLElement,
  itemWidth: string
): Observable<number> {
  return this.winRef.resize$.pipe(
    map(() => (nativeElement as HTMLElement).clientWidth),
    map((totalWidth) => this.calculateItems(totalWidth, itemWidth))
  );
}

/**
 * Calculates the number of items per given hostSize.  calculated based on the given
 * intended size in pixels or percentages. The
 *
 * @param availableWidth The available width in pixels for the carousel items.
 * @param itemWidth The width per carousel item, in px or percentage.
 */
private calculateItems(availableWidth: number, itemWidth: string) {
  let calculatedItems = 0;
  if (itemWidth.endsWith('px')) {
    const num = itemWidth.substring(0, itemWidth.length - 2);
    calculatedItems = availableWidth / <number>(<any>num);
  }

  if (itemWidth.endsWith('%')) {
    const perc = itemWidth.substring(0, itemWidth.length - 1);
```

```
      calculatedItems =
        availableWidth / (availableWidth * (<number>(<any>perc) / 100));
    }

    return Math.floor(calculatedItems) || 1;
  }
```

5.12 GenericLink: cosa è "esterno"

core-libs/storefront/shared/components/generic-link/generic-link-component.service.ts:L12-L39

```ts
export class GenericLinkComponentService {
  /**
   * Pattern matching string starting with `http://` or `https://`.
   */
  protected HTTP_PROTOCOL_REGEX: RegExp = /^https?:\/\//i;

  /**
   * Pattern matching string starting with `mailto:`.
   */
  protected MAILTO_PROTOCOL_REGEX: RegExp = /^mailto:/i;

  /**
   * Pattern matching string starting with `tel:`.
   */
  protected TEL_PROTOCOL_REGEX: RegExp = /^tel:/i;

  /**
   * Returns true when the @Input `url` is a string starting with `http://`, `https://`, `mailto:`,
`tel:`.
   */
  isExternalUrl(url: string | any[]): boolean {
    return (
      typeof url === 'string' &&
      (this.HTTP_PROTOCOL_REGEX.test(url) ||
        this.MAILTO_PROTOCOL_REGEX.test(url) ||
        this.TEL_PROTOCOL_REGEX.test(url))
    );
  }
}
```

5.13 ItemCounter: clamp del valore

core-libs/storefront/shared/components/item-counter/item-counter.component.ts:L95-L101

```ts
ngOnInit() {
  this.sub = this.control.valueChanges
    .pipe(startWith(this.control.value))
    .subscribe((value) =>
      this.control.setValue(this.getValidCount(value), { emitEvent: false })
    );
}
```

core-libs/storefront/shared/components/item-counter/item-counter.component.ts:L138-L146

```ts
private getValidCount(value: number) {
  if (value < this.min && !(value === 0 && this.allowZero)) {
    value = this.min;
  }
  if (this.max && value > this.max) {
    value = this.max;
  }
  return value;
}
```

5.14 FormErrors: ngDoCheck e il workaround per `aria-live`

core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L96-L112

```ts
ngDoCheck(): void {
  if (this.control.touched !== this.previousTouchedState) {
    if (
      this.elementRef?.nativeElement?.getAttribute('aria-live') === 'polite'
    ) {
      // due to the way we detect changes here, JAWS doesn't always respect
      // aria live `polite`, so we need to move this in the next event-loop queue
      setTimeout(() => {
        this.previousTouchedState = this.control.touched;
        this.ChangeDetectionRef.markForCheck();
      });
    } else {
      this.previousTouchedState = this.control.touched;
      this.ChangeDetectionRef.markForCheck();
    }
  }
}
```

5.15 SearchBox: merge a tre livelli

core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L246-L266

```ts
  protected config$: Observable<SearchBoxConfig> = (
    this.componentData?.data$ || of({} as any)
  ).pipe(
    map((config) => {
      const isBool = (obj: SearchBoxConfig, prop: string): boolean =>
        obj[prop as keyof SearchBoxConfig] !== 'false' &&
        obj[prop as keyof SearchBoxConfig] !== false;

      return {
        ...DEFAULT_SEARCH_BOX_CONFIG,
        ...config,
        displayProducts: isBool(config, 'displayProducts'),
        displayProductImages: isBool(config, 'displayProductImages'),
        displaySuggestions: isBool(config, 'displaySuggestions'),
        // we're merging the (optional) input of this component, but write the merged
        // result back to the input property, as the view logic depends on it.
        ...this.config,
      };
    }),
    tap((config) => (this.config = config))
  );
```

5.16 Pagination: fallback e mutazione

core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L16-L23

```ts
const FALLBACK_PAGINATION_OPTIONS: PaginationOptions = {
  rangeCount: 3,
  dotsLabel: '...',
  startLabel: '«',
  previousLabel: '‹',
  nextLabel: '›',
  endLabel: '»',
};
```

core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L370-L375

```ts
  protected get config(): PaginationOptions {
    return Object.assign(
      FALLBACK_PAGINATION_OPTIONS,
      this.paginationConfig.pagination
    );
  }
```

5.17 Popover: FocusConfig solo con appendToBody

core-libs/storefront/shared/components/popover/popover.service.ts:L19-L32

```ts
getFocusConfig(event: PopoverEvent, appendToBody: boolean): FocusConfig {
  let config = {};

  if (event === PopoverEvent.OPEN_BY_KEYBOARD && appendToBody) {
    config = {
      trap: true,
      block: true,
      focusOnEscape: false,
      autofocus: true,
    };
  }

  return config;
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`cmsComponents.<typeCode>` (banner, link, paragraph, navigation, ...)	`CmsComponentMapping`	vedi sezione 3.1 e A03	moduli wrapper, es. `core-libs/storefront/cms-components/content/banner/banner.module.ts:L31-L44`	`core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L33-L34`
`cmsComponents.CategoryNavigationComponent.data.resetMenuOnClose`	`boolean`	`true`	`core-libs/storefront/cms-components/navigation/category-navigation/category-navigation.module.ts:L25-L27`	`core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L40-L51`
`icon.symbols`	`{[type]: string}`	classi Font Awesome	`core-libs/storefront/cms-components/misc/icon/fontawesome-icon.config.ts:L11-L78`	`core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L162-L166`
`icon.resources`	`IconConfigResource[]`	`[{ type: LINK, url: '' }]`	`core-libs/storefront/cms-components/misc/icon/fontawesome-icon.config.ts:L79-L84`	`core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L139-L160`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
icon.flipDirection	{[type]: DirectionMode}	CARET_RIGHT/CARET_LEFT: RTL	core-libs/storefront/cms-components/misc/icon/default-icon.config.ts:L10-L17	core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L61-L63
mediaFormats	{[format]: { width }}	— nella libreria; la recipe mediaConfig (mobile: 400 … zoom: 515) è fornita dall'app con provideConfig(mediaConfig) e dallo schematic add-spartacus (core-libs/storefront/recipes/config/default-media.config.ts:L9-L37, projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L47-L47, core-libs/schematics/src/add-spartacus/configuration.ts:L105-L111)	core-libs/storefront/shared/components/media/media.config.ts:L58-L64	core-libs/storefront/shared/components/media/media.service.ts:L175-L190
media.pictureElementFormats / media.pictureFormatsOrder	object / string[]	— nella libreria; recipe mediaConfig: 4 formati, pictureFormatsOrder: ['mobile', 'tablet', 'desktop', 'widescreen'] (core-libs/storefront/recipes/config/default-media.config.ts:L21-L36)	core-libs/storefront/shared/components/media/media.config.ts:L27-L52	core-libs/storefront/shared/components/media/media.service.ts:L197-L231
imageLoadingStrategy	ImageLoadingStrategy	LAZY (fallback nel service)	core-libs/storefront/shared/components/media/media.config.ts:L76-L76	core-libs/storefront/shared/components/media/media.service.ts:L163-L168
backend.media.baseUrl / backend.occ.baseUrl / backend.media.prefix	string	''	config core (capitolo 11)	core-libs/storefront/shared/components/media/media.service.ts:L399-L428
pagination (PaginationOptions)	object	{ addStart: true, addEnd: true } + fallback rangeCount: 3 e label	core-libs/storefront/shared/components/list-navigation/pagination/config/default-pagination.config.ts:L9-L14, core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L16-L23	core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L370-L375

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`form.passwordVisibilityToggle`	`boolean`	`true`	`core-libs/storefront/shared/config/default-form-config.ts:L9-L13`	`core-libs/storefront/shared/components/form/password-visibility-toggle/password-visibility-toggle.directive.ts:L33-L36`
`view.defaultPageSize`, `view.infiniteScroll.*`	number / object	non definito qui	`core-libs/storefront/shared/config/view-config.ts:L14-L26`	fuori capitolo (product list, capitolo 24)
`captchaRenderer`, `apiUrl`, `fields`	`Type` / string / object	`captchaRenderer: MockCaptchaService`	`core-libs/storefront/shared/components/captcha/mock-captcha/config/mock-captcha-api-config.ts:L10-L12`, `core-libs/storefront/shared/components/captcha/captcha-api-config.ts:L11-L33`	**DA VERIFICARE**: `CaptchaComponent` non letto per intero
`tableOptions.headerComponent` / `dataComponent`, `table[type]`	`Type` / config	header/data cell di default	`core-libs/storefront/shared/components/table/config/default-table.config.ts:L11-L16`, `core-libs/storefront/shared/components/table/config/table.config.ts:L20-L43`	`TableService` (solo mappato)
`SearchBoxConfig` (dati CMS o `@Input config`)	`SearchBoxConfig`	`DEFAULT_SEARCH_BOX_CONFIG`	`core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L61-L72`	`core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L246-L266`
`TabConfig` del container	`TabConfig`	`{ openTabs: [0], breakpoint: md }`	`core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L39-L42`	`core-libs/storefront/cms-components/content/tab/tab.component.ts:L66-L68`, `core-libs/storefront/cms-components/content/tab/tab.component.ts:L254-L268`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`itemWidth` del carousel	string px/%	`'300px'`	`core-libs/storefront/shared/components/carousel/carousel.component.ts:L110-L110`	`core-libs/storefront/shared/components/carousel/carousel.service.ts:L46-L60`

Feature toggle letti dai componenti del capitolo (default in `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts`; il meccanismo è nel capitolo 42 e l'elenco completo nell'appendice A09):

TOGGLE	DEFAULT	RIGA DEFAULT	EFFETTO IN QUESTO CAPITOLO
`enableMediaPrefix`	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1032-L1032`	`getBaseUrl` compone `backend.media.prefix`
`searchBoxRecentSearchesRemoval`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1041-L1041`	search box (**DA VERIFICARE** dettaglio, vedi 3.6)
`searchBoxEmptyQueryResultsPanel`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1042-L1042`	search box (**DA VERIFICARE** dettaglio, vedi 3.6)
`a11yCarouselPreventNavigationFocus`	`true`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1047-L1047`	prev/next del carousel non prendono il focus al `mousedown`
`a11yCarouselPreventIndicatorFocus`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1048-L1048`	indicatori del carousel
`a11yItemCounterValueText`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1064-L1064`	`aria-valuetext` sull'item counter
`a11yRestoreFocusOnNgSelect`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1065-L1065`	`SortingComponent` dispatcha un focus sintetico
`a11ySiteContextCaretClick`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1077-L1077`	site context selector
`a11yNavigationSpaceKeyOnKeyUp`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1093-L1093`	`toggleOpen` su `keyup` invece che `keydown`
`a11yBannerArrowKeyNavigation`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1114-L1114`	frecce tra banner fratelli
`a11yPaginationKeyboardNavigation`	`false`	`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1116-L1116`	roving tabindex sulla paginazione

7. Commenti e note notevoli dal sorgente

- `// TODO: move the opinionated fontawesome config to a recipe` — la config Font Awesome vive ancora in `IconModule` (core-libs/storefront/cms-components/misc/icon/icon.module.ts:L18-L18). Se l'app non usa Font Awesome deve sovrascrivere `icon.symbols`/`icon.resources`.
- `NOTE: this is not working when the shadow is used as there's no head element available...` — `addLinkResource` non funziona dentro uno shadow DOM (core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L114-L116).
- JSDoc del carousel: l'hydration è disattivata per differenze CSR/SSR (core-libs/storefront/shared/components/carousel/carousel.component.ts:L57-L59).
- `// DELIBERATELY PRIVATE PROPERTY, to remove easily in the future` sul campo `featureToggles` (core-libs/storefront/shared/components/carousel/carousel.component.ts:L136-L136, core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L50-L50): non fateci affidamento in una sottoclasse.
- `// TODO: Improve a11y with better text appropriate to usage (example: loading cart spinner)` (core-libs/storefront/shared/components/spinner/spinner.component.ts:L10-L10).
- `// TODO: Insert mustEndWithLegalCharacter into this array when removing feature toogle: useEnhancedSecurePasswordValidators` (core-libs/storefront/shared/utils/validators/custom-form-validators.ts:L208-L208).
- `// CHECK SONAR` in fondo a `navigation.service.ts` (core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L226-L226).
- `// to avoid infinite loop(tabs$ emit -> component load -> tabTitleParams$ emit -> tabs$ emit)` in `tabCompLoaded` (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L206-L207).
- `// Convert string value to number(prevents us from having '1' + 1 = '11' on increment)` in `updateValue` (core-libs/storefront/shared/components/item-counter/item-counter.component.ts:L114-L114).
- Il commento sul workaround di `aria-live` in `FormErrorsComponent.ngDoCheck` (estratto 5.14, core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L101-L102).
- `@deprecated This method will be removed.` su `activeTabNum`, `ariaLabel`, `children`, `tabTitleParams`, `select()`, `ngOnInit()` di `TabParagraphContainerComponent` (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L59-L80, core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L144-L167); `@deprecated since 2011.21 removed unused property` su `siteContextService` (core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.component.ts:L43-L46). Elenco completo nell'appendice A10.
- Commento esteso sul wrapper `span display:contents` per il roving tabindex nella paginazione (core-libs/storefront/shared/components/list-navigation/pagination/pagination.component.html:L3-L13) e JSDoc della classe `CxRovingTabindexDirective` su SSR, host tabella e gating via feature toggle a carico del chiamante (core-libs/storefront/shared/directives/roving-tabindex/roving-tabindex.directive.ts:L19-L54).
- `<!-- https://github.com/angular/angular/issues/24567 -->` in testa al template generic-link (core-libs/storefront/shared/components/generic-link/generic-link.component.html:L1-L1).
- JSDoc di `MediaComponent` sull'uso di più media query invece di `srcset` con densità nei `<source>` (core-libs/storefront/shared/components/media/media.component.ts:L27-L48).

8. Extension point

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
Mapping `cmsComponents` (sostituire `component`, aggiungere `providers`, `data`)	`provideConfig(<CmsConfig>{ cmsComponents: { SimpleResponsiveBannerComponent: { component: MyBanner } } });` ordine e precedenza della config nel capitolo 11	core-libs/storefront/cms-components/content/banner/banner.module.ts:L31-L44	Ricette CMS, outlet e layout
Estendere un componente CMS (ereditarietà + `CmsComponentData`)	i componenti espongono metodi `protected` (es. `setRouterLink` del banner, `getLink` di `NavigationService`)	core-libs/storefront/cms-components/content/banner/banner.component.ts:L137-L161, core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L201-L223	Ricette CMS, outlet e layout
`providers` per componente (pattern site context)	fornire un service con `CmsComponentData` nelle deps solo per un typeCode	core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.module.ts:L36-L45	Ricette CMS, outlet e layout
`IconConfig` (`symbols`, `resources`, `flipDirection`) e `ICON_TYPE` augmentabile	sostituire Font Awesome con sprite SVG o testo	core-libs/storefront/cms-components/misc/icon/icon.model.ts:L11-L149	Ricette config/DI, Ricette stili
Override di service `providedIn: 'root'` (`IconLoaderService`, `NavigationService`, `SearchBoxComponentService`, `CarouselService`, `GenericLinkComponentService`, `MediaService`)	`{ provide: X, useClass: MyX };` molti metodi sono `protected` (`resolveFormat`, `resolveSrcSet`, `resolveAbsoluteUrl`, regex di `GenericLinkComponentService`)	core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L33-L160, core-libs/storefront/shared/components/media/media.service.ts:L255-L428, core-libs/storefront/shared/components/generic-link/generic-link-component.service.ts:L12-L39	Ricette config/DI
`MediaConfig` (`mediaFormats`, `media.pictureElementFormats`, `media.pictureFormatsOrder`, `imageLoadingStrategy`)	config	core-libs/storefront/shared/components/media/media.config.ts:L19-L81	Ricette config/DI

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
`PaginationConfig` / `PaginationBuilder` (step `protected`)	config o sottoclasse del builder	`core-libs/storefront/shared/components/list-navigation/pagination/config/pagination.config.ts:L11-L21`, `core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L65-L76`	Ricette config/DI
`SearchBoxConfig` (dati CMS o `@Input`) e `SearchBoxOutlets`	config per istanza o outlet per recent/trending searches	`core-libs/storefront/cms-components/navigation/search-box/search-box-outlets.model.ts:L7-L11`	Ricette CMS, outlet e layout
Eventi `SearchBoxSuggestionSelectedEvent`, `SearchBoxProductSelectedEvent`	`EventService.get(...)`	`core-libs/storefront/cms-components/navigation/search-box/search-box.events.ts:L12-L32`	Ricette config/DI (eventi)
`PopoverOptions`, `PopoverService`, `PositioningService`	opzioni della direttiva o override del service	`core-libs/storefront/shared/components/popover/popover.model.ts:L27-L84`, `core-libs/storefront/shared/services/positioning/positioning.service.ts:L285-L348`	Ricette config/DI
`FormConfig`, `ViewConfig`, `CaptchaApiConfig.captchaRenderer`, `TableConfig`	config	`core-libs/storefront/shared/config/form-config.ts:L10-L22`, `core-libs/storefront/shared/config/view-config.ts:L10-L30`, `core-libs/storefront/shared/components/captcha/captcha-api-config.ts:L11-L33`, `core-libs/storefront/shared/components/table/config/table.config.ts:L20-L43`	Ricette config/DI
`CustomFormValidators`, `controlsMustMatch`, `sortTitles`, pipe e direttive shared	riuso diretto nei propri componenti	`core-libs/storefront/shared/utils/validators/custom-form-validators.ts:L26-L463`	Ricette feature/checkout

EXTENSION POINT	COME SI USA	DEFINIZIONE	RICETTA
Traduzioni delle chiavi usate qui (`navigation.categoryNavLabel`, `formErrors.*`, `<uid>.tabs.<tabUid>`, `common.language`)	override delle risorse i18n (capitolo 37)	`core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L121-L124`	Ricette traduzioni
Testing: `IconTestingModule` (mock pubblico)	importarlo negli spec al posto di `IconModule`	`core-libs/storefront/cms-components/misc/icon/testing/icon-testing.module.ts:L10-L50`	Upgrade-safe

Cosa **non** è un extension point: `defaultIconConfig`, `tab.utils`, `defaultFormConfig` e `form-test-utils` non sono esportati dai rispettivi barrel (`core-libs/storefront/cms-components/misc/icon/index.ts:L7-L12`, `core-libs/storefront/cms-components/content/index.ts:L7-L25`, `core-libs/storefront/shared/config/index.ts:L7-L8`, `core-libs/storefront/shared/utils/index.ts:L7-L8`); importarli da path interni è l'anti-pattern descritto in Upgrade-safe. Il campo `featureToggles` di carousel e search box è dichiarato "deliberately private" (sezione 7). I barrel `generic-link/index.ts` e `popover/index.ts` riesportano invece tutti i file (`core-libs/storefront/shared/components/generic-link/index.ts:L7-L9`, `core-libs/storefront/shared/components/popover/index.ts:L7-L11`).

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **HTML del paragrafo non sanitizzato.** `bypassSecurityTrustHtml` rende il contenuto CMS fidato per definizione: chi può scrivere in Backoffice può iniettare script (`core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L49-L51`).
2. **Banner OnPush e link asincrono.** Con `contentPage`, `routerLink` è assegnato dentro una `subscribe` senza `markForCheck`; il video invece chiama `markForCheck` (`core-libs/storefront/cms-components/content/banner/banner.component.ts:L140-L149`, `core-libs/storefront/cms-components/content/video/video.component.ts:L107-L111`). `DA VERIFICARE` a runtime: se `getPage` emette in modo sincrono (pagina già in store) il problema non si manifesta; altrimenti il link appare solo al prossimo ciclo di change detection innescato da altro.
3. **Booleani stringa ovunque.** Il backend manda `'true'`/`'false'`: banner, navigation, video e search box convertono a mano (`core-libs/storefront/cms-components/content/banner/banner.component.ts:L133-L135`, `core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L180-L182`, `core-libs/storefront/cms-components/content/video/video.component.ts:L92-L96`, `core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L250-L252`). Un componente custom che fa `if (data.flag)` sbaglia: `'false'` è truthy.
4. **Navigation: solo la prima entry, solo** `CMSLinkComponent`**.** Altri tipi di entry non generano link (`core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L166-L170`); un nodo senza titolo non riceve `url` anche se l'item ha un link (`core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L174-L177`).
5. `createNavigation` **vs** `getNavigationNode`**.** Il primo aggiunge un nodo radice con `title: data.name`; sostituire uno con l'altro in un componente custom sposta tutto di un livello (`core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L27-L38`).

6. **Accumulo di sottoscrizioni nella navigation UI.** `onSpace` chiama `setupArrowControls()` a ogni pressione e ogni chiamata aggiunge una nuova sottoscrizione a `arrowControls`; ↑/↓ sono ascoltati su `document` da *ogni* istanza di `cx-navigation-ui` (core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L100-L104, core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L256-L275).
7. **Testi hardcoded e config null in** `cx-tab`**.** `getTitle` concatena `'Collapse'`/`'Expand'` in inglese (core-libs/storefront/cms-components/content/tab/tab.component.ts:L239-L242); `handleKeydownEvent` e `getMode` accedono a `this.config.*` senza optional chaining, mentre `ngOnInit` usa `this.config?.openTabs` (core-libs/storefront/cms-components/content/tab/tab.component.ts:L66-L68, core-libs/storefront/cms-components/content/tab/tab.component.ts:L148-L151, core-libs/storefront/cms-components/content/tab/tab.component.ts:L255-L259).
8. `document` **globale.** Breadcrumb, scroll-to-top e `FileDownloadService` usano `document` invece di `WindowRef.document` (core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.component.ts:L37-L42, core-libs/storefront/cms-components/navigation/scroll-to-top/scroll-to-top.component.ts:L85-L86, core-libs/storefront/shared/services/file/file-download.service.ts:L20-L29). DA VERIFICARE il comportamento in SSR: dipende da cosa espone l'ambiente `@angular/platform-server` come `document` globale.
9. `HighlightPipe` **senza match.** Solo con `caseSensitive = false`: se il termine non c'è, `textStartIndex` vale -1 e viene evidenziato un prefisso di `match.length - 1` caratteri; se il termine è lungo 1, `substring(0, 0)` vale `''` e `replaceAll('', ...)` inserisce uno span tra ogni carattere (core-libs/storefront/cms-components/navigation/search-box/highlight.pipe.ts:L15-L24). Con il default `true` la pipe usa `text.replace(match.trim(), ...)`, che senza match non evidenzia nulla; il search box usa il default e non è affetto (core-libs/storefront/cms-components/navigation/search-box/highlight.pipe.ts:L11-L29, core-libs/storefront/cms-components/navigation/search-box/search-box.component.html:L96-L96).
10. **Icone.** `fontawesomeIconConfig` contiene `PDF_FILE`, che non è nell'enum `ICON_TYPE` (core-libs/storefront/cms-components/misc/icon/fontawesome-icon.config.ts:L49-L49); alcuni simboli (`ARROW_RIGHT`/`ARROW_DOWN`/`ARROW_UP`) hanno doppio spazio e funzionano al primo render perché l'aggiunta delle classi salta le stringhe vuote (core-libs/storefront/cms-components/misc/icon/fontawesome-icon.config.ts:L72-L74, core-libs/storefront/cms-components/misc/icon/icon.component.ts:L121-L125). La rimozione no: cambiare `type` partendo da una di queste icone chiama `renderer.removeClass(host, '')` (core-libs/storefront/cms-components/misc/icon/icon.component.ts:L116-L118), cioè `el.classList.remove('')` (angular:packages/platform-browser/src/dom/dom_renderer.ts:L376-L378), che nel browser lancia una `DOMException` (SyntaxError) per token vuoto.
11. **Getter che creano Observable.** `SiteContextSelectorComponent` espone `items$`/`activeItem$` come getter e chiama `ariaLabel$()` nel template: a ogni ciclo di CD si creano nuovi Observable (core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.component.ts:L60-L82, core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.component.html:L1-L9).
12. `PaginationBuilder` **muta la costante di fallback.** `Object.assign(FALLBACK_PAGINATION_OPTIONS, ...)` scrive dentro l'oggetto di modulo condiviso (core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L370-L375). Inoltre il JSDoc di `addDots` dice "defaults to true" mentre il modello dice false e il codice esce se `!addDots` (core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L104-L115, core-libs/storefront/shared/components/list-navigation/pagination/pagination.model.ts:L59-L66); `PaginationComponent` usato standalone fuori da `PaginationModule` non riceve `defaultPaginationConfig` (core-libs/storefront/shared/components/list-navigation/pagination/pagination.module.ts:L28-L28).
13. **Media.** Con un array usa solo `container[0]` (core-libs/storefront/shared/components/media/media.component.ts:L165-L165); `sortedFormats` è memorizzato al primo accesso, quindi cambiare `mediaFormats` a runtime non ha effetto (core-libs/storefront/shared/components/media/media.service.ts:L178-L189);

`MediaSourcesPipe` è importato in `MediaModule` ma non usato nel template (core-libs/storefront/shared/components/media/media.module.ts:L20-L28); `loading` è inizializzato da un getter che usa `this.mediaService`, possibile solo perché `useDefineForClassFields` è `false` (core-libs/storefront/shared/components/media/media.component.ts:L89-L89, tsconfig.json:L799-L799).

14. **Carousel in SSR.** `WindowRef.resize$` emette `of(null)` sul server e `clientWidth` è 0: `calculateItems` restituisce 1 item per slide (anche con `%`, dove `0/0` dà `NaN || 1`) (core-libs/core/src/window/window-ref.ts:L97-L99, core-libs/storefront/shared/components/carousel/carousel.service.ts:L46-L60). È coerente con la scelta di saltare l'hydration (core-libs/storefront/shared/components/carousel/carousel.component.ts:L57-L65); vedi il capitolo 39.
15. **ItemCounter.** I bottoni usano `aria-disabled` e restano cliccabili: è la sottoscrizione `valueChanges` a riportare il valore nel range; il bottone + non toglie il tabindex a `value >= max` (core-libs/storefront/shared/components/item-counter/item-counter.component.html:L1-L54, core-libs/storefront/shared/components/item-counter/item-counter.component.ts:L95-L101).
16. **FormErrors senza control.** Il setter ignora `null`, ma `ngDoCheck` legge `this.control.touched` senza guard: se `control` non viene mai impostato si ha un errore (core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L74-L77, core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L96-L97).
17. **URL protocol-relative.** `//host/path` non è considerato esterno da `GenericLinkComponentService` e finisce in `routerLink` (core-libs/storefront/shared/components/generic-link/generic-link-component.service.ts:L16-L37).
18. **Popover.** La direttiva sottoscrive `eventSubject` senza disiscriversi e usa `ComponentFactoryResolver` deprecato (core-libs/storefront/shared/components/popover/popover.directive.ts:L167-L182, core-libs/storefront/shared/components/popover/popover.directive.ts:L223-L235); il componente ricalcola la posizione a ogni `ngAfterViewChecked` (core-libs/storefront/shared/components/popover/popover.component.ts:L265-L267); trap e autofocus scattano solo con `appendToBody` (core-libs/storefront/shared/components/popover/popover.service.ts:L22-L29).
19. **CSV e download.** L'export non fa escape delle `"` interne né `URL.revokeObjectURL`; l'import fa uno split ingenuo e rimuove tutte le virgolette (core-libs/storefront/shared/services/file/csv/export-csv-file.service.ts:L27-L51, core-libs/storefront/shared/services/file/csv/import-csv-file.service.ts:L86-L87); senza `fileName` il download imposta `download="undefined"` (core-libs/storefront/shared/services/file/file-download.service.ts:L24-L24).
20. `emailValidator` **con valore vuoto.** Restituisce `cxInvalidEmail` anche per stringa vuota; il ramo `!email.length` è irraggiungibile (core-libs/storefront/shared/utils/validators/custom-form-validators.ts:L37-L43).
21. `SupplementHashAnchorsPipe` **e SSR.** Usa `WindowRef.location.href`, che sul server lancia se `SERVER_REQUEST_URL` manca (core-libs/storefront/shared/pipes/suplement-hash-anchors/supplement-hash-anchors.pipe.ts:L28-L34, core-libs/core/src/window/window-ref.ts:L70-L89). DA VERIFICARE l'effetto pratico della purezza della pipe (href non ricalcolato se cambia solo l'URL): dipende dal riuso dell'istanza del componente tra navigazioni.
22. **Card.** Gli output emettono costanti numeriche (delete=1 … cancel=5), non l'indice della card (core-libs/storefront/shared/components/card/card.component.ts:L115-L139).

---

10. Come fare debug

- **Quale componente è mappato su un typeCode?** Cercate il typeCode nell'appendice A03 o con `grep -rn "SimpleResponsiveBannerComponent:" core-libs feature-libs`. Ricordate che uno stesso typeCode può

essere mappato in più librerie: vince l'ultima config applicata (capitolo 11).

- **Cosa riceve il componente?** Mettete un breakpoint in `ComponentDataProvider.get` (`core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L35-L59`): vedete i dati statici e quelli dello store. Per i dati grezzi dello store CMS usate NgRx DevTools (slice `cms`, capitolo 19).
- **Link del banner sbagliato.** Controllate quale campo è valorizzato tra `urlLink`, `contentPage`, `product`, `category`: la cascata si ferma al primo (`core-libs/storefront/cms-components/content/banner/banner.component.ts:L137-L161`). Se il link è esterno ma finisce in `routerLink`, verificate il prefisso con le regex di `GenericLinkComponentService` (`core-libs/storefront/shared/components/generic-link/generic-link-component.service.ts:L16-L37`).
- **Voce di menu mancante.** Verificate in DevTools che l'azione `LoadCmsNavigationItems` sia partita e che la chiave `${itemId}_${itemSuperType}` esista negli entry item; ricordate che solo la prima entry e solo `CMSLinkComponent` contano (`core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L166-L170`). Un nodo senza alcuna proprietà viene scartato (`core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L159-L160`).
- **Immagini rotte.** Ispezionate l'host `cx-media`: la classe `is-missing` indica `src` assente (`core-libs/storefront/shared/components/media/media.component.ts:L155-L208`). Se l'URL è sbagliato confrontate `backend.media.baseUrl`/`backend.occ.baseUrl`/`backend.media.prefix` e il toggle `enableMediaPrefix`; un errore di composizione viene loggato da `getBaseUrl` (`core-libs/storefront/shared/components/media/media.service.ts:L414-L426`).
- **Icona invisibile.** Controllate che `icon.symbols` contenga il tipo e che esista una risorsa del tipo giusto: con Font Awesome di default la risorsa LINK ha `url: ''` e il CSS va caricato dall'app (`core-libs/storefront/cms-components/misc/icon/fontawesome-icon.config.ts:L79-L84`).
- **Tab in modalità sbagliata.** Verificate `config.mode` e `config.breakpoint`: senza nessuno dei due la modalità è sempre `TAB` (`core-libs/storefront/cms-components/content/tab/tab.component.ts:L254-L268`).
- **Comportamenti a11y diversi tra ambienti.** Quasi tutte le differenze di tastiera (banner, navigation, carousel, paginazione) dipendono dai feature toggle elencati nella sezione 6: verificate la config `features` dell'app.
- **Test unitari.** Importate `IconTestingModule` per evitare `IconLoaderService` reale (`core-libs/storefront/cms-components/misc/icon/testing/icon-testing.module.ts:L10-L50`); strategie di test nel capitolo 44.

---

11. Domande di autoverifica

1. **Perché `SimpleBannerComponent` e `SimpleResponsiveBannerComponent` producono lo stesso DOM?** Perché `BannerModule` li mappa entrambi (insieme a `BannerComponent`) sulla stessa classe `BannerComponent`. — `core-libs/storefront/cms-components/content/banner/banner.module.ts:L31-L44`
2. **Se `CategoryNavigationComponent` arriva dal backend con `resetMenuOnClose: false`, quale valore vede il componente?** `false`: `ComponentDataProvider.get` emette prima i dati statici (`true`) e poi `{...static, ...backend}`, dove il backend sovrascrive. — `core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L44-L51`, `core-libs/storefront/cms-components/navigation/category-navigation/category-navigation.module.ts:L21-L30`
3. **Come fa `SiteContextSelectorComponent` a sapere se è un selettore di lingua o di valuta quando arriva dal CMS?** Il mapping fornisce per ogni istanza un `SiteContextComponentService` con `CmsComponentData`; `getContext` legge `data.context` (LANGUAGE/CURRENCY) e `getInjectedService` risolve il service con `injector.get(contextServiceMap[ctx])`. — `core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.module.ts:L36-L45`, `core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L91-L125`

4. **Un nodo di navigazione ha due entry: la prima è un `CMSParagraphComponent`, la seconda un `CMSLinkComponent`. Il nodo avrà un URL?** No: `populateNavigationNode` passa solo `entries[0]` a `populateLink`, che considera solo `itemType === 'CMSLinkComponent'`. — core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L146-L148, core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L166-L170
5. **Perché il `cx-carousel` non partecipa all'hydration e quanti item per slide calcola in SSR?** È marcato `ngSkipHydration` per differenze CSR/SSR; sul server `clientWidth` è 0 e `calculateItems` restituisce `Math.floor(...) || 1`, cioè 1. — core-libs/storefront/shared/components/carousel/carousel.component.ts:L57-L65, core-libs/storefront/shared/components/carousel/carousel.service.ts:L46-L60
6. **Quando `cx-media` usa `loading="eager"`?** Quando `fetchPriority === ImageFetchPriority.HIGH` (es. passato dal contesto LCP del banner); altrimenti usa `loading` o `imageLoadingStrategy` (default `LAZY`). — core-libs/storefront/shared/components/media/media.component.ts:L217-L222, core-libs/storefront/shared/components/media/media.service.ts:L163-L168
7. **Un link CMS `//cdn.example.com/file.pdf` viene reso con `href` o con `routerLink`?** Con `routerLink`: `isExternalUrl` riconosce solo `http://`, `https://`, `mailto:`, `tel:`. — core-libs/storefront/shared/components/generic-link/generic-link-component.service.ts:L16-L37
8. **Perché `FormErrorsComponent` implementa `ngDoCheck`?** Perché il passaggio a `touched` non emette su `statusChanges`; `ngDoCheck` confronta lo stato precedente e chiama `markForCheck`, differito con `setTimeout` quando `aria-live` è `polite`. — core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L96-L112
9. **Con quale meccanismo `TabParagraphContainerComponent` evita un loop infinito nel calcolo dei parametri dei titoli?** `tabCompLoaded` aggiorna `tabTitleParams$` solo se l'uid del figlio non è già nella `Map`. — core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L202-L213
10. **Che problema c'è nel getter `config` di `PaginationBuilder`?** Usa `Object.assign(FALLBACK_PAGINATION_OPTIONS, ...)`, che muta la costante di modulo invece di crearne una copia. — core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L370-L375

TAVOLA 45.1 **4.1 Render di un banner CMS**

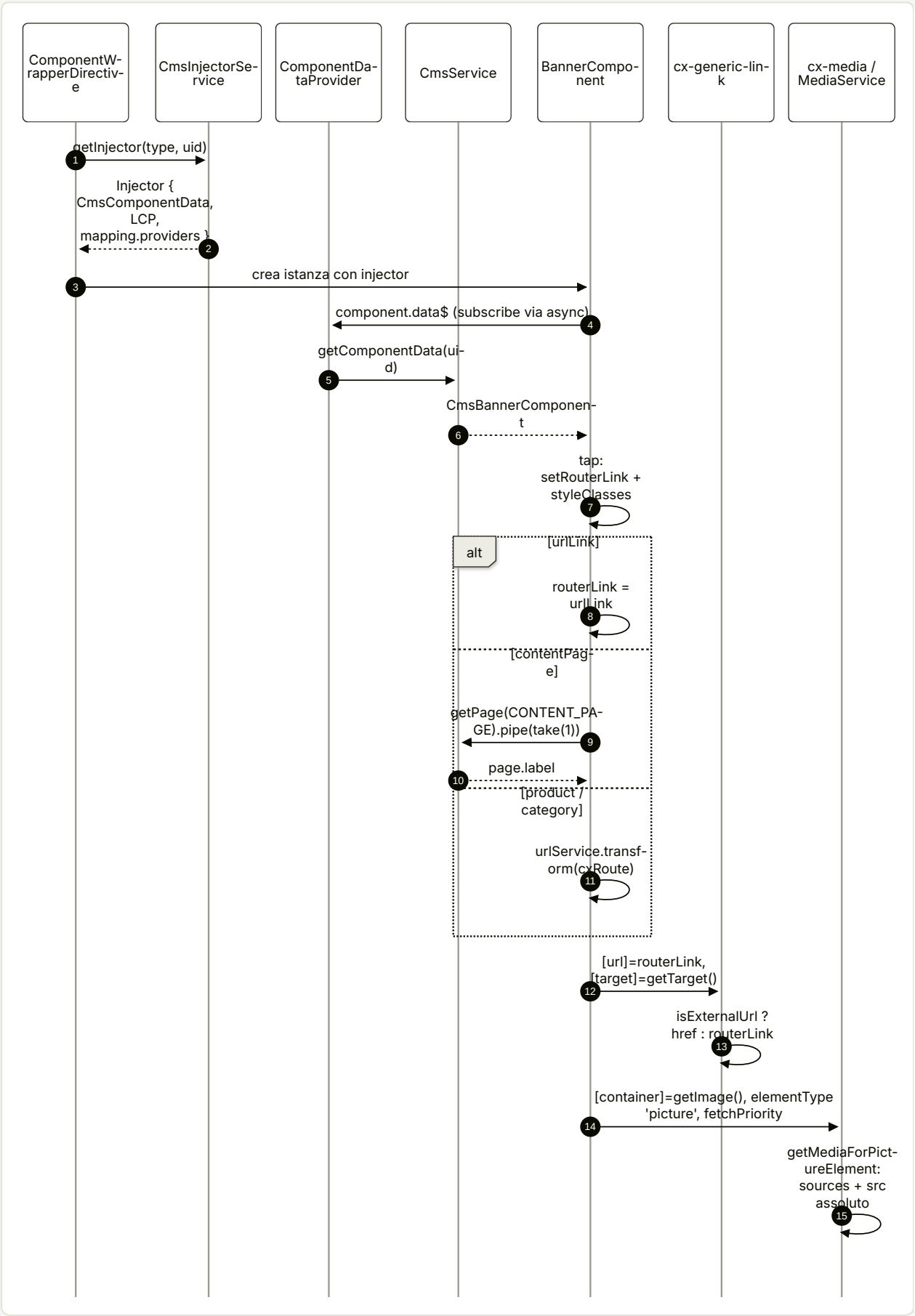
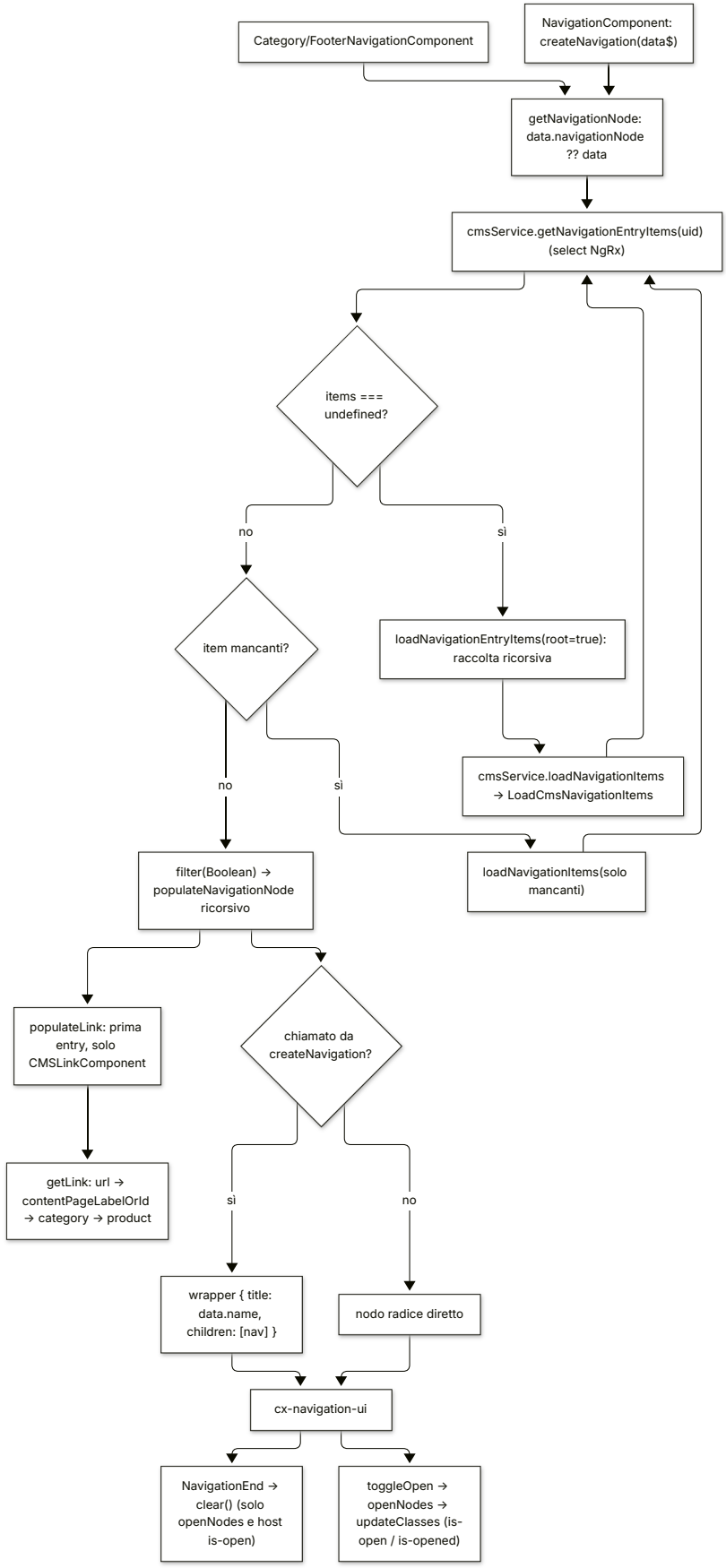


TAVOLA 45.2 **4.2 Costruzione dell'albero di navigazione**



46

Utility, window e modelli di dominio

Gli "attrezzi" di `@spartacus/core`: `ConverterService`, `resolveApplicable`, `backOff`, normalizzazione degli errori.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Ogni capitolo precedente ha incontrato, quasi sempre senza fermarcisi, una manciata di "attrezzi" che `@spartacus/core` mette a disposizione di tutto il resto del framework: il `ConverterService` che normalizza le risposte OCC, `resolveApplicable` che sceglie l'handler giusto tra molti multi-provider, l'operatore `backOff` che ritenta le chiamate fallite, `tryNormalizeHttpError` che rende serializzabile un errore HTTP prima di metterlo in una action NgRx, `WindowRef` che permette di toccare `window` senza esplodere in SSR, e infine le decine di interfacce (`Product`, `Address`, `CmsComponent`, `BaseSite`, …) che danno forma ai dati di dominio. Questo capitolo li raccoglie tutti e li smonta pezzo per pezzo.

Le tre cartelle coinvolte sono riesportate per intero dal `public_api.ts` di core (`core-libs/core/public_api.ts:L26-L36`):

- `core-libs/core/src/util/` — un barrel di 23 moduli (`core-libs/core/src/util/index.ts:L7-L29`) che spazia da pattern architetturali (Applicable, Converter, Command/Query) a operatori RxJS, encoding, regex, script loader, token SSR, helper di test;
- `core-libs/core/src/window/` — il solo `WindowRef` (`core-libs/core/src/window/index.ts:L7-L7`);
- `core-libs/core/src/model/` — 14 file di modelli di dominio (`core-libs/core/src/model/index.ts:L7-L20`), estendibili dalle feature lib tramite *module augmentation* TypeScript.

A questi si aggiungono due utility di `@spartacus/storefront` (`getAddressNumbers` e `FormUtils`), riesportate da `core-libs/storefront/utils/index.ts:L7-L9` tramite `core-libs/storefront/public_api.ts:L30-L30`.

Perché esistono come utility trasversali? Perché risolvono problemi che si ripetono in decine di feature:

PROBLEMA RICORRENTE	UTILITY	QUANTI LA USANO
Scegliere tra più implementazioni registrate come multi-provider	`resolveApplicable`	`PageMetaService`, `HttpErrorInterceptor`, `I18nextBackendService`, `ComponentHandlerService`, `LaunchDialogService` (`core-libs/core/src/cms/facade/page-meta.service.ts:L137-L137`, `core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L62-L64`)
Trasformare payload backend ↔ UI in modo componibile	`ConverterService` + `Converter`	tutti gli adapter OCC (es. `feature-libs/order/occ/adapters/occ-order.adapter.ts:L57-L65`)
Rendere un errore HTTP serializzabile per NgRx	`tryNormalizeHttpError`	113 file non-spec (conteggio grep, dossier F66)
Accedere a `window`/storage/location in modo SSR-safe	`WindowRef`	121 file non-spec (conteggio grep, dossier F94)
Estendere i tipi di dominio senza fork	`declare module '@spartacus/core'`	88 file non-spec (90 blocchi `declare module '@spartacus/core'`)

Due argomenti della cartella `util/` sono trattati altrove e qui solo richiamati: il pattern Command & Query (`core-libs/core/src/util/command-query/index.ts:L7-L8`) in 20 — Facade, command & query, eventi, e `deepMerge` — che, nonostante il nome, **non** sta in `util/` ma in `core-libs/core/src/config/utils/deep-merge.ts:L18-L18` — in 11 — Configurazione. `DirectiveStateTransferService`, pur riesportato dal barrel delle utility di storefront (`core-libs/storefront/utils/index.ts:L8-L8`), appartiene a 39 — SSR ottimizzato.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/util/index.ts	barrel pubblico di util (23 export; **esclude** glob-utils)
core-libs/core/src/util/applicable.ts	Priority, Applicable, resolveApplicable
core-libs/core/src/util/converter.service.ts	interfaccia Converter e ConverterService (catena di converter con cache per token)
core-libs/core/src/util/combined-injector.ts	CombinedInjector (interno, usato da LazyModulesService)
core-libs/core/src/util/create-from.ts	createFrom (istanze di classi evento da dati)
core-libs/core/src/util/glob-utils.ts	glob → regex (interno, derivato dal service worker Angular)
core-libs/core/src/util/glob.service.ts	GlobService.getValidator (pubblico, usato dal routing)
core-libs/core/src/util/base64.ts	encodeBase64 / decodeBase64 UTF-8, opzione URL-safe
core-libs/core/src/util/http-params-uri.encoder.ts	HttpParamsURIEncoder (codec con encodeURIComponent puro)
core-libs/core/src/util/is-key-invalid.ts	guard anti prototype pollution (lancia)
core-libs/core/src/util/normalize-empty.ts	normalizeEmpty: '' → undefined
core-libs/core/src/util/regex-pattern.ts	costanti regex email/password
core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts	JavaRegExpConverter (regex Java dei urlPatterns OCC → JS)
core-libs/core/src/util/java-reg-exp-converter/index.ts	barrel
core-libs/core/src/util/occ-http-error-constants.ts	enum OccHttpErrorType, OccHttpErrorReason (aumentabili)
core-libs/core/src/util/occ-http-error-handlers.ts	predicati isJaloError, isServerError, isAuthorizationError + costanti di retry
core-libs/core/src/util/try-normalize-http-error.ts	tryNormalizeHttpError (HttpErrorResponse → HttpErrorModel)
core-libs/core/src/util/rxjs/index.ts	barrel rxjs (esclude buffer-debounce-time)
core-libs/core/src/util/rxjs/back-off.ts	backOff (retry esponenziale)
core-libs/core/src/util/rxjs/buffer-debounce-time.ts	bufferDebounceTime (interno)
core-libs/core/src/util/rxjs/get-last-value-sync.ts	getLastValueSync
core-libs/core/src/util/rxjs/unite-latest.ts	uniteLatest (combineLatest con emissioni raggruppate)
core-libs/core/src/util/rxjs/withdraw-on.ts	withdrawOn (ri-sottoscrizione su notifier)
core-libs/core/src/util/rxjs/wrap-into-observable.ts	wrapIntoObservable
core-libs/core/src/util/script-loader.service.ts	ScriptLoader, ScriptPlacement
core-libs/core/src/util/ssr.tokens.ts	token SERVER_REQUEST_URL, SERVER_REQUEST_ORIGIN
core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts	classe base per notifiche cross-tab via BroadcastChannel
core-libs/core/src/util/browser-tab-notification/index.ts	barrel
core-libs/core/src/util/type-guards.ts	isNotUndefined, isNotNullable

PATH	RUOLO
core-libs/core/src/util/type-utils.ts	tipo `RequiredPick`
core-libs/core/src/util/object-comparison-utils.ts	`ObjectComparisonUtils` (shallow/deep equal)
core-libs/core/src/util/time-utils.ts	`TimeUtils` (offset timezone, conversioni data/datetime)
core-libs/core/src/util/testing-time-utils.ts	`TestingTimeUtils` (mock di `Date.prototype` per i test)
core-libs/core/src/util/command-query/command.service.ts, query.service.ts, index.ts	Command & Query (cap. 20)
core-libs/core/src/window/window-ref.ts	`WindowRef`
core-libs/core/src/window/index.ts	barrel
core-libs/core/src/model/index.ts	barrel dei modelli (**esclude** `unused.model.ts`)
core-libs/core/src/model/misc.model.ts	tipi trasversali: `User`, `Currency`, `Language`, `BaseSite`, `PaginationModel`, `ErrorModel`, `HttpErrorModel`, …
core-libs/core/src/model/product.model.ts	`Product` e sottotipi, enum `VariantType`, `PriceType`, `ProductTypes`
core-libs/core/src/model/product-search.model.ts	`Facet`, `Breadcrumb`, `ProductSearchPage`, `ClearSearch`
core-libs/core/src/model/image.model.ts	`Images`, `ImageGroup`, `Image`, `ImageType`
core-libs/core/src/model/cms.model.ts	`CmsComponent`, `PageType` e interfacce dei componenti CMS standard
core-libs/core/src/model/address.model.ts	`Country`, `Region`, `Address`, `AddressValidation`, `CountryType`
core-libs/core/src/model/consent.model.ts	consensi, `ANONYMOUS_CONSENTS_HEADER`
core-libs/core/src/model/org-unit.model.ts	modelli B2B (`B2BUnit`, `B2BUser`, `CostCenter`, ruoli)
core-libs/core/src/model/payment.model.ts	`CardType`, `PaymentDetails`
core-libs/core/src/model/point-of-service.model.ts	`PointOfService`, orari di apertura
core-libs/core/src/model/customer-coupon.model.ts	coupon cliente
core-libs/core/src/model/notification-preference.model.ts	preferenze di notifica
core-libs/core/src/model/product-interest.model.ts	interessi di prodotto (back in stock)
core-libs/core/src/model/scoped-data.ts	`ScopedData<T>`
core-libs/core/src/model/unused.model.ts	tipi **non** esportati (`Catalog`, `Pagination`, `Sort`, `ProductList`, …)
core-libs/storefront/utils/index.ts	barrel delle utility di storefront
core-libs/storefront/utils/address-number-utils.ts	`getAddressNumbers`
core-libs/storefront/utils/form-utils.ts	namespace `FormUtils`

3. Come funziona passo per passo

3.1 La superficie pubblica e ciò che ne resta fuori

Prima di entrare nei singoli meccanismi conviene sapere **cosa è importabile** da `@spartacus/core`, perché il barrel di `util/` non espone tutto ciò che c'è nella cartella:

1. `public_api.ts` di core fa `export *` di `src/model/index`, `src/util/index` e `src/window/index` (core-libs/core/public_api.ts:L26-L36).
2. `util/index.ts` elenca 23 moduli (core-libs/core/src/util/index.ts:L7-L29). Manca `glob-utils`: `globToRegex`, `getGlobMatcher` e `processGlobPatterns` sono interni, è pubblico solo `GlobService` (core-libs/core/src/util/index.ts:L13-L13). Manca anche `combined-injector`: `CombinedInjector` non è riesportato da alcun barrel (dossier Q1).
3. `util/rxjs/index.ts` esporta `back-off`, `get-last-value-sync`, `unite-latest`, `withdraw-on`, `wrap-into-observable`, ma **non** `buffer-debounce-time` (core-libs/core/src/util/rxjs/index.ts:L7-L11). `bufferDebounceTime` è importato con path relativo solo dagli effect `ComponentsEffects`, `ProductEffects`, `ProductSearchByCategoryEffects` e `ProductSearchByCodeEffects` (core-libs/core/src/cms/store/effects/components.effect.ts:L17-L17, core-libs/core/src/product/store/effects/product.effect.ts:L16-L16, core-libs/core/src/product/store/effects/product-search-by-category.effect.ts:L14-L14, core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L16-L16).
4. `model/index.ts` esporta 14 file ma non `unused.model.ts` (core-libs/core/src/model/index.ts:L7-L20); quest'ultimo è importato solo da `customer-coupon.model.ts` e `product-interest.model.ts` per `Pagination` e `Sort` (core-libs/core/src/model/customer-coupon.model.ts:L8-L8, core-libs/core/src/model/product-interest.model.ts:L8-L8).

Le conseguenze per chi estende (non si può importare per nome ciò che non è nel barrel) sono discusse in 54 — API pubblica vs interna.

3.2 Applicable e `resolveApplicable`: scegliere il miglior candidato

Molti extension point di Spartacus sono **multi-provider**: più classi registrate sullo stesso token, di cui a runtime ne serve una sola. Il contratto è minimale:

1. L'enum `Priority` fissa quattro livelli: `FALLBACK = -50`, `LOW = -10`, `NORMAL = 0`, `HIGH = 10` (core-libs/core/src/util/applicable.ts:L7-L24).
2. L'interfaccia `Applicable` ha due metodi **entrambi opzionali**: `hasMatch?(...params)` e `getPriority?(...params): Priority | number` (core-libs/core/src/util/applicable.ts:L32-L45). Il tipo di ritorno `number` consente priorità intermedie (es. `Priority.NORMAL + 1`).
3. `resolveApplicable(applicables, matchParams, priorityParams)` filtra prima i candidati: passa chi non ha `hasMatch` (match implicito) o il cui `hasMatch(...matchParams)` è `true`; un array `null`/`undefined` è tollerato grazie a `applicables ?? []` (core-libs/core/src/util/applicable.ts:L63-L70).
4. Con 0 o 1 match restituisce subito `matchedApplicables[0]` — cioè il match o `undefined` — **senza** chiamare `getPriority` (core-libs/core/src/util/applicable.ts:L72-L74).
5. Con 2+ match esegue un `reduce`: chi non ha `getPriority` vale `Priority.NORMAL`; il candidato corrente viene scartato solo se `lastPriority > currPriority`, quindi a parità di priorità **vince l'ultimo**, cioè il provider registrato più tardi (core-libs/core/src/util/applicable.ts:L76-L86). Il JSDoc lo dice esplicitamente: "the applicable that was provided later wins" (core-libs/core/src/util/applicable.ts:L47-L62).

Gli usi concreti: `PageMetaService` sceglie il `PageMetaResolver` passando la pagina sia come match che come priority param (core-libs/core/src/cms/facade/page-meta.service.ts:L137-L137); `HttpErrorInterceptor` sceglie l'`HttpErrorHandler` per la risposta (core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L62-L64, vedi 41 — Errori HTTP); lo stesso schema compare in `I18nextBackendService`, `ComponentHandlerService` e `LaunchDialogService` (core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.service.ts:L34-L36, core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L52-L52, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L184-L184).

3.3 Converter e `ConverterService`: normalize e serialize a catena

Il livello OCC (vedi 21 — Livello OCC) non consegna alla UI i payload grezzi del backend: li fa passare per converter registrati come multi-provider su un `InjectionToken`. Il motore è `ConverterService`.

1. **Contratto.** `Converter<SOURCE, TARGET>` ha un unico metodo `convert(source, target?)`. Il JSDoc distingue *Normalize* (backend → UI) e *Serialize* (UI → backend) e prevede che i converter si "impilino" per applicare personalizzazioni disaccoppiate (core-libs/core/src/util/converter.service.ts:L13-L30).
2. **Servizio.** `ConverterService` è `providedIn: 'root'`, implementa `OnDestroy` e dipende da `UnifiedInjector` (core-libs/core/src/util/converter.service.ts:L32-L38), l'aggregatore di injector root + lazy descritto in 12 — Feature module e lazy loading.
3. **Cache e invalidazione.** La cache è una `Map<InjectionToken<Converter>, Converter[]>` privata (core-libs/core/src/util/converter.service.ts:L47-L50). Nel costruttore il servizio sottoscrive `unifiedInjector.injectors$` e svuota la cache a ogni nuovo injector, cioè ogni volta che viene caricato un modulo lazy che potrebbe portare converter aggiuntivi (core-libs/core/src/util/converter.service.ts:L39-L44). `ngOnDestroy` chiude la sottoscrizione (core-libs/core/src/util/converter.service.ts:L139-L141).
4. **Lookup.** `getConverters(token)` legge i multi-provider con `getLastValueSync(this.unifiedInjector.getMulti(token))` e mette in cache solo se il valore è definito (core-libs/core/src/util/converter.service.ts:L52-L65). `UnifiedInjector.getMulti` scarta le emissioni con array vuoto e accumula con `scan` le istanze di tutti gli injector; se trova un provider non-multi lancia `Multi-providers mixed with single providers` (core-libs/core/src/lazy-loading/unified-injector.ts:L75-L95).
5. **Presenza.** `hasConverters` è vero solo per un array non vuoto (core-libs/core/src/util/converter.service.ts:L70-L75).
6. **API di uso.**
   - `pipeable(token)` restituisce `map(convertSource)` se ci sono converter, altrimenti un operatore identità che fa solo un cast di tipo (core-libs/core/src/util/converter.service.ts:L80-L88);
   - `pipeableMany(token)` è l'analogo per array, via `convertMany` (core-libs/core/src/util/converter.service.ts:L93-L101);
   - `convert(source, token)` restituisce `source` invariato se non ci sono converter (core-libs/core/src/util/converter.service.ts:L106-L112);
   - `convertMany` converte solo se ci sono converter **e** `sources` è un array, altrimenti restituisce l'input (core-libs/core/src/util/converter.service.ts:L117-L128).
7. **La catena.** `convertSource` fa un `reduce` sui converter partendo da `target = undefined`: ogni converter riceve **lo stesso** `source` e il `target` prodotto dal precedente (core-libs/core/src/util/converter.service.ts:L130-L137). È il *populator pattern*: il primo converter crea l'oggetto UI, i successivi lo arricchiscono.

Uso tipico in un adapter: `catchError → tryNormalizeHttpError`, `backOff({ shouldRetry: isJaloError })`, poi `this.converter.pipeable(ORDER_NORMALIZER)` (feature-libs/order/occ/adapters/occ-order.adapter.ts:L57-L65). La ricetta per scrivere un converter custom è in 52 — Ricette OCC.

3.4 Errori HTTP OCC: normalizzazione, predicati e retry

Questa è la seconda metà della pipeline tipica di un adapter. Il "chi gestisce l'errore dopo" è in 41 — Errori HTTP, error handling, logger e global message; qui interessa come l'errore viene **preparato**.

1. **Perché normalizzare.** Il JSDoc di `tryNormalizeHttpError` dichiara lo scopo: incorporare in modo sicuro gli errori HTTP nei payload delle action NgRx, eliminando le parti non serializzabili di `HttpErrorResponse` (core-libs/core/src/util/try-normalize-http-error.ts:L12-L26). La funzione restituisce l'errore normalizzato o, in mancanza, l'originale: `normalizeHttpError(error, logger) ?? error`.
2. **Il modello.** `HttpErrorModel` è una `class` (unico tipo non-interfaccia di `misc.model.ts`), il che permette a `normalizeHttpError` di riconoscerla con `instanceof` (motivazione non documentata nel sorgente) (core-libs/core/src/model/misc.model.ts:L100-L106, core-libs/core/src/util/try-normalize-http-error.ts:L33-L35). I suoi `details` sono `ErrorModel[]` con `message`, `reason`, `subject`, `subjectType`, `type` (core-libs/core/src/model/misc.model.ts:L92-L98).
3. **Conversione.** `normalizeHttpError` restituisce invariato un `HttpErrorModel`; da un `HttpErrorResponse` crea un `HttpErrorModel` copiando `message`, `status`, `statusText`, `url` (core-libs/core/src/util/try-normalize-http-error.ts:L33-L42).
4. **Dettagli backend.** I `details` vengono da `error.error.errors` (formato errori OCC) oppure, se `error.error.error` è una stringa (formato OAuth), da `{ type: error, message: error_description }` (core-libs/core/src/util/try-normalize-http-error.ts:L44-L54).
5. **Altri errori.** Per qualunque altro tipo, in dev mode logga con `logger.error(...)` e restituisce `undefined`, così `tryNormalizeHttpError` restituisce l'errore originale (core-libs/core/src/util/try-normalize-http-error.ts:L59-L65).
6. **Predicati.** `isJaloError` verifica `err.details?.[0]?.type === 'JaloObjectNoLongerValidError'` (core-libs/core/src/util/occ-http-error-handlers.ts:L12-L14); `isServerError` è vero per status 500..511, `isAuthorizationError` per 401 (core-libs/core/src/util/occ-http-error-handlers.ts:L19-L28). Le costanti `DEFAULT_SERVER_ERROR_RETRIES_COUNT = 2` e `DEFAULT_AUTHORIZATION_ERROR_RETRIES_COUNT = 2` (core-libs/core/src/util/occ-http-error-handlers.ts:L30-L31) sono usate come `maxTries` di un `backOff`: per esempio `OccOrderAdapter` concatena un secondo `backOff({ shouldRetry: isServerError, maxTries: DEFAULT_SERVER_ERROR_RETRIES_COUNT })` dopo quello per Jalo (feature-libs/order/occ/adapters/occ-order.adapter.ts:L92-L102), e il servizio OPF degli hosted fields usa la coppia `isAuthorizationError` / `DEFAULT_AUTHORIZATION_ERROR_RETRIES_COUNT` (integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L119-L122).
7. **Enum aumentabili.** `OccHttpErrorType` (`NotFoundError`, `ClassMismatchError`) e `OccHttpErrorReason` (`notFound`) (core-libs/core/src/util/occ-http-error-constants.ts:L9-L16) stanno nel blocco `AUGMENTABLE_TYPES_START/END` del public_api (core-libs/core/public_api.ts:L54-L57): una feature lib può aggiungere valori di tipo.

3.5 `backOff` e gli altri operatori RxJS

1. `backOff(options)`. Default: `shouldRetry = () => true`, `maxTries = 3`, `delay = 300` ms (core-libs/core/src/util/rxjs/back-off.ts:L39-L42). L'implementazione usa `retry({ delay })` di RxJS: dentro la callback, se `currentRetry > maxTries` o `!shouldRetry(err)` rilancia l'errore, altrimenti restituisce `timer(currentRetry² × delay)` (core-libs/core/src/util/rxjs/back-off.ts:L44-L61). In RxJS 7.8.2 `retry` senza `count` usa `count = Infinity` e passa a `delay` il contatore già incrementato, per cui il primo retry ha `currentRetry = 1` (deps:rxjs-7.8.2/package/dist/cjs/internal/operators/retry.js:L20-L20, deps:rxjs-7.8.2/package/dist/cjs/internal/operators/retry.js:L34-L34, deps:rxjs-7.8.2/package/dist/cjs/internal/operators/retry.js:L46-L46). Con i default si hanno quindi 3 retry, con attese di 300 + 1200 + 2700 = 4200 ms, esattamente come dice il JSDoc di `delay` (core-libs/core/src/util/rxjs/back-off.ts:L26-L26).

2. `bufferDebounceTime(time = 0, scheduler?)` (interno) accumula i valori in un array e li emette in blocco dopo il `debounceTime`, poi azzera il buffer; il buffer vive nella closure creata ogni volta che l'operatore viene applicato a una sorgente (al momento del `pipe`, non della subscription): due subscription allo stesso Observable risultante condividerebbero il buffer (`core-libs/core/src/util/rxjs/buffer-debounce-time.ts:L14-L22`). È ciò che permette agli effect CMS/prodotto di accorpare più richieste di caricamento in una sola chiamata (vedi 15 — CMS core e 24 — Prodotto e ricerca).
3. `getLastValueSync(source)` sottoscrive e disiscrive subito, restituendo l'ultimo valore emesso **in modo sincrono** o `undefined`; il JSDoc lo sconsiglia per l'uso generale (`core-libs/core/src/util/rxjs/get-last-value-sync.ts:L9-L23`). Lo usano `ConverterService.getConverters` (§3.3) e `HttpErrorInterceptor` per leggere gli handler (`core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L62-L64`).
4. `uniteLatest(sources)` è `combineLatest(sources).pipe(audit(() => trigger))` dentro un `defer`: la prima emissione è sincrona, le successive sono raggruppate e schedulate con `asapScheduler` (`core-libs/core/src/util/rxjs/unite-latest.ts:L26-L47`). Il JSDoc spiega lo scopo: se più sorgenti emettono "nello stesso momento", `uniteLatest` emette una sola volta dove `combineLatest` emetterebbe una volta per sorgente (`core-libs/core/src/util/rxjs/unite-latest.ts:L15-L25`). Usato da `PageMetaService` e `ProductLoadingService` (`core-libs/core/src/cms/facade/page-meta.service.ts:L99-L99`, `core-libs/core/src/product/services/product-loading.service.ts:L75-L75`).
5. `withdrawOn(notifier)` riscrive lo stream come `notifier.pipe(startWith(undefined), switchMap(() => source))`: a ogni emissione del notifier la sorgente viene ri-sottoscritta, abbandonando la trasformazione in corso (`core-libs/core/src/util/rxjs/withdraw-on.ts:L19-L27`).
6. `wrapIntoObservable(value)` restituisce l'Observable così com'è, converte un thenable con `from(Promise.resolve(value))`, altrimenti `of(value)`; `isPromise` è duck typing su `then` (`core-libs/core/src/util/rxjs/wrap-into-observable.ts:L17-L36`). `GuardsComposer` lo usa per normalizzare il risultato di `canActivate` (`core-libs/storefront/cms-structure/services/guards-composer.ts:L65-L65`).

3.6 `CombinedInjector` e `createFrom`

1. `CombinedInjector` implementa `Injector` combinando un `mainInjector` con una lista di `complementaryInjectors`; il JSDoc lo indica per istanziare moduli lazy che devono vedere le istanze dei propri moduli di dipendenza (`core-libs/core/src/util/combined-injector.ts:L17-L35`).
2. Se riceve l'opzione `self` restituisce `notFoundValue` se fornito, altrimenti lancia: non supporta dipendenze `@Self` (`core-libs/core/src/util/combined-injector.ts:L44-L51`).
3. La risoluzione procede in tre passi — (a) a livello `self` in ogni complementary injector, in ordine; (b) a tutti i livelli in ogni complementary injector; (c) fallback a `mainInjector.get(token, notFoundValue)` — usando come sentinella l'oggetto privato `NOT_FOUND_SYMBOL = {}` (`core-libs/core/src/util/combined-injector.ts:L15-L15`, `core-libs/core/src/util/combined-injector.ts:L53-L70`).
4. `LazyModulesService` lo crea solo quando ci sono **2+** moduli di dipendenza; con uno usa direttamente il suo injector, con zero il proprio (`core-libs/core/src/lazy-loading/lazy-modules.service.ts:L83-L94`). Il contesto completo è in 12 — Feature module e lazy loading.
5. `createFrom(type, data)` è un one-liner: `Object.assign(new type(), data)`; serve a creare istanze di classi evento (`core-libs/core/src/util/create-from.ts:L15-L17`; eventi in 20).

3.7 Glob: pattern di URL per il routing

1. `glob-utils.ts` porta il copyright Google/Angular: è derivato dal service worker di Angular (`core-libs/core/src/util/glob-utils.ts:L1-L7`).

2. Le regole di escaping trasformano `.` → `\.`, `+` → `\+`, `*` → `[^/]*` e `?` → `[^/]` (o `\?` con `literalQuestionMark`) (core-libs/core/src/util/glob-utils.ts:L9-L24).
3. `globToRegex` lavora per segmenti separati da `/`: `**` diventa `(?:.+\/)?` se non è l'ultimo segmento, `.*` se lo è; gli altri segmenti vengono escapati e uniti con `\/` (core-libs/core/src/util/glob-utils.ts:L38-L64).
4. `processGlobPatterns` marca come negativo un pattern che inizia con `!`, lo rimuove e ancora la regex con `^...$`; chiama `globToRegex(url)` senza `literalQuestionMark` (core-libs/core/src/util/glob-utils.ts:L91-L99).
5. `getGlobMatcher` restituisce un predicato vero se almeno un pattern positivo matcha e nessun negativo matcha (core-libs/core/src/util/glob-utils.ts:L71-L86); `GlobService.getValidator` (root, pubblico) duplica la stessa logica (core-libs/core/src/util/glob.service.ts:L10-L32).
6. Il consumatore è `UrlMatcherService.getFromGlob`, che costruisce un `UrlMatcher` Angular ricomponendo il path completo `/seg1/seg2` e testandolo col validator (core-libs/core/src/routing/services/url-matcher.service.ts:L178-L190; routing in 14).

3.8 Encoding, sicurezza e regex

1. **Base64.** `encodeBase64` codifica UTF-8 con `TextEncoder`, poi `btoa` su una binary string; con `urlSafe` applica l'alfabeto RFC 4648 §5 (core-libs/core/src/util/base64.ts:L16-L29). `toUrlSafe` sostituisce `+`→`-`, `/`→`_` e rimuove il padding `=` (core-libs/core/src/util/base64.ts:L54-L59). `decodeBase64` inverte: ripristina alfabeto e padding, `atob`, poi `TextDecoder` (core-libs/core/src/util/base64.ts:L31-L52). Unico consumatore: `FederatedLoginContextSerializerService`, con `urlSafe: true` (core-libs/core/src/federated-login/services/federated-login-context-serializer.service.ts:L8-L8, core-libs/core/src/federated-login/services/federated-login-context-serializer.service.ts:L27-L27).
2. `HttpParamsURIEncoder` implementa `HttpParameterCodec` con solo `encodeURIComponent`/`decodeURIComponent`, perché l'encoder di default di Angular "de-codifica" alcuni caratteri ammessi da RFC 3986 (core-libs/core/src/util/http-params-uri.encoder.ts:L7-L23). È usato in `OccEndpointsService` per i query param degli URL OCC e negli adapter di pagamento checkout (core-libs/core/src/occ/services/occ-endpoints.service.ts:L126-L126, feature-libs/checkout/base/occ/adapters/occ-checkout-payment.adapter.ts:L203-L203).
3. `isKeyInvalid(key)` non restituisce un boolean: lancia `Error` se la chiave è `__proto__`, `constructor` o `prototype` (core-libs/core/src/util/is-key-invalid.ts:L17-L27). `ProductLoadingService` lo chiama su `productCode` prima di indicizzare la sua cache interna (core-libs/core/src/product/services/product-loading.service.ts:L50-L50, core-libs/core/src/product/services/product-loading.service.ts:L59-L59).
4. `normalizeEmpty` converte `''` in `undefined` per poter usare `??` (core-libs/core/src/util/normalize-empty.ts:L14-L18); lo usa per esempio il riepilogo OPF B2B checkout (integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-review/opf-b2b-checkout-review.component.ts:L149-L149).
5. **Regex.** `regex-pattern.ts` esporta `EMAIL_PATTERN`, `PASSWORD_PATTERN` (maiuscola + cifra + speciale, almeno 6), pattern atomici per i requisiti password, `STRONG_PASSWORD_PATTERN`, `SECURE_PASSWORD_PATTERN` (8-128, nessun carattere ripetuto consecutivo) e `ENDS_WITH_LEGAL_CHARACTER_PATTERN` (core-libs/core/src/util/regex-pattern.ts:L7-L36).

3.9 `JavaRegExpConverter`: dal `BaseSite` OCC all'URL corrente

1. `JavaRegExpConverter` è `providedIn: 'root'` e inietta `LoggerService` con `inject()` (core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L10-L12).
2. La regex `EXTRACT_JAVA_REGEXP_MODIFIERS` estrae un prefisso opzionale `(?modifiers)` di sole lettere minuscole e il resto del pattern (core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.

ts:L14-L29).

3. `toJsRegExp` passa i modificatori Java **così come sono** come flag JS (`new RegExp(jsSyntax, modifiers)`); se il costruttore lancia restituisce `null` e, solo in dev mode, logga un warning (core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L42-L58).
4. Il consumatore è `SiteContextConfigInitializer.isCurrentBaseSite`, che testa l'URL corrente contro gli `urlPatterns` del `BaseSite` restituito da OCC (core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L100-L106; `BaseSite.urlPatterns` in core-libs/core/src/model/misc.model.ts:L116-L133). Il flusso di inizializzazione del site context è in 13 — Site context.

3.10 `ScriptLoader`: script di terze parti, SSR-aware

1. `ScriptPlacement` ha `HEAD = 'head'` e `BODY = 'body'`; `ScriptLoader` è root e inietta `DOCUMENT` e `PLATFORM_ID` (core-libs/core/src/util/script-loader.service.ts:L10-L22).
2. `embedScript` accetta `src`, `params`, `attributes`, `callback`, `errorCallback`, `placement` (default `HEAD`), `disableKeyRestriction` (core-libs/core/src/util/script-loader.service.ts:L36-L53).
3. In SSR lo script **non** viene aggiunto se ci sono `callback` o `errorCallback` (non avrebbero senso sul server); nel browser non viene aggiunto se esiste già uno `script[src="..."]` identico; in SSR il controllo dei duplicati è saltato (core-libs/core/src/util/script-loader.service.ts:L55-L63).
4. Lo script creato ha sempre `async = true` e `defer = true` (core-libs/core/src/util/script-loader.service.ts:L65-L68).
5. Gli `attributes` con chiave `data-*` — o, con `disableKeyRestriction`, qualsiasi chiave che non sia già una proprietà di `HTMLScriptElement` — diventano attributi via `setAttribute`; tutte le altre sono assegnate come proprietà DOM (core-libs/core/src/util/script-loader.service.ts:L70-L82).
6. `callback`/`errorCallback` sono agganciati a `load`/`error` e lo script è appeso a `head` o `body` (core-libs/core/src/util/script-loader.service.ts:L84-L93). `hasScript` usa `querySelector('script[src="..."]')` e `parseParams` costruisce `?k=v&...` con `encodeURI` (core-libs/core/src/util/script-loader.service.ts:L99-L121).

3.11 `WindowRef` e i token SSR

`WindowRef` è il modo canonico di toccare il "mondo browser" in Spartacus. Il funzionamento è interamente guidato da `PLATFORM_ID` e dai due token di `ssr.tokens.ts`:

1. `SERVER_REQUEST_URL` e `SERVER_REQUEST_ORIGIN` sono `InjectionToken<string>` **senza** factory di default (core-libs/core/src/util/ssr.tokens.ts:L12-L21). Lato server li forniscono `provideServer()` (factory basate sulle `ServerOptions`) e `getServerRequestProviders()` (factory sulla `REQUEST` Express) (core-libs/setup/ssr/providers/ssr-providers.ts:L27-L37, core-libs/setup/ssr/providers/ssr-providers.ts:L53-L65; dettagli in 39).
2. `WindowRef` è root; inietta `DOCUMENT`, `PLATFORM_ID` e, come `@Optional`, i due token; espone `readonly document: Document` (core-libs/core/src/window/window-ref.ts:L13-L27). In Angular 21 `DOCUMENT` di `@angular/common` è una riesportazione di quello di `@angular/core` (angular:packages/common/src/common.ts:L116-L116).
3. `isBrowser()` delega a `isPlatformBrowser(platformId)`, che confronta con la stringa `'browser'` (core-libs/core/src/window/window-ref.ts:L33-L35, angular:packages/common/src/platform_id.ts:L9-L9, angular:packages/common/src/platform_id.ts:L16-L18).
4. `nativeWindow` restituisce il `window` globale solo nel browser, altrimenti `undefined` (core-libs/core/src/window/window-ref.ts:L41-L44); `sessionStorage` e `localStorage` sono getter che passano da `nativeWindow` e restituiscono `undefined` in SSR (core-libs/core/src/window/window-ref.ts:L50-L60).

5. `location` nel browser restituisce `document.location`; in SSR restituisce un `Partial<Location>` con soli `href` (da `SERVER_REQUEST_URL`) e `origin` (da `SERVER_REQUEST_ORIGIN`), e **lancia** se uno dei due manca (core-libs/core/src/window/window-ref.ts:L70-L89).
6. `resize$` in SSR emette `of(null)`; nel browser è `fromEvent(window, 'resize')` con `debounceTime(300)`, `startWith({ target: window })` e `distinctUntilChanged()` (core-libs/core/src/window/window-ref.ts:L97-L107). È la sorgente da cui parte il `BreakpointService` (core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L41-L41; vedi 18 — Layout e breakpoint), e lo usano anche carousel e popover (core-libs/storefront/shared/components/carousel/carousel.service.ts:L33-L33, core-libs/storefront/shared/components/popover/popover.component.ts:L239-L239).

3.12 `AbstractBrowserTabNotificationService`: messaggi tra tab

Classe base generica per comunicare tra più tab dello stesso storefront tramite `BroadcastChannel`.

1. **Formato di trasporto.** `TabNotificationWrapper<T>` è `{ baseSite, payload }`; `isTabNotificationWrapperEvent` è un type guard che verifica che `event.data` sia un oggetto con entrambe le chiavi (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L14-L34).
2. **Contratto per le sottoclassi.** Vanno definiti `channelId` (nome del canale) e `isPayloadOfExpectedType`; `isolateBySite` vale `true` di default; la classe inietta `BaseSiteService`, `LoggerService`, `WindowRef`, `DestroyRef` (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L49-L58); `isPayloadOfExpectedType` è dichiarato `abstract` (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L134-L134).
3. **Uscita.** Un `Subject` interno viene avvolto con il base site attivo (`withLatestFrom(getActive())`) e pubblicato con `channel?.postMessage` (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L60-L67).
4. **Entrata.** `notifications$` applica in sequenza: guard del wrapper, `isPayloadOfExpectedType`, estrazione di `data`, confronto col base site attivo, estrazione del `payload` (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L73-L84); `passesSiteIsolation` è `!this.isolateBySite || currentBaseSite === wrapper.baseSite` (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L122-L127).
5. **Avvio.** `listen()` agisce solo nel browser: apre il canale, inoltra i `message` su `inboundEvents$`, sottoscrive lo stream in uscita e registra su `DestroyRef` chiusura e disiscrizione; eventuali eccezioni sono loggate come warning (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L93-L111).
6. **Invio.** `sendNotification` spinge sul `Subject`: se `listen()` non è stato chiamato il payload viene scartato in silenzio, e il JSDoc lo documenta (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L113-L120).
7. **Unica sottoclasse.** `AuthNotificationService` accetta solo stringhe dell'enum `AuthNotificationType` (core-libs/core/src/auth/user-auth/facade/auth-notification.service.ts:L26-L39). Il suo `listen()` è chiamato da un initializer solo se il feature toggle `propagateLogoutToAllTabs` è attivo, e `AuthService.coreLogout` invia `LOGOUT` sotto lo stesso toggle (core-libs/core/src/auth/user-auth/user-auth.module.ts:L85-L92, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L195-L199); la ricezione è gestita in `AuthService.subscribeToAuthNotifications` (core-libs/core/src/auth/user-auth/facade/auth.service.ts:L240-L247). Contesto in 22 — Autenticazione e 42 — Feature toggle.

3.13 Helper vari: type guard, tipi, confronto, tempo

- `isNotUndefined` controlla `typeof value !== 'undefined'`; `isNotNullable` aggiunge `value !== null` e restringe a `NonNullable<T>` (core-libs/core/src/util/type-guards.ts:L7-L13) — utili come predicati di `filter` in RxJS.
- `RequiredPick<T, K>` rende obbligatorie le chiavi `K` lasciando invariate le altre: `Omit<T, K> & Required<Pick<T, K>>` (core-libs/core/src/util/type-utils.ts:L7-L11).
- `ObjectComparisonUtils.shallowEqualObjects` confronta con `===` le chiavi proprie e il loro numero (core-libs/core/src/util/object-comparison-utils.ts:L8-L30); `deepEqualObjects` richiede che entrambi siano `instanceof Object` con lo stesso `constructor` e confronta ricorsivamente le proprietà proprie in entrambe le direzioni (core-libs/core/src/util/object-comparison-utils.ts:L32-L75); `countOfDeepEqualObjects` e `indexOfFirstOccurrence` lavorano su array (core-libs/core/src/util/object-comparison-utils.ts:L77-L92).
- `TimeUtils.getLocalTimezoneOffset(invert?)` produce `±HH:MM` dall'offset del browser, invertendo il segno di `getTimezoneOffset` (core-libs/core/src/util/time-utils.ts:L16-L24); `convertDateToDatetime` aggiunge `T00:00:00` o `T23:59:59` più l'offset, `convertDatetimeToDate` fa il percorso inverso restituendo `YYYY-MM-DD` (core-libs/core/src/util/time-utils.ts:L26-L38).
- `TestingTimeUtils` — esportato nell'API pubblica (core-libs/core/src/util/index.ts:L25-L25) — sostituisce temporaneamente `Date.prototype.toLocaleTimeString` o `getTimezoneOffset`, esegue la callback e ripristina l'originale (core-libs/core/src/util/testing-time-utils.ts:L7-L21).
- In storefront, `getAddressNumbers(address, textPhone, textMobile)` produce la riga dei recapiti: solo mobile se `cellphone === phone`, entrambe le righe separate da `\n` se diversi, solo quella presente, oppure `undefined` (core-libs/storefront/utils/address-number-utils.ts:L12-L33). `FormUtils.deepUpdateValueAndValidity` ricorre su `UntypedFormGroup`/`UntypedFormArray` e chiama `updateValueAndValidity({ onlySelf: true, emitEvent })` su ogni controllo, figli prima del genitore (core-libs/storefront/utils/form-utils.ts:L16-L52).

3.14 I modelli di dominio

I file di `model/` sono quasi solo **tipi**: interfacce TypeScript che descrivono i dati "UI" (cioè già normalizzati dai converter OCC) consumati da facade e componenti.

1. **Opzionalità quasi totale.** Le proprietà sono praticamente tutte opzionali (`?`), con poche eccezioni come `ListModel.ids`, `EntitiesModel.values` (core-libs/core/src/model/misc.model.ts:L60-L71), `CaptchaConfig.enabled` (core-libs/core/src/model/misc.model.ts:L135-L138), `SiteTheme.*` (core-libs/core/src/model/misc.model.ts:L140-L143) e `CmsPickupItemDetails.*` (core-libs/core/src/model/cms.model.ts:L249-L252). La scelta riflette i payload OCC, che variano in base ai `fields` richiesti.
2. `misc.model.ts` definisce i tipi trasversali: `Currency`, `Time`, `GeoPoint`, `Language`, `Principal`, `User`, `ListModel`, `EntitiesModel<T>`, `PaginationModel`, `SortModel`, `Title`, `ErrorModel`, `HttpErrorModel`, `BaseStore`, `BaseSite`, `CaptchaConfig`, `SiteTheme` (core-libs/core/src/model/misc.model.ts:L10-L143). `BaseSite` include `urlPatterns` (regex Java, §3.9), `urlEncodingAttributes`, `requiresAuthentication`, `isolated`, `captchaConfig`, `theme` (core-libs/core/src/model/misc.model.ts:L116-L133). `PaginationModel` e `SortModel` sono i modelli UI di paginazione e ordinamento, distinti da `Pagination`/`Sort` di `unused.model.ts` (core-libs/core/src/model/misc.model.ts:L73-L85, core-libs/core/src/model/unused.model.ts:L33-L43).
3. **Prodotto.** `Product` raccoglie i campi OCC (codice, nome, `slug`, immagini, prezzi, stock, varianti, classificazioni, riferimenti, recensioni, promozioni) più `sapUnit` e `productTypes` (core-libs/core/src/model/product.model.ts:L167-L210); `slug` è documentato come base per gli URL "pretty" della PDP (core-libs/core/src/model/product.model.ts:L181-L186). Enum: `VariantType`, `PriceType`, `VariantQualifier`, `ProductTypes` (core-libs/core/src/model/product.model.ts:L57-L66, core-libs/core/src/model/

product.model.ts:L222-L230, core-libs/core/src/model/product.model.ts:L267-L271). `ProductReferences` e `Images` sono dizionari indicizzati per stringa; `ImageGroup` è indicizzato per formato (core-libs/core/src/model/product.model.ts:L157-L159, core-libs/core/src/model/image.model.ts:L7-L9, core-libs/core/src/model/image.model.ts:L41-L43). `Image.width/height` non vengono dal CMS: sono stati aggiunti per impostare manualmente le dimensioni nel `MediaComponent` (core-libs/core/src/model/image.model.ts:L18-L33). `PointOfServiceStock` ripete i campi di `PointOfService` aggiungendo `stockInfo` invece di estenderlo (core-libs/core/src/model/product.model.ts:L232-L247).

4. **Ricerca.** `Facet` documenta `expanded`, `topValueCount` e `topValues` (duplicato OCC normalizzato in `topValueCount`) (core-libs/core/src/model/product-search.model.ts:L43-L60); `ProductSearchPage` usa `PaginationModel`/`SortModel` e contiene `breadcrumbs`, `facets`, `freeTextSearch`, `keywordRedirectUrl`, `spellingSuggestion` (core-libs/core/src/model/product-search.model.ts:L68-L79); `ClearSearch` ha `clearPageResults` e `clearSearchboxResults` (core-libs/core/src/model/product-search.model.ts:L85-L88).
5. **CMS.** `CmsComponent` è la base dei dati di ogni componente CMS: `uid`, `typeCode`, `name`, `modifiedTime`, `otherProperties`, `composition.inner`, `styleClasses` (core-libs/core/src/model/cms.model.ts:L7-L32). `PageType` ha `ContentPage`, `ProductPage`, `CategoryPage`, `CatalogPage` (core-libs/core/src/model/cms.model.ts:L38-L43). Il file prosegue con le interfacce dei componenti standard che estendono `CmsComponent` — link, search box, paragraph, banner, carousel, navigation, add to cart, video, pickup item details, … (core-libs/core/src/model/cms.model.ts:L45-L252). Molti attributi semanticamente booleani o liste sono tipizzati `string`, riflettendo il payload CMS di OCC (core-libs/core/src/model/cms.model.ts:L88-L92, core-libs/core/src/model/cms.model.ts:L137-L144, core-libs/core/src/model/cms.model.ts:L219-L224). I media di banner e video possono essere singoli o responsive (`desktop`, `mobile`, `tablet`, `widescreen`) (core-libs/core/src/model/cms.model.ts:L106-L117). Come questi dati arrivano ai componenti è in 15 e 16.
6. **Indirizzi, consensi, B2B, pagamenti, punti vendita.** `Address` con `titleCode`, `line1/2`, `town`, `postalCode`, `region`, `country`, `cellphone`, `phone`, `defaultAddress`, `visibleInAddressBook`, `formattedAddress`, … (core-libs/core/src/model/address.model.ts:L38-L67); `CountryType` e `AddressValidation` (core-libs/core/src/model/address.model.ts:L14-L17, core-libs/core/src/model/address.model.ts:L69-L74). `consent.model.ts` definisce consensi, consensi anonimi, `ANONYMOUS_CONSENT_STATUS` e la costante `ANONYMOUS_CONSENTS_HEADER = 'X-Anonymous-Consents'` (core-libs/core/src/model/consent.model.ts:L7-L32), letta dall'interceptor dei consensi anonimi (core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L61-L61). `org-unit.model.ts` definisce `CostCenter`, `B2BUnit`, `B2BUser extends User`, `B2BApprovalProcess`, `B2BUserRole`, `B2BUserRight` (core-libs/core/src/model/org-unit.model.ts:L11-L49). `PaymentDetails` (core-libs/core/src/model/payment.model.ts:L13-L28), `PointOfService` e `OpeningSchedule` (core-libs/core/src/model/point-of-service.model.ts:L11-L49).
7. **Modelli "my account".** `customer-coupon.model.ts`, `notification-preference.model.ts` e `product-interest.model.ts` (core-libs/core/src/model/customer-coupon.model.ts:L10-L36, core-libs/core/src/model/notification-preference.model.ts:L7-L16, core-libs/core/src/model/product-interest.model.ts:L10-L29).
8. `ScopedData<T>` avvolge un `data$` Observable con lo `scope` associato (core-libs/core/src/model/scoped-data.ts:L9-L15): `ScopedProductData` la estende per il caricamento di prodotto per scope (core-libs/core/src/product/connectors/product/scoped-product-data.ts:L13-L13; vedi 24).

3.15 Module augmentation dei modelli

I modelli sono interfacce proprio perché TypeScript permette di **fonderle** (declaration merging) da qualsiasi altro file, a patto di rientrare nel modulo giusto:

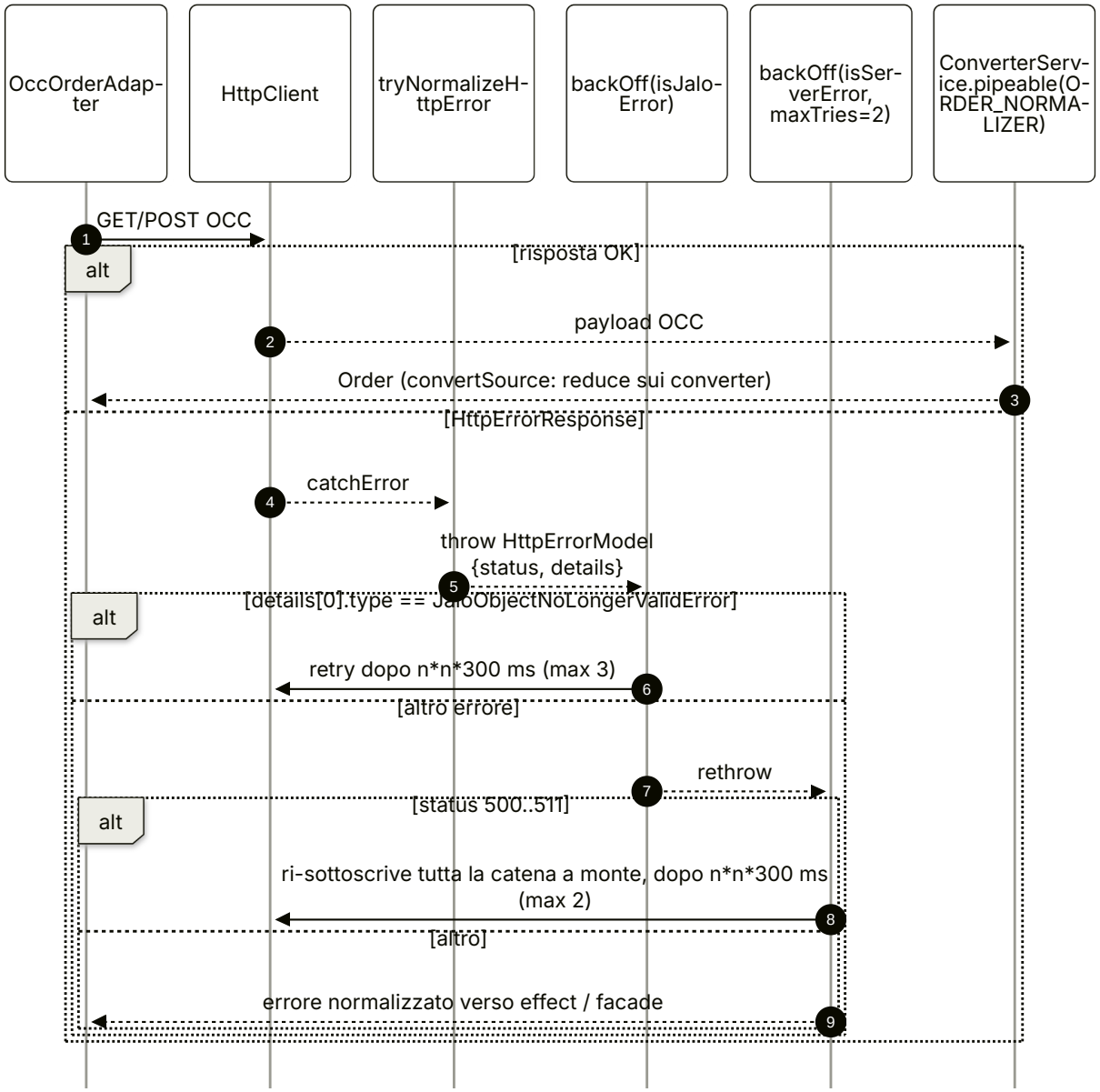
1. Una feature lib scrive `declare module '@spartacus/core' { interface Product { ... } }`; 88 file non-spec del repo lo fanno, per 90 blocchi in tutto (alcuni file ne hanno due, es. `feature-libs/subscription-billing/root/model/subscription-product.model.ts:L98-L98`, `feature-libs/subscription-billing/root/model/subscription-product.model.ts:L105-L105`). Esempi: product-configurator aggiunge a `Product` i campi `configurable` e `configuratorType` (`feature-libs/product-configurator/common/core/model/augmented-core.model.ts:L8-L13`); organization aggiunge a `B2BUnit` `parentOrgUnit`, `approvalProcess`, `administrators`, … (`feature-libs/organization/administration/core/model/augmented-core.model.ts:L9-L25`).
2. Il public_api di core racchiude tra i commenti `AUGMENTABLE_TYPES_START/END` un blocco di re-export espliciti: `AuthToken`, `Config`, `Address`, `B2BUnit`, `B2BUser`, `B2BUserRole`, `CostCenter`, `Facet`, `ProductSearchPage`, `Price`, `Product`, `Stock`, `OccHttpError*`, … (`core-libs/core/public_api.ts:L38-L58`).
3. Esiste un builder `augmented-types` che sposta quel blocco dal d.ts del public API al d.ts radice del pacchetto, come workaround a microsoft/TypeScript#9532 e #18877 (`tools/build-lib/augmented-types/index.ts:L20-L32`, `tools/build-lib/augmented-types/index.ts:L117-L148`). Il target `build` di core, però, usa l'executor `./tools/build-lib:declaration-merging`, che corregge i path relativi dei `declare module '../foo'` nel d.ts appiattito (workaround ng-packagr#3085) (`core-libs/core/project.json:L8-L13`, `tools/build-lib/declaration-merging/index.ts:L22-L28`, `tools/build-lib/declaration-merging/index.ts:L108-L111`).
4. Nessun `project.json` referenzia `augmented-types`, e `declaration-merging` non tocca i delimitatori: alla SHA il blocco è un semplice re-export esplicito, e l'augmentation funziona perché i `declare module '@spartacus/core'` puntano all'entry del pacchetto (`tools/build-lib/declaration-merging/index.ts:L108-L128`, `core-libs/core/project.json:L8-L13`). Nessuno script `.sh`, workflow `.yml` o `project.json` invoca il builder `augmented-types`: compare solo nella sua registrazione (`tools/build-lib/builders.json:L3-L5`, `tools/build-lib/package.json:L5-L5`).
5. Un **enum** si può aumentare solo a livello di tipo: subscription-billing aggiunge `SUBSCRIPTION` a `ProductScope` con `declare module` e poi assegna il valore a runtime con un cast `(ProductScope as any)['SUBSCRIPTION'] = 'subscription'` (`feature-libs/subscription-billing/root/model/subscription-product.model.ts:L98-L110`).

---

4. Diagrammi

4.1 Pipeline di un adapter OCC: errore, retry, conversione

Il diagramma segue l'esempio di `OccOrderAdapter` (`feature-libs/order/occ/adapters/occ-order.adapter.ts:L92-L102`): `tryNormalizeHttpError` (§3.4), due `backOff` in cascata (§3.5) e `pipeable` (§3.3).

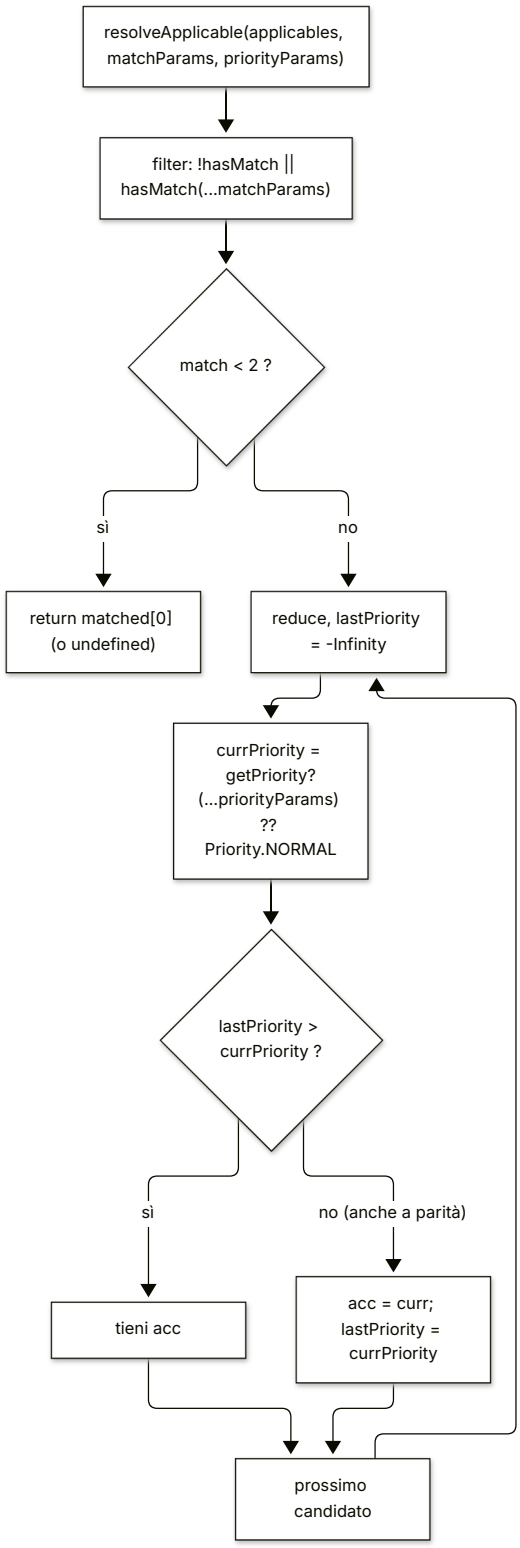


4.2 `ConverterService`: cache, lookup e catena

■ **TAVOLA 46.1** diagramma a tutta pagina, in fondo al capitolo

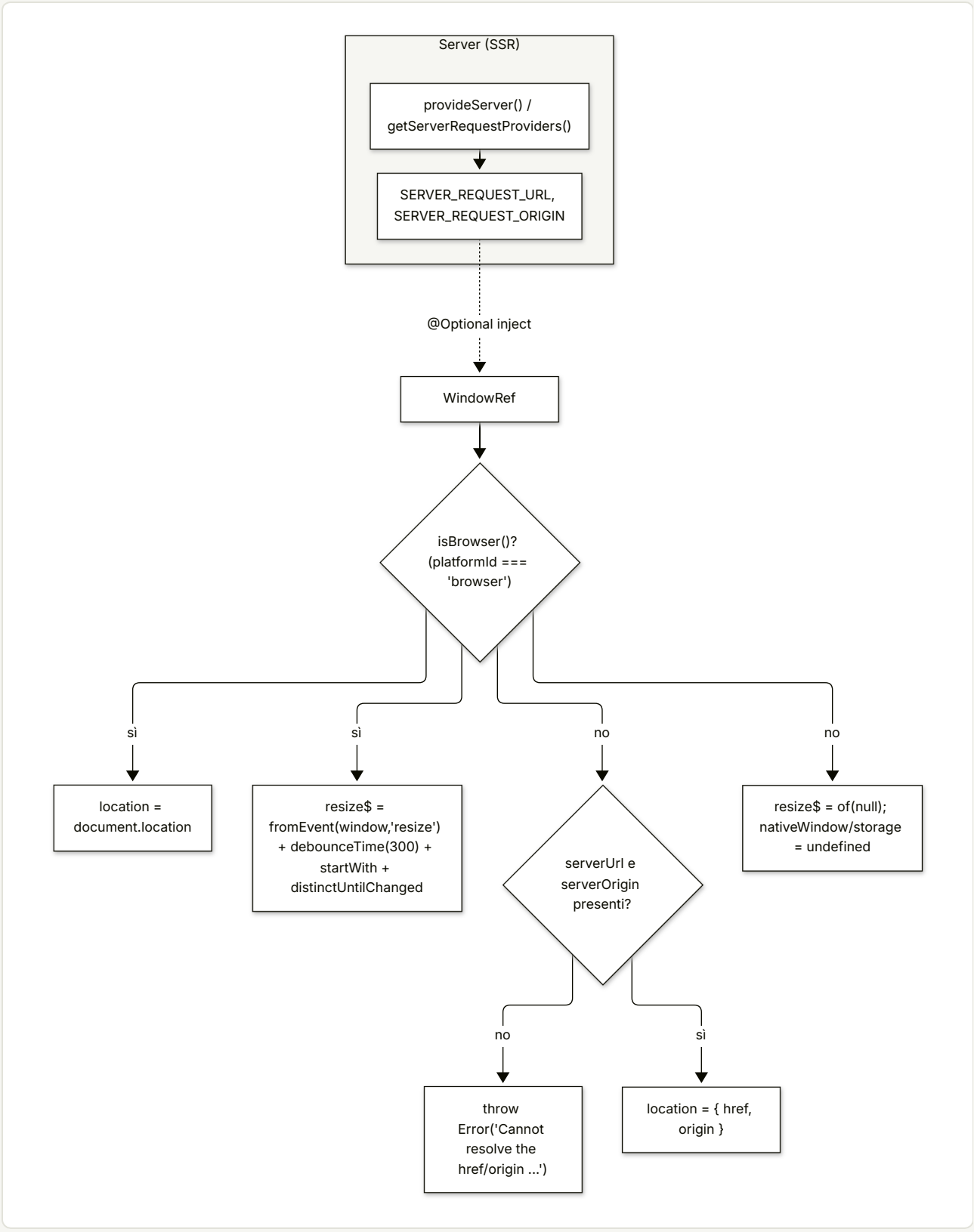
Riferimenti: cache e reset `core-libs/core/src/util/converter.service.ts:L39-L50`; lookup `core-libs/core/src/util/converter.service.ts:L52-L65`; `getMulti` `core-libs/core/src/lazy-loading/unified-injector.ts:L75-L95`; passthrough `core-libs/core/src/util/converter.service.ts:L80-L88`; catena `core-libs/core/src/util/converter.service.ts:L130-L137`.

4.3 `resolveApplicable`



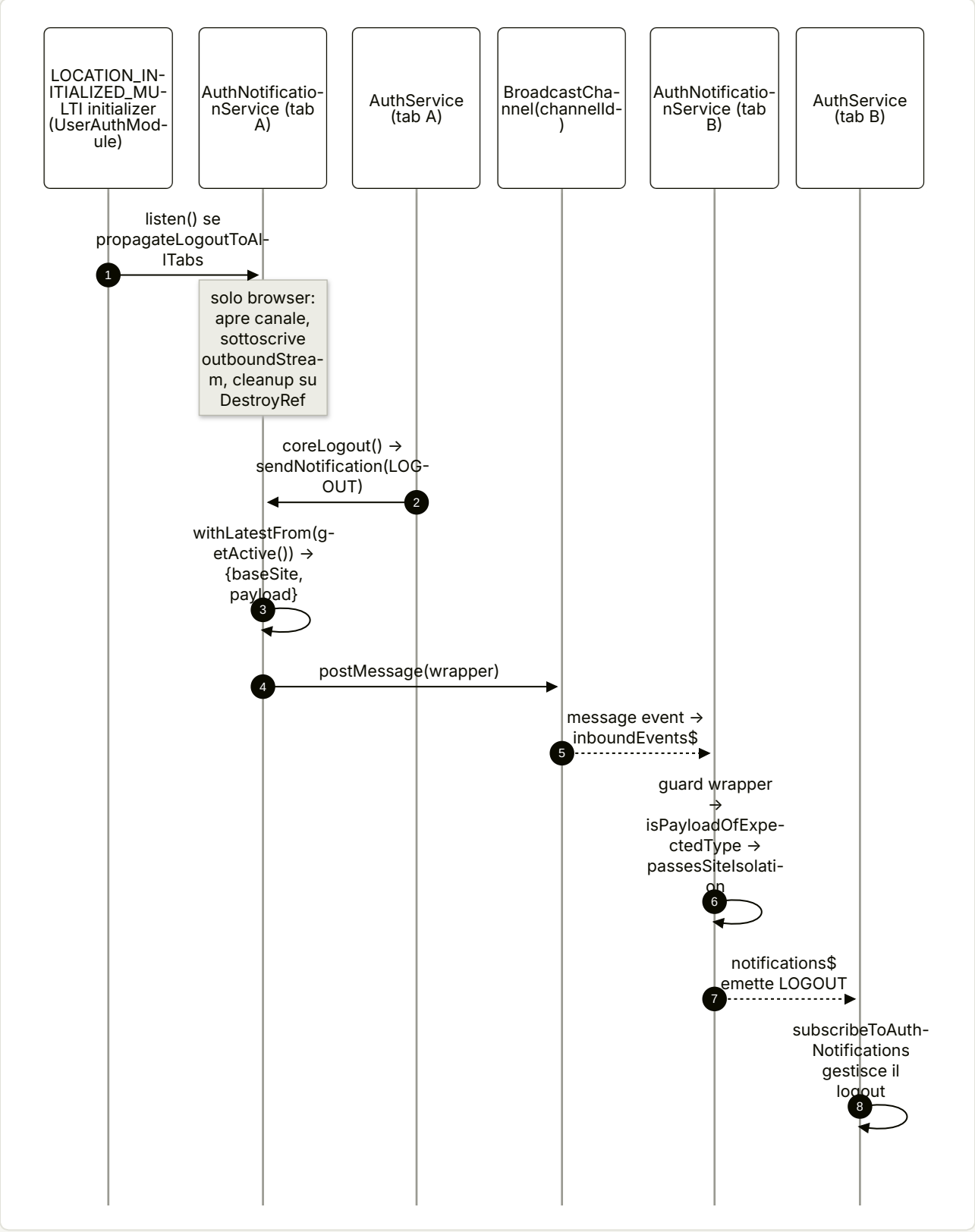
Riferimenti: `core-libs/core/src/util/applicable.ts:L63-L87`.

4.4 `WindowRef.location` e `resize$` tra browser e server



Riferimenti: `core-libs/setup/ssr/providers/ssr-providers.ts:L27-L37`, `core-libs/core/src/window/window-ref.ts:L19-L27`, `core-libs/core/src/window/window-ref.ts:L70-L107`.

4.5 Logout propagato tra tab



Riferimenti: core-libs/core/src/auth/user-auth/user-auth.module.ts:L85-L92, core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L60-L111, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L195-L199, core-libs/core/src/auth/user-auth/facade/auth.service.ts:L240-L247.

5. Estratti di codice reale

5.1 `resolveApplicable`

core-libs/core/src/util/applicable.ts:L63-L87

```ts
export function resolveApplicable<T extends Applicable>(
  applicables: T[],
  matchParams: any[] = [],
  priorityParams: any[] = []
): T | undefined {
  const matchedApplicables = (applicables ?? []).filter(
    (applicable) => !applicable.hasMatch || applicable.hasMatch(...matchParams)
  );

  if (matchedApplicables.length < 2) {
    return matchedApplicables[0];
  }

  let lastPriority = -Infinity;
  return matchedApplicables.reduce((acc: T | undefined, curr: T) => {
    const currPriority = curr.getPriority
      ? curr.getPriority(...priorityParams)
      : Priority.NORMAL;
    if (lastPriority > currPriority) {
      return acc;
    }
    lastPriority = currPriority;
    return curr;
  }, undefined);
}
```

Si noti l'operatore `>` stretto: è ciò che fa vincere l'ultimo a parità di priorità.

5.2 `ConverterService`: reset cache, lookup, catena

core-libs/core/src/util/converter.service.ts:L38-L65

```ts
  constructor(protected unifiedInjector: UnifiedInjector) {
    // Clear cached converters when new injectors appear
    const cacheResetLogic = this.unifiedInjector.injectors$.pipe(
      tap(() => this.converters.clear())
    );

    this.subscriptions.add(cacheResetLogic.subscribe());
  }

  private converters: Map<
```

```
    InjectionToken<Converter<any, any>>,
    Converter<any, any>[]
  > = new Map();

  private getConverters<S, T>(
    injectionToken: InjectionToken<Converter<S, T>>
  ): Converter<S, T>[] | undefined {
    if (!this.converters.has(injectionToken)) {
      const converters = getLastValueSync(
        this.unifiedInjector.getMulti(injectionToken)
      );
      if (converters) {
        this.converters.set(injectionToken, converters);
      }
    }

    return this.converters.get(injectionToken);
  }
```

core-libs/core/src/util/converter.service.ts:L130-L137

```ts
  private convertSource<S, T>(
    source: S,
    injectionToken: InjectionToken<Converter<S, T>>
  ): T {
    return this.getConverters(injectionToken)?.reduce((target, converter) => {
      return converter.convert(source, target);
    }, undefined as any);
  }
```

5.3 tryNormalizeHttpError

core-libs/core/src/util/try-normalize-http-error.ts:L21-L58

```ts
export function tryNormalizeHttpError(
  error: HttpErrorResponse | HttpErrorModel | any,
  logger: LoggerService
): HttpErrorModel | Error {
  return normalizeHttpError(error, logger) ?? error;
}

// CXSPA-11104: Revise type to avoid redundant types - typescript:S6571
function normalizeHttpError(
  error: HttpErrorResponse | HttpErrorModel | any,
  logger: LoggerService
): HttpErrorModel | undefined {
  if (error instanceof HttpErrorModel) {
    return error;
```

```
  }

  if (error instanceof HttpErrorResponse) {
    const normalizedError = new HttpErrorModel();
    normalizedError.message = error.message;
    normalizedError.status = error.status;
    normalizedError.statusText = error.statusText;
    normalizedError.url = error.url;

    // include backend's error details
    if (Array.isArray(error.error.errors)) {
      normalizedError.details = error.error.errors;
    } else if (typeof error.error.error === 'string') {
      normalizedError.details = [
        {
          type: error.error.error,
          message: error.error.error_description,
        },
      ];
    }

    return normalizedError;
  }
```

5.4 backOff

core-libs/core/src/util/rxjs/back-off.ts:L39-L62

TS

```
export function backOff<T>(options?: BackOffOptions): OperatorFunction<T, T> {
  const shouldRetry = options?.shouldRetry ?? (() => true);
  const maxTries = options?.maxTries ?? 3;
  const delay = options?.delay ?? 300;

  return (source$) =>
    source$.pipe(
      // retries the source stream in case of an error.
      retry({
        delay: (attemptError: HttpErrorModel | Error, currentRetry) => {
          // if we've re-tried more than the maxTries, OR
          // if the source error is not the one we want to exponentially retry
          if (currentRetry > maxTries || !shouldRetry(attemptError)) {
            throw attemptError;
          }

          // exponential
          const exponent = currentRetry * currentRetry;
          // back-off
          return timer(exponent * delay);
```

```
        },
      })
    );
}
```

5.5 Due `backOff` in cascata in un adapter reale

feature-libs/order/occ/adapters/occ-order.adapter.ts:L91-L102

```ts
      .pipe(
        catchError((error) => {
          throw tryNormalizeHttpError(error, this.logger);
        }),
        backOff({
          shouldRetry: isJaloError,
        }),
        backOff({
          shouldRetry: isServerError,
          maxTries: DEFAULT_SERVER_ERROR_RETRIES_COUNT,
        }),
        this.converter.pipeable(ORDER_NORMALIZER)
```

5.6 `uniteLatest`

core-libs/core/src/util/rxjs/unite-latest.ts:L26-L47

```ts
export function uniteLatest<R extends Observables>(
  sources: R
): Observable<ReturnTypes<R>> {
  return defer(() => {
    let subNo = 0;
    const trigger = new Observable<void>((subscriber) => {
      const action = () => {
        subscriber.next();
        subscriber.complete();
      };

      if (subNo) {
        asapScheduler.schedule(action);
      } else {
        action();
      }
      subNo++;
    });
```

```
    return combineLatest(sources).pipe(audit(() => trigger));
  }) as any;
}
```

`audit` sottoscrive `trigger` a ogni valore che apre una finestra: la prima volta (`subNo === 0`) il trigger scatta subito, le volte successive scatta su `asapScheduler`, lasciando che le altre sorgenti "sincrone" aggiornino il valore prima dell'emissione.

5.7 `CombinedInjector.get`

core-libs/core/src/util/combined-injector.ts:L43-L71

TS

```
get(token: any, notFoundValue?: any, optional?: InjectOptions): any {
  if (optional && optional.self) {
    if (notFoundValue !== undefined) {
      return notFoundValue;
    }
    throw new Error(
      "CombinedInjector should be used as a parent injector / doesn't support self dependencies"
    );
  }

  for (const injector of this.complementaryInjectors) {
    // First we are resolving providers provided at Self level
    // in all complementary injectors...
    const service = injector.get(token, NOT_FOUND_SYMBOL, { self: true });
    if (service !== NOT_FOUND_SYMBOL) {
      return service;
    }
  }

  for (const injector of this.complementaryInjectors) {
    // next we try to resolve tokens from all levels
    const service = injector.get(token, NOT_FOUND_SYMBOL);
    if (service !== NOT_FOUND_SYMBOL) {
      return service;
    }
  }
  // ...and then fallback to main injector
  return this.mainInjector.get(token, notFoundValue);
}
```

5.8 `WindowRef.location` e `resize$`

core-libs/core/src/window/window-ref.ts:L70-L107

```ts
  get location(): Partial<Location> {
    if (this.isBrowser()) {
      return this.document.location;
    } else {
      if (!this.serverUrl) {
        throw new Error(
          'Cannot resolve the href as the SERVER_REQUEST_URL is undefined'
        );
      }
      if (!this.serverOrigin) {
        throw new Error(
          'Cannot resolve the origin as the SERVER_REQUEST_ORIGIN is undefined'
        );
      }
      return {
        href: this.serverUrl,
        origin: this.serverOrigin,
      };
    }
  }

  /**
   * Returns an observable for the window resize event and emits an event
   * every 300ms in case of resizing. An event is simulated initially.
   *
   * If there's no window object available (i.e. in SSR), a null value is emitted.
   */
  get resize$(): Observable<any> {
    if (!this.nativeWindow) {
      return of(null);
    } else {
      return fromEvent(this.nativeWindow, 'resize').pipe(
        debounceTime(300),
        startWith({ target: this.nativeWindow }),
        distinctUntilChanged()
      );
    }
  }
```

5.9 `ScriptLoader.embedScript` (corpo)

core-libs/core/src/util/script-loader.service.ts:L55-L93

```ts
    const isSSR = isPlatformServer(this.platformId);
    if ((callback || errorCallback) && isSSR) {
      return;
    }
```

```
  const source = params ? src + this.parseParams(params) : src;
  if (!isSSR && this.hasScript(source)) {
    return;
  }

  const script: HTMLScriptElement = this.document.createElement('script');
  script.src = source;
  script.async = true;
  script.defer = true;

  if (attributes) {
    Object.keys(attributes).forEach((key) => {
      // custom attributes
      if (
        key.startsWith('data-') ||
        (disableKeyRestriction && !(key in script))
      ) {
        script.setAttribute(key, attributes[key as keyof object]);
      } else {
        (script as any)[key] = attributes[key as keyof object];
      }
    });
  }

  if (callback) {
    script.addEventListener('load', callback);
  }
  if (errorCallback) {
    script.addEventListener('error', errorCallback);
  }

  placement === ScriptPlacement.HEAD
    ? this.document.head.appendChild(script)
    : this.document.body.appendChild(script);
```

5.10 JavaRegExpConverter.toJsRegExp

core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L42-L58

```ts
toJsRegExp(javaSyntax: string): RegExp | null {
  const parts = javaSyntax.match(this.EXTRACT_JAVA_REGEXP_MODIFIERS);
  if (!parts) {
    return null;
  }
  const [, , modifiers, jsSyntax] = parts;
  try {
    return new RegExp(jsSyntax, modifiers);
  } catch (error) {
```

```
      if (isDevMode()) {
        this.logger.warn(
          `WARNING: Could not convert Java regexp into Javascript. Original regexp: ${javaSyntax}
\nMessage: ${error}`
        );
      }
      return null;
    }
  }
```

5.11 `listen()` e `sendNotification()` della notifica cross-tab

core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L93-L120

TS

```
  listen(): void {
    if (!this.windowRef.isBrowser()) {
      return;
    }
    try {
      this.channel = new BroadcastChannel(this.channelId);
      this.channel.addEventListener('message', (event) =>
        this.inboundEvents$.next(event)
      );
      this.destroyRef.onDestroy(() => this.channel?.close());

      const sub = this.outboundStream.subscribe();
      this.destroyRef.onDestroy(() => sub.unsubscribe());
    } catch (err) {
      this.logger.warn(
        `Could not open ${this.channelId} channel: ${(err as Error)?.message ?? ''}`
      );
    }
  }

  /**
   * Sends `data` to other tabs. No-op when `listen()` was never called
   * (e.g. running on the server, or unsupported browser): the payload
   * is dropped silently.
   */
  sendNotification(data: T): void {
    this.outbound.next(data);
  }
```

5.12 Module augmentation, anche di un enum

feature-libs/subscription-billing/root/model/subscription-product.model.ts:L98-L110

```ts
declare module '@spartacus/core' {
  interface Product {
    sapSubscriptionTerm?: SubscriptionTerm;
    sapPricePlan?: PricePlan;
  }
}

declare module '@spartacus/core' {
  enum ProductScope {
    SUBSCRIPTION = 'subscription',
  }
}
(ProductScope as any)['SUBSCRIPTION'] = 'subscription';
```

6. Configurazione

Il capitolo non definisce chiavi di `Config` (quelle sono in 11). Le "manopole" sono parametri di funzione, default di classe e token:

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
BackOffOptions.maxTries	number	3	core-libs/core/src/util/rxjs/back-off.ts:L23-L24	core-libs/core/src/util/rxjs/back-off.ts:L41-L41
BackOffOptions.delay	number (ms)	300	core-libs/core/src/util/rxjs/back-off.ts:L26-L27	core-libs/core/src/util/rxjs/back-off.ts:L42-L42
BackOffOptions.shouldRetry	(err) => boolean	() => true	core-libs/core/src/util/rxjs/back-off.ts:L21-L21	core-libs/core/src/util/rxjs/back-off.ts:L40-L40
DEFAULT_SERVER_ERROR_RETRIES_COUNT	costante	2	core-libs/core/src/util/occ-http-error-handlers.ts:L30-L30	feature-libs/order/occ/adapters/occ-order.adapter.ts:L98-L101
DEFAULT_AUTHORIZATION_ERROR_RETRIES_COUNT	costante	2	core-libs/core/src/util/occ-http-error-handlers.ts:L31-L31	integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L119-L122
embedScript.placement	ScriptPlacement	HEAD	core-libs/core/src/util/script-loader.service.ts:L42-L42	core-libs/core/src/util/script-loader.service.ts:L91-L93
embedScript.disableKeyRestriction	boolean	undefined	core-libs/core/src/util/script-loader.service.ts:L43-L43	core-libs/core/src/util/script-loader.service.ts:L75-L75
B64EncodingOptions.urlSafe	boolean	undefined	core-libs/core/src/util/base64.ts:L7-L14	core-libs/core/src/util/base64.ts:L25-L25, core-libs/core/src/util/base64.ts:L35-L35

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`globToRegex(…, literalQuestionMark)`	`boolean`	`false`	`core-libs/core/src/util/glob-utils.ts:L38-L38`	`core-libs/core/src/util/glob-utils.ts:L39-L41`
`bufferDebounceTime(time)`	`number`	`0`	`core-libs/core/src/util/rxjs/buffer-debounce-time.ts:L11-L11`	`core-libs/core/src/util/rxjs/buffer-debounce-time.ts:L19-L19`
`isolateBySite` (cross-tab)	`boolean`	`true`	`core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L51-L51`	`core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L126-L126`
debounce di `WindowRef.resize$`	costante	`300` ms	`core-libs/core/src/window/window-ref.ts:L102-L102`	—
`SERVER_REQUEST_URL` / `SERVER_REQUEST_ORIGIN`	`InjectionToken<string>`	nessuno (opzionali in `WindowRef`)	`core-libs/core/src/util/ssr.tokens.ts:L12-L21`	`core-libs/core/src/window/window-ref.ts:L23-L24` (forniti in `core-libs/setup/ssr/providers/ssr-providers.ts:L29-L36`)
feature toggle `propagateLogoutToAllTabs`	`boolean`	vedi 42	—	`core-libs/core/src/auth/user-auth/user-auth.module.ts:L88-L88`

Nota: `OccCartAccessCodeAdapter` usa `maxTries: 2` letterale invece della costante (`feature-libs/cart/base/occ/adapters/occ-cart-access-code.adapter.ts:L42-L45`): cambiare la costante non lo influenza.

---

7. Commenti e note notevoli dal sorgente

- `// TODO(#11133): Consider throwing in SSR` su `nativeWindow`: oggi restituisce `undefined`, ma è stata valutata un'eccezione esplicita (`core-libs/core/src/window/window-ref.ts:L42-L42`).
- `// https://github.com/angular/angular/issues/20351` sul parametro `document: any` del costruttore di `WindowRef`: spiega perché il tipo è `any` e poi viene fatto il cast a `Document` (`core-libs/core/src/window/window-ref.ts:L20-L20`).
- JSDoc di `JavaRegExpConverter.toJsRegExp`: "**CAUTION!** Not all features and modifiers of Java regexps are valid in Javascript!" (`core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L35-L36`).
- JSDoc di `getLastValueSync`: "Should be used with caution, as it's not a legitimate way for getting value from the observable." (`core-libs/core/src/util/rxjs/get-last-value-sync.ts:L13-L15`).
- JSDoc di `backOff`: cita come fonte la guida Angular "practical observable usage / exponential backoff" e il tag `@returns` è troncato ("or the") (`core-libs/core/src/util/rxjs/back-off.ts:L34-L37`).
- JSDoc di `resolveApplicable`: "the applicable that was provided later wins" (`core-libs/core/src/util/applicable.ts:L47-L62`).
- JSDoc di `FormUtils.deepUpdateValueAndValidity`: la funzione potrà essere rimossa quando Angular la implementerà nativamente (angular/angular#6170) (`core-libs/storefront/utils/form-utils.ts:L24-L25`).

- `// CXSPA-11104: Revise type to avoid redundant types - typescript:S6571` su `normalizeHttpError` (core-libs/core/src/util/try-normalize-http-error.ts:L28-L28).
- `// TODO: Simplify with converter` su `AddressValidation.errors`, che ha la forma annidata `{ errors: ErrorModel[] }` (core-libs/core/src/model/address.model.ts:L71-L71).
- `// TODO: Upgrade model when Breadcrumbs will be finally used in project` su `CmsBreadcrumbsComponent` (core-libs/core/src/model/cms.model.ts:L167-L168).
- `// TODO(#8875): Do we need it here?` su `EntitiesModel` (core-libs/core/src/model/misc.model.ts:L66-L66) e `// TODO(#8876): Should we simplify the models only for the fields required by the B2B checkout?` in testa ai modelli B2B (core-libs/core/src/model/org-unit.model.ts:L10-L10).
- `// CHECK SONAR` in coda a `object-comparison-utils.ts` (core-libs/core/src/util/object-comparison-utils.ts:L95-L95).
- Il commento in `fromUrlSafe` dice "'_' to '+'" ma il codice converte _ in `/`: il codice è corretto, il commento no (core-libs/core/src/util/base64.ts:L46-L47).
- `glob-utils.ts` porta il copyright "Google LLC. http://angular.io/license" (core-libs/core/src/util/glob-utils.ts:L1-L7).
- Nessun `@deprecated` né `FIXME` nei 57 file del capitolo (verifica grep riportata nel dossier).

8. Extension point

EXTENSION POINT	COME SI ESTENDE	DEFINIZIONE	EXPORT PUBBLICO	RICETTA
`Applicable` / `Priority` / `resolveApplicable`	implementare `hasMatch`/`getPriority` in un nuovo provider (es. resolver, handler); usare `Priority.HIGH` o `NORMAL + n` per vincere sul default	core-libs/core/src/util/applicable.ts:L7-L87	core-libs/core/src/util/index.ts:L7-L7	50 — Ricette DI, 52 — meta resolver
`Converter<S,T>` su un `InjectionToken` (multi)	aggiungere un converter con `multi: true`: riceverà `(source, target)` dopo quelli già registrati	core-libs/core/src/util/converter.service.ts:L21-L35	core-libs/core/src/util/index.ts:L11-L11	52 — Ricette OCC, converter
`ConverterService`	servizio root; normalmente si usa, non si sostituisce	core-libs/core/src/util/converter.service.ts:L32-L142	idem	52
Predicati `isJaloError` / `isServerError` / `isAuthorizationError` + `backOff`	riusabili negli adapter custom come `shouldRetry`	core-libs/core/src/util/occ-http-error-handlers.ts:L12-L31, core-libs/core/src/util/rxjs/back-off.ts:L39-L62	core-libs/core/src/util/index.ts:L7-L29, core-libs/core/src/util/rxjs/index.ts:L7-L7	52

EXTENSION POINT	COME SI ESTENDE	DEFINIZIONE	EXPORT PUBBLICO	RICETTA
`AbstractBrowserTabNotificationService<T>`	sottoclasse con `channelId`, `isPayloadOfExpectedType`, opz. `isolateBySite = false`; chiamare `listen()` (es. da un initializer)	core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L49-L135	core-libs/core/src/util/index.ts:L9-L9	50
`WindowRef`	iniettarlo invece di usare `window`; sostituibile/mockabile via DI	core-libs/core/src/window/window-ref.ts:L13-L108	core-libs/core/src/window/index.ts:L7-L7	50
`SERVER_REQUEST_URL`, `SERVER_REQUEST_ORIGIN`	fornirli in un server custom se non si usa `provideServer()`	core-libs/core/src/util/ssr.tokens.ts:L12-L21	core-libs/core/src/util/index.ts:L24-L24	50
`ScriptLoader`	classe root con metodi `public` (`parseParams` è `private`)	core-libs/core/src/util/script-loader.service.ts:L15-L122	core-libs/core/src/util/index.ts:L23-L23	50
`JavaRegExpConverter`, `GlobService`	servizi root sostituibili via DI	core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L10-L59, core-libs/core/src/util/glob.service.ts:L10-L33	core-libs/core/src/util/index.ts:L13-L16	51 — route custom
Modelli aumentabili (`Address`, `B2BUnit`, `B2BUser`, `Product`, `Price`, `Stock`, `Facet`, `ProductSearchPage`, `OccHttpErrorType`, …)	`declare module '@spartacus/core' { interface Product { ... } }`	core-libs/core/public_api.ts:L38-L58	blocco `AUGMENTABLE_TYPES_START/END`	52 — campi OCC aggiuntivi, 53
Tutti gli altri modelli di `model/`	stessa tecnica di augmentation	core-libs/core/src/model/index.ts:L7-L20	core-libs/core/public_api.ts:L26-L26	52
`FormUtils`, `getAddressNumbers`	utility da riusare nei componenti custom	core-libs/storefront/utils/form-utils.ts:L16-L53, core-libs/storefront/utils/address-number-utils.ts:L12-L33	core-libs/storefront/utils/index.ts:L7-L9	51
Non extension point: `CombinedInjector`, `glob-utils`, `bufferDebounceTime`, `unused.model.ts`	interni: non importabili da `@spartacus/core`	core-libs/core/src/util/combined-injector.ts:L27-L72, core-libs/core/src/util/rxjs/index.ts:L7-L11	—	54 — API pubblica vs interna

Uno schizzo (illustrativo, non presente nel repo) di augmentation + converter, usando solo simboli pubblici (`Product`, `Converter`, `Occ`, `PRODUCT_NORMALIZER`); la ricetta completa con la registrazione del provider è in 52:

```ts
// esempio illustrativo
declare module '@spartacus/core' {
  interface Product {
    myBadge?: string;
  }
}

@Injectable()
export class MyBadgeNormalizer implements Converter<Occ.Product, Product> {
  convert(source: Occ.Product, target?: Product): Product {
    target = target ?? { ...(source as any) }; // il primo della catena riceve undefined
    target.myBadge = (source as any).badge;
    return target;
  }
}
```

DA VERIFICARE: lo schizzo non è compilato (node_modules assente); `Occ` e `PRODUCT_NORMALIZER` sono simboli pubblici di core ma la loro firma va controllata nella ricetta 52.

---

9. Trappole, casi limite, comportamenti controintuitivi

1. **A parità di priorità vince l'ultimo.** `resolveApplicable` scarta il corrente solo se `lastPriority > currPriority`: l'ordine dei provider conta, e un handler custom con la stessa priorità del default vince solo se registrato dopo (core-libs/core/src/util/applicable.ts:L81-L85).
2. **Con un solo match** `getPriority` **non viene neppure chiamato** (core-libs/core/src/util/applicable.ts:L72-L74): un breakpoint in `getPriority` non scatta.
3. **Nessuna cache dei "nessun converter".** `getConverters` mette in cache solo valori definiti; per un token senza provider `getMulti` non emette (filtra gli array vuoti) e ogni chiamata rifà il lookup (core-libs/core/src/util/converter.service.ts:L55-L62, core-libs/core/src/lazy-loading/unified-injector.ts:L82-L92).
4. `pipeable` **decide al momento della creazione dell'operatore**, non a ogni emissione: se la pipe è costruita prima che un modulo lazy porti il primo converter del token, quella pipe resta passthrough (inferenza dalla struttura di core-libs/core/src/util/converter.service.ts:L83-L87).
5. **Il primo converter riceve** `target === undefined` (core-libs/core/src/util/converter.service.ts:L134-L136): i converter "aggiuntivi" devono gestirlo, perché a priori non sanno se sono primi.
6. `backOff` **ritenta qualsiasi errore per default** (`shouldRetry = () => true`, core-libs/core/src/util/rxjs/back-off.ts:L40-L40), compresi 4xx non transitori. Inoltre con due `backOff` in cascata (§5.5) il secondo ri-sottoscrive tutta la catena a monte, primo `backOff` compreso, il cui contatore riparte (inferenza dalla semantica di `retry`).
7. `tryNormalizeHttpError` **e** `error.error` **nullo.** Il codice accede a `error.error.errors` senza optional chaining (core-libs/core/src/util/try-normalize-http-error.ts:L45-L47): un `HttpErrorResponse` con `error` `null`/`undefined` farebbe lanciare un `TypeError`. DA VERIFICARE: inferenza dal codice, non confermata da spec o esecuzione.

8. **Fuori dev mode niente log**: per errori non HTTP `tryNormalizeHttpError` restituisce silenziosamente l'originale (core-libs/core/src/util/try-normalize-http-error.ts:L59-L65), che potrebbe non essere serializzabile.
9. `getLastValueSync` **su sorgenti asincrone restituisce** `undefined` (core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23).
10. **Simboli non pubblici.** `bufferDebounceTime`, `globToRegex`/`getGlobMatcher`/`processGlobPatterns` e `CombinedInjector` non si importano da `@spartacus/core` (core-libs/core/src/util/rxjs/index.ts:L7-L11, core-libs/core/src/util/index.ts:L7-L29).
11. `CombinedInjector` **e** `@Self`**.** Con `{ self: true }` e senza `notFoundValue` lancia (core-libs/core/src/util/combined-injector.ts:L44-L51); inoltre ignora le altre `InjectOptions` (`optional`, `skipSelf`, `host`) e al main injector passa solo `notFoundValue` (core-libs/core/src/util/combined-injector.ts:L43-L70).
12. `isKeyInvalid` **lancia invece di restituire un boolean**, nonostante il nome (core-libs/core/src/util/is-key-invalid.ts:L17-L27).
13. **Glob "sporchi".** `globToRegex` escapa solo `.`, `+`, `*`, `?`: `(`, `[`, `$`, `^`, `|`, `{` restano metacaratteri attivi (core-libs/core/src/util/glob-utils.ts:L12-L24). Nei pattern di routing `?` è sempre un wildcard di un carattere non `/` (core-libs/core/src/util/glob-utils.ts:L97-L97). E `GlobService.getValidator` duplica `getGlobMatcher`: modificarne uno non cambia l'altro (core-libs/core/src/util/glob.service.ts:L17-L32, core-libs/core/src/util/glob-utils.ts:L71-L86).
14. **Modificatori Java senza equivalente JS** (es. `x`) fanno fallire `new RegExp` → `null`, e il pattern viene trattato come non matchante perché `?.test` dà `undefined` (core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L47-L57, core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L102-L102). In produzione non c'è neanche il warning.
15. `ScriptLoader` **in SSR** aggiunge lo script al DOM server (quindi all'HTML) se non ci sono callback, e salta il controllo duplicati: chiamate ripetute possono duplicarlo (core-libs/core/src/util/script-loader.service.ts:L55-L63). `parseParams` usa `encodeURI`, che non codifica `&`, `=`, `?`, `#` (core-libs/core/src/util/script-loader.service.ts:L114-L117); `hasScript` interpola `src` in un selettore CSS senza escaping (core-libs/core/src/util/script-loader.service.ts:L99-L101).
16. `WindowRef.location` **in SSR** lancia se i token non sono forniti e restituisce solo `href` e `origin`: `pathname`, `search`, `hash` sono `undefined` (core-libs/core/src/window/window-ref.ts:L74-L88).
17. `resize$` **e** `distinctUntilChanged()`: confronta per riferimento e gli `Event` sono sempre oggetti nuovi, quindi di fatto non filtra nulla dopo il valore iniziale (inferenza da core-libs/core/src/window/window-ref.ts:L101-L105).
18. **Notifiche cross-tab perse.** `sendNotification` prima di `listen()` (o in SSR) scarta silenziosamente; e poiché `outboundStream` usa `withLatestFrom(getActive())`, anche i messaggi inviati prima che il base site attivo abbia emesso vanno persi (seconda parte: inferenza dalla semantica di `withLatestFrom`) (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L60-L67, core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L113-L120).
19. `deepEqualObjects` **e oggetti senza proprietà proprie.** Due `Date` diverse risultano uguali; `NaN` non è uguale a se stesso (inferenza da core-libs/core/src/util/object-comparison-utils.ts:L46-L75). `indexOfFirstOccurrence` restituisce `undefined`, non `-1` (core-libs/core/src/util/object-comparison-utils.ts:L86-L92).
20. **Offset timezone frazionari negativi.** Per UTC−03:30 (`offset = -210`) `Math.floor(-3.5) = -4` e `-210 % 60 = -30`, producendo `-04:-30` (core-libs/core/src/util/time-utils.ts:L17-L23). DA VERIFICARE: inferenza aritmetica, non confermata da spec.
21. `TestingTimeUtils` **senza** `try/finally`: se la callback lancia, `Date.prototype` resta patchato per i test successivi; ed è esportato nell'API pubblica di produzione (core-libs/core/src/util/testing-time-utils.

ts:L8-L20, core-libs/core/src/util/index.ts:L25-L25).

22. `FormUtils` **e form tipizzati.** Usa `instanceof UntypedFormGroup/UntypedFormArray`, ma in Angular 21 quei valori runtime sono gli stessi costruttori di `FormGroup`/`FormArray`, quindi funziona anche con form tipizzati e `FormRecord` (core-libs/storefront/utils/form-utils.ts:L37-L46, angular:packages/forms/src/model/form_group.ts:L706-L706, angular:packages/forms/src/model/form_group.ts:L739-L739, angular:packages/forms/src/model/form_array.ts:L584-L584).
23. **Tipi pubblici che usano tipi non pubblici.** `CustomerCouponSearchResult` e `ProductInterestSearchResult` usano `Pagination`/`Sort` di `unused.model.ts`, che non sono importabili per nome (core-libs/core/src/model/index.ts:L7-L20, core-libs/core/src/model/customer-coupon.model.ts:L27-L31). Non confonderli con `PaginationModel`/`SortModel`.
24. **Dipendenza circolare di tipo**: `misc.model.ts` importa `B2BUnit` da `'.'` (il barrel), mentre `org-unit.model.ts` importa `User` da `misc.model` (core-libs/core/src/model/misc.model.ts:L7-L7, core-libs/core/src/model/org-unit.model.ts:L8-L8).
25. **Augmentation di enum = solo tipo**: serve anche l'assegnazione runtime (feature-libs/subscription-billing/root/model/subscription-product.model.ts:L105-L110).
26. `CmsBannerCarouselEffect.CURTAIN` **vale** `'CURTAINX'` (core-libs/core/src/model/cms.model.ts:L128-L128); e molti flag CMS sono stringhe `'true'`/`'false'`, con `otherProperties?: any` (core-libs/core/src/model/cms.model.ts:L10-L10, core-libs/core/src/model/cms.model.ts:L219-L224): nei componenti servono conversioni esplicite.

---

10. Come fare debug

- **"Il mio converter non viene applicato".** Verificare che sia registrato con `multi: true` sullo **stesso** token (altrimenti `getMulti` lancia `Multi-providers mixed with single providers`, core-libs/core/src/lazy-loading/unified-injector.ts:L75-L95). Se è in un modulo lazy, controllare quando viene costruita la pipe che usa `pipeable` (trappola 4). Un breakpoint in `convertSource` (core-libs/core/src/util/converter.service.ts:L134-L136) mostra l'elenco effettivo e l'ordine dei converter.
- **"Viene scelto l'handler/resolver sbagliato".** Breakpoint nel `reduce` di `resolveApplicable` (core-libs/core/src/util/applicable.ts:L77-L86) e ispezione di `matchedApplicables`: si vede subito chi passa `hasMatch` e con quale priorità. Ricordare che a parità vince l'ultimo.
- **Retry inattesi o assenti.** Nella tab Network del browser, i retry di `backOff` appaiono come richieste ripetute a 300 ms, 1,2 s, 2,7 s. Verificare il `shouldRetry` passato e lo `status`/`details` dell'`HttpErrorModel` (core-libs/core/src/util/occ-http-error-handlers.ts:L12-L28).
- **Errore non serializzabile nello store.** Se nelle action NgRx compare un `HttpErrorResponse` invece di un `HttpErrorModel`, l'adapter non usa `tryNormalizeHttpError`; in dev mode, se è stato passato un errore non HTTP, la console mostra "Error passed to tryNormalizeHttpError is not HttpErrorResponse instance" (core-libs/core/src/util/try-normalize-http-error.ts:L59-L65).
- **SSR: "Cannot resolve the href as the SERVER_REQUEST_URL is undefined".** Il server non fornisce i token: usare `provideServer()` o `getServerRequestProviders()` (core-libs/setup/ssr/providers/ssr-providers.ts:L27-L37, core-libs/setup/ssr/providers/ssr-providers.ts:L53-L65).
- **Base site non riconosciuto dall'URL.** In dev mode cercare in console "WARNING: Could not convert Java regexp into Javascript" (core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L51-L55); poi provare a mano `new RegExp(...)` con i flag estratti.
- **Notifiche cross-tab.** Verificare che `listen()` sia stato chiamato (per l'auth: toggle `propagateLogoutToAllTabs`, core-libs/core/src/auth/user-auth/user-auth.module.ts:L85-L92), cercare il

warning "Could not open ... channel" (core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L106-L109) e confrontare il `baseSite` del wrapper con quello attivo nell'altra tab.

- **Script duplicati nell'HTML SSR.** Ispezionare il sorgente della pagina (non il DOM): `ScriptLoader` in SSR non deduplica (trappola 15).
- **Test che dipendono dal fuso orario.** Usare `TestingTimeUtils.fakeDateTimezoneOffset` (core-libs/core/src/util/testing-time-utils.ts:L7-L21), ricordando che non ripristina se la callback lancia.

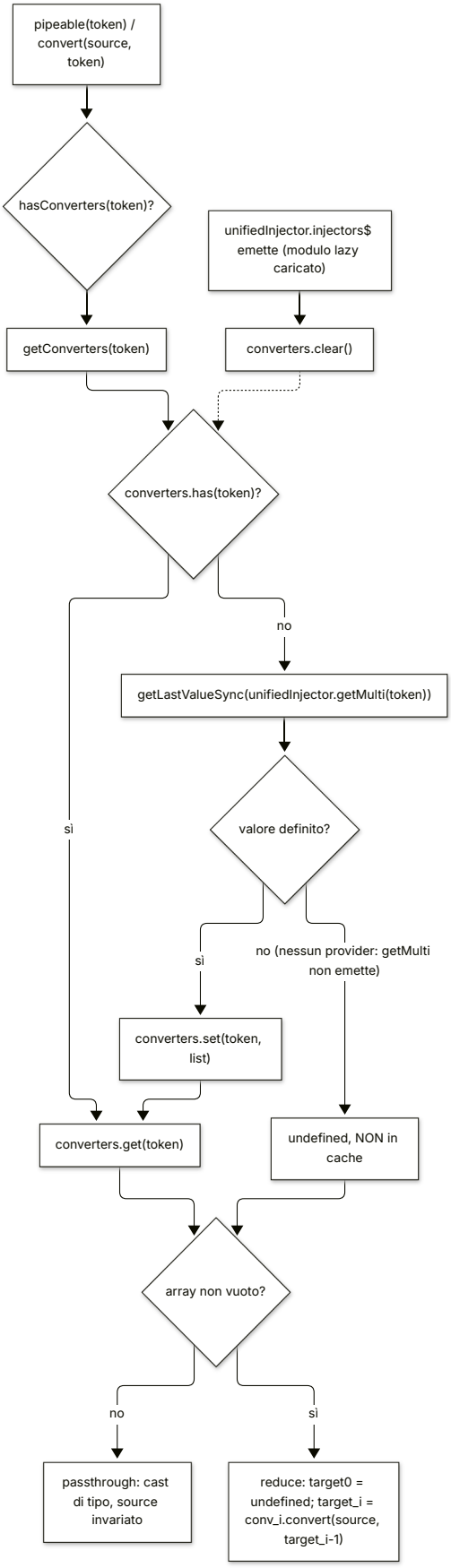
---

11. Domande di autoverifica

1. **Due `PageMetaResolver` matchano la stessa pagina ed entrambi restituiscono `Priority.NORMAL`. Quale viene scelto?** L'ultimo nell'ordine dei provider: il `reduce` scarta il corrente solo se `lastPriority > currPriority`. — core-libs/core/src/util/applicable.ts:L76-L86
2. **Cosa restituisce `ConverterService.convert(source, token)` se per `token` non è registrato alcun converter?** Lo stesso `source`, con un cast di tipo; `pipeable` in quel caso è un operatore identità. — core-libs/core/src/util/converter.service.ts:L106-L112, core-libs/core/src/util/converter.service.ts:L80-L88
3. **Perché un converter registrato in un modulo lazy viene visto dal `ConverterService` root?** Perché il servizio legge i converter da `UnifiedInjector.getMulti` (che aggrega tutti gli injector) e svuota la cache a ogni emissione di `injectors$`. — core-libs/core/src/util/converter.service.ts:L39-L44, core-libs/core/src/util/converter.service.ts:L52-L65
4. **Con le opzioni di default, quanti retry fa `backOff` e quanto attende in totale?** 3 retry, con attese 300 + 1200 + 2700 = 4200 ms, perché il primo `currentRetry` passato da `retry` di RxJS 7.8 vale 1. — core-libs/core/src/util/rxjs/back-off.ts:L39-L61, deps:rxjs-7.8.2/package/dist/cjs/internal/operators/retry.js:L34-L34
5. **Perché `HttpErrorModel` è una classe e non un'interfaccia?** Il sorgente non lo motiva; di fatto, essendo una classe, `normalizeHttpError` può riconoscerla con `instanceof` e restituirla invariata. — core-libs/core/src/model/misc.model.ts:L100-L106, core-libs/core/src/util/try-normalize-http-error.ts:L33-L35
6. **Da dove prende i `details` `tryNormalizeHttpError` per una risposta d'errore OAuth?** Se `error.error.error` è una stringa, crea un unico dettaglio `{ type: error, message: error_description }`. — core-libs/core/src/util/try-normalize-http-error.ts:L44-L54
7. **Cosa succede leggendo `windowRef.location.pathname` durante il rendering SSR?** Si ottiene `undefined` (in SSR `location` ha solo `href` e `origin`), oppure un'eccezione se `SERVER_REQUEST_URL`/`SERVER_REQUEST_ORIGIN` non sono forniti. — core-libs/core/src/window/window-ref.ts:L70-L89
8. **Perché in SSR `ScriptLoader.embedScript` con una `callback` non fa nulla?** Perché in quel caso esce subito: `if ((callback || errorCallback) && isSSR) return;`. — core-libs/core/src/util/script-loader.service.ts:L55-L58
9. **Si può importare `bufferDebounceTime` o `CombinedInjector` da `@spartacus/core`?** No: `bufferDebounceTime` non è nel barrel `util/rxjs/index.ts`, `CombinedInjector` non è riesportato da alcun barrel. — core-libs/core/src/util/rxjs/index.ts:L7-L11, core-libs/core/src/util/index.ts:L7-L29
10. **Come aggiunge subscription-billing il valore `SUBSCRIPTION` all'enum `ProductScope`, e perché non basta `declare module`?** Con `declare module '@spartacus/core' { enum ProductScope { SUBSCRIPTION = 'subscription' } }` più l'assegnazione runtime `(ProductScope as any)['SUBSCRIPTION'] =`

`'subscription'`, perché l'augmentation agisce solo sul tipo. — `feature-libs/subscription-billing/root/model/subscription-product.model.ts:L98-L110`

TAVOLA 46.1 **4.2 ConverterService: cache, lookup e catena**



47

L'app demo storefrontapp

Il banco di prova che accende qualunque combinazione di librerie, guidato dai test e2e.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

---

1. A cosa serve e perché esiste

`projects/storefrontapp` è l'applicazione Angular con cui il team Spartacus sviluppa, prova e collauda le librerie del monorepo. Non è un'app "di esempio" nel senso di un template da copiare: è un banco di prova che deve poter accendere **qualunque** combinazione di librerie (B2C, B2B, integrazioni CDC/CDS/OPF/S4/...) a partire dalla stessa base di codice, deve essere guidabile dai test e2e (Cypress e SSR) e deve esercitare quotidianamente i percorsi di codice più recenti. Il codice stesso lo dichiara in più punti:

- il root component "belongs to the example app, not Spartacus libraries" (`projects/storefrontapp/src/app/app.component.ts:L10-L19`);
- `privateProviders` sono "used only in our example Storefrontapp for testing purposes. CAUTION: IT IS NOT MEANT FOR CUSTOMERS' APPLICATIONS!" (`projects/storefrontapp/src/app/private/private.providers.ts:L14-L19`);
- la scelta del canale B2B/B2C "is enabled only in our repo via a special environment variable at build time. In customers' applications, either B2B or B2C configuration is used, which is decided at the installation time" (`projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L33-L42`).

Leggere la demo serve quindi a due cose diverse:

1. **Capire come si assembla un'app Spartacus reale**: entry point standalone browser/server, `AppModule` che importa `SpartacusModule`, la triade *BaseStorefront + Features + Configuration*, i `*FeatureModule` con `featureModules` lazy, `server.ts` con Express e l'engine ottimizzato. La struttura ricalca quella che genera `ng add @spartacus/schematics` (vedi il commento su `ExternalRoutesModule`, "is added by default schematics", `projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L244-L247`).
2. **Distinguere ciò che è solo del repo**: il sistema di flag di build `CX_*` iniettati da un plugin esbuild custom, `env-cmd` con 26 ambienti, `TestConfigModule` guidato da cookie, i test outlet, i feature toggle tutti `true` imposti da ESLint. Tutto questo **non** va riprodotto in un progetto cliente.

Questo capitolo non ripete ciò che è trattato altrove:

- l'ordine di inizializzazione (`bootstrapApplication` → `ENVIRONMENT_INITIALIZER` → `APP_INITIALIZER` → navigazione iniziale) è in 10 — Bootstrap e ordine di inizializzazione: qui ci fermiamo a *quali* provider la demo registra e *in che ordine* entrano nell'injector;
- l'interno di `OptimizedSsrEngine` (timeout, cache, concorrenza, fallback CSR) è in 39 — SSR ottimizzato: qui vediamo come `server.ts` lo configura;
- il meccanismo dei feature toggle è in 42 — Feature toggle e deprecazioni: qui vediamo come la demo li accende tutti;
- `ConfigChunk`/`deepMerge` sono in 11 — Configurazione, `featureModules` e `FeatureModulesService` in 12 — Feature module e lazy loading.

2. Mappa dei file

PATH	RUOLO
projects/storefrontapp/src/main.ts	bootstrap browser standalone (bootstrapApplication(AppComponent, appConfig))
projects/storefrontapp/src/main.server.ts	funzione bootstrap(context) per SSR/prerender
projects/storefrontapp/src/server.ts	app Express: engine Spartacus decorato, middleware origin, statici, catch-all di rendering, error handler, run()
projects/storefrontapp/src/index.html	documento host: meta tag OCC/media con placeholder, <base href>, <app-root>
projects/storefrontapp/src/app/app.component.ts	root app-root con <cx-storefront />
projects/storefrontapp/src/app/app.config.ts	ApplicationConfig browser (HTTP, hydration, zone, importProvidersFrom(AppModule))
projects/storefrontapp/src/app/app.config.server.ts	mergeApplicationConfig(appConfig, serverConfig)
projects/storefrontapp/src/app/app.module.ts	NgModule radice: NgRx root, AppRoutingModule, SpartacusModule, privateProviders
projects/storefrontapp/src/app/app.module.server.ts	provideServer({ serverRequestOrigin })
projects/storefrontapp/src/app/private/private.providers.ts	config solo-demo: backend, rotta product, features.level, Google Maps, TestOutletModule, TestConfigModule, devtools
projects/storefrontapp/src/app/spartacus/spartacus.module.ts	aggrega BaseStorefrontModule, SpartacusFeaturesModule, SpartacusConfigurationModule
projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts	moduli eager, *FeatureModule, lista featureModules da flag, token My Account V2, 107 feature toggle
projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts	layout/media/CMS/PWA/i18n + provider di canale
projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts	context B2C (8 siti), cart validation e selective cart
projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts	recipe defaultB2bOccConfig, checkout B2B, context powertools
projects/storefrontapp/src/app/spartacus/features/**	48 feature module + 9 wrapper module (lazy)
projects/storefrontapp/src/environments/environment.ts / environment.prod.ts	oggetto environment costruito da buildProcess.env.CX_*
projects/storefrontapp/src/environments/models/environment.model.ts	interfaccia Environment (23 campi)
projects/storefrontapp/src/environments/models/build.process.env.d.ts	declare var buildProcess e interfaccia Env dei CX_*
projects/storefrontapp/esbuild/plugins.ts	plugin esbuild: resolve-env-plugin (inietta CX_*) e filter-warnings
projects/storefrontapp/project.json	target Nx build, build-csr, prerender, serve, test
projects/storefrontapp/tsconfig.app.json / tsconfig.app.prod.json	path @spartacus/* su sorgenti (dev) o su dist/ (prod)

PATH	RUOLO
`projects/storefrontapp/prerender.txt`	11 rotte per la configurazione `withPrerender`
`projects/storefrontapp/eslint.config.mjs`	regole lint specifiche della demo
`projects/storefrontapp/src/styles.scss`, `styles-config.scss`, `cms-themes.scss`, `styles/lib-*.scss`	stili globali, `$useLatestStyles`, temi CMS, bundle per libreria
`projects/storefrontapp/src/test-outlets/**`	pagine statiche di test per `cxOutletRef` (e2e)
`projects/storefrontapp/src/test.ts`	bootstrap Karma
`.env-cmdrc` (root)	26 ambienti `env-cmd` con `CX_BASE_URL` e flag `CX_*`
`package.json` (root)	script `build`, `start*`, `build:ssr*`, `serve:ssr*`, `dev:ssr`, `prerender`

3. Come funziona passo per passo

3.1 Dalla shell al bundle: i flag di build

La demo è compilata in varianti diverse senza toccare il codice. Il meccanismo ha cinque anelli.

(a) `env-cmd` **carica ambienti nominati.** Lo script `build` è `env-cmd --no-override -e dev,b2c,$SPA_ENV -- nx run storefrontapp:build`; `start` usa gli stessi ambienti con `nx serve … --configuration=development`, e `start:b2b` sostituisce `b2c` con `b2b` (package.json:L16-L16, package.json:L133-L134). Gli ambienti stanno in `.env-cmdrc`: `dev` imposta solo `CX_BASE_URL`, `b2c` imposta `CX_B2B: "false"` e `CX_CDS: "false"`, `b2b` solo `CX_B2B: "true"` (`.env-cmdrc:L2-L4`, `.env-cmdrc:L51-L57`). Alcuni ambienti implicano B2B: `cpq`, `s4om` (B2B + S4OM + RDD + PDF invoices), `s4-service`, `punchout` (`.env-cmdrc:L42-L46`, `.env-cmdrc:L58-L64`, `.env-cmdrc:L96-L104`). La variabile di shell `SPA_ENV` permette di accodare un ambiente extra senza scrivere un nuovo script.

> DA VERIFICARE: la semantica precisa di `env-cmd -e a,b,c` (gli ambienti successivi sovrascrivono i precedenti) e di `--no-override` (non sovrascrive variabili già presenti nella shell) è dedotta dall'uso, es. `build:cds:ssr` con `-e dev,b2c,cds` dove `b2c` mette `CX_CDS: "false"` e `cds` lo riporta a `"true"` (package.json:L161-L161, `.env-cmdrc:L20-L22`). Il pacchetto `env-cmd` non è disponibile né in `/tmp/deps` né in `node_modules`.

(b) Nx esegue un builder esbuild personalizzato. Il target `build` usa `@angular-builders/custom-esbuild:application` con `plugins: ["./projects/storefrontapp/esbuild/plugins.ts"]` (projects/storefrontapp/project.json:L9-L13).

(c) Il plugin raccoglie le `CX_*`**.** `resolve-env-plugin` scorre `process.env`, tiene solo le chiavi con prefisso `CX_`, converte `'true'`/`'false'` in booleani e imposta `build.initialOptions.define['buildProcess.env']` al JSON risultante; stampa `env=` in console (projects/storefrontapp/esbuild/plugins.ts:L10-L31). Un secondo plugin, `filter-warnings`, elimina dai warning di esbuild quelli con "no side effects", "[ignored-bare-import]" e "is not ESM" (projects/storefrontapp/esbuild/plugins.ts:L34-L46).

(d) TypeScript conosce `buildProcess` **come globale.** `build.process.env.d.ts` dichiara `declare var buildProcess: BuildProcess` con l'interfaccia `Env` dei 19 `CX_*` (projects/storefrontapp/src/environments/models/build.process.env.d.ts:L7-L11, projects/storefrontapp/src/environments/models/build.process.env.d.ts:L13-L33); il file è incluso esplicitamente nei `files` di `tsconfig.app.json` (projects/storefrontapp/tsconfig.app.json:L12-L17).

(e) `environment.ts` **traduce le variabili in flag.** In sviluppo ogni flag è `buildProcess.env.CX_* ?? false`, `occBaseUrl` è `CX_BASE_URL` e `occApiPrefix` è fisso `'/occ/v2/'` (projects/storefrontapp/src/environments/environment.ts:L22-L44). La configurazione `production` del build sostituisce il file con `environment.prod.ts` tramite `fileReplacements` (projects/storefrontapp/project.json:L157-L162), che ha `production: true` e flag **senza** `?? false`, tranne `myAccountV2` (projects/storefrontapp/src/environments/environment.prod.ts:L9-L31). L'interfaccia `Environment` ha 23 campi: `production`, `occBaseUrl`, `occApiPrefix`, opzionali `mediaBaseUrl`/`mediaApiPrefix` e 18 flag booleani (projects/storefrontapp/src/environments/models/environment.model.ts:L7-L31).

I flag diventano **struttura** a import-time dei moduli ES, prima ancora che Angular parta:

- `spartacus-features.module.ts` costruisce l'array di modulo `featureModules`: con `b2b` aggiunge 7 moduli B2B, altrimenti `PickupInStoreFeatureModule`; poi un `if` per ciascuno di `cdc`, `s4Service`, `cds`, `digitalPayments`, `epdVisualization`, `pdfInvoices`, `opps`, `s4om`, `opf`, `punchout`, `segmentRefs`, `requestedDeliveryDate`, `estimatedDeliveryDate`, `omf`, `cpq` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L115-L175);
- il decoratore `@NgModule({ imports: [..., ...featureModules] })` "congela" la lista (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L285-L285);
- `spartacus-configuration.module.ts` sceglie i provider di canale con un ternario su `environment.b2b` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L40-L42);
- ogni wrapper module lazy costruisce il proprio array `extensions` (§3.5).

> DA VERIFICARE: se esbuild, grazie al `define` di `buildProcess.env` come oggetto JSON, faccia constant-folding di `buildProcess.env.CX_B2B` ed elimini i rami morti (e quindi i moduli non usati dal bundle). Non è possibile eseguire il build in questo ambiente.

3.2 Entry point browser e configurazione applicativa

`main.ts` chiama `enableProdMode()` solo se `environment.production` e poi `bootstrapApplication(AppComponent, appConfig)`, con un `.catch` che fa `console.error` ("It's just an example application file") (projects/storefrontapp/src/main.ts:L13-L23). L'avvio vero e proprio, con l'ordine degli initializer, è descritto nel capitolo 10.

`appConfig` registra (projects/storefrontapp/src/app/app.config.ts:L26-L35):

- `provideHttpClient(withFetch(), withInterceptorsFromDi())`: `withInterceptorsFromDi()` è ciò che rende attivi gli interceptor Spartacus registrati come `HTTP_INTERCEPTORS` nei NgModule (projects/storefrontapp/src/app/app.config.ts:L28-L28);
- `provideClientHydration(withEventReplay(), withNoHttpTransferCache())`: la cache HTTP di trasferimento di Angular è disattivata (angular:packages/platform-browser/src/hydration.ts:L72-L72), perché Spartacus trasferisce lo stato con il proprio meccanismo (vedi 07 — SSR, hydration e TransferState);
- `provideZoneChangeDetection({ eventCoalescing: true })`, `provideBrowserGlobalErrorListeners()`;
- `importProvidersFrom(AppModule)`: tutto Spartacus entra da qui.

`AppComponent` (selettore `app-root`) è standalone e ha come template `<cx-storefront />` (projects/storefrontapp/src/app/app.component.ts:L10-L19). `index.html` contiene `<app-root>Loading...</app-root>`, `<base href="/" />` e due meta tag con placeholder, `OCC_BACKEND_BASE_URL_VALUE` e `MEDIA_BACKEND_BASE_URL_VALUE` (projects/storefrontapp/src/index.html:L71-L82). Le factory che leggono quei meta tag ignorano il valore quando coincide con il placeholder e sono registrate da `MetaTagConfigModule.forRoot()` dentro `BaseCoreModule` (core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L18-L41, core-libs/core/src/base-core.module.ts:L39-L39): nella demo i placeholder re-

stano, quindi i meta tag non contribuiscono configurazione. Lo stesso file dichiara manifest, favicon, icona apple-touch e splash screen iOS (projects/storefrontapp/src/index.html:L4-L69).

3.3 La composizione dei moduli e l'ordine dei provider

`AppModule` importa, in quest'ordine, `BrowserModule`, `StoreModule.forRoot({})`, `AppRoutingModule`, `EffectsModule.forRoot([])`, `SpartacusModule`, e mette `privateProviders` nei propri `providers` (projects/storefrontapp/src/app/app.module.ts:L15-L25). `SpartacusModule` importa `BaseStorefrontModule`, `SpartacusFeaturesModule`, `SpartacusConfigurationModule` e ri-esporta `BaseStorefrontModule` (projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L20).

Il punto chiave è **l'ordine**: Angular, quando appiattisce un NgModule in lista di provider, include prima i moduli importati semplici (ricorsivamente), poi i provider degli import `ModuleWithProviders` (`forRoot()`), che vengono elaborati dopo tutti gli altri import, e infine i `providers` dichiarati nel modulo stesso (angular:packages/core/src/di/provider_collection.ts:L293-L354, angular:packages/core/src/di/provider_collection.ts:L316-L321). In `AppModule`, quindi, i provider di `StoreModule.forRoot({})` ed `EffectsModule.forRoot([])` entrano dopo l'intero albero di `SpartacusModule`; lo stesso spostamento vale dentro `SpartacusFeaturesModule` per `AuthModule.forRoot()`, `AnonymousConsentsModule.forRoot()`, `ProductModule.forRoot()` ed `ExternalRoutesModule.forRoot()`. Di conseguenza:

1. i chunk di config di `BaseStorefrontModule` e delle librerie entrano per primi;
2. poi quelli di `SpartacusFeaturesModule` (eager + `*FeatureModule`);
3. poi `SpartacusConfigurationModule` (layout, media, CMS, PWA, i18n, canale);
4. **per ultimi** i `privateProviders` di `AppModule`, cioè la config backend e `TestConfigModule`.

Questo è ciò che rende vero il commento "Should be imported after other config modules" accanto a `TestConfigModule.forRoot` (projects/storefrontapp/src/app/private/private.providers.ts:L54-L54). Il merge dei chunk segue quest'ordine: `Config = deepMerge({}, DefaultConfig, RootConfig)` e `RootConfig = deepMerge({}, ...ConfigChunk)` (core-libs/core/src/config/config-tokens.ts:L10-L12, core-libs/core/src/config/config-tokens.ts:L37-L39). Tutti i `provideConfig` dell'app vincono sui default delle librerie e, tra loro, vince l'ultimo. `deepMerge` fonde ricorsivamente solo oggetti; array e primitivi, compreso `undefined`, **sostituiscono** il valore precedente (core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/config/utils/deep-merge.ts:L36-L48). Il dettaglio del meccanismo è nel capitolo 11.

`provideConfig` produce un `ValueProvider` multi su `ConfigChunk` (o `DefaultConfigChunk` se `defaultConfig = true`) e `provideConfigFactory` l'equivalente `FactoryProvider` (core-libs/core/src/config/config-providers.ts:L18-L48). La demo usa sempre le varianti **non default**: per questo il suo ESLint spegne le regole `use-provide-default-config(-factory)` e `use-provide-default-feature-toggles(-factory)` che altrove le impongono (projects/storefrontapp/eslint.config.mjs:L10-L15).

3.4 Le tre "fette" di SpartacusModule

SpartacusConfigurationModule. A livello di modulo ES chiama `registerLocaleData` per `de`, `ja`, `zh` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L29-L31). I suoi provider, nell'ordine, sono `provideConfigFactory(layoutConfigFactory)`, `provideConfig(mediaConfig)`, `...defaultCmsContentProviders`, `pwa: { enabled: false, addToHomeScreen: true }`, l'i18n statico (risorse en/ja/de/zh, `translationChunksConfig`, `fallbackLang: 'en'`) e infine i provider di canale (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L44-L72).

- **B2C**: `context.urlParameters: ['baseSite','language','currency']` e `baseSite` con 8 siti electronics/apparel, con `electronics-epdvisualization-spa` anteposto se `environment.epdVisualization` (projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L11-L31); abi-

lita anche `cart.validation.enabled` e `cart.selectiveCart.enabled` (projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L32-L39).

- **B2B**: `provideConfig(defaultB2bOccConfig)`, `provideConfig(defaultB2BCheckoutConfig)` e `context.baseSite = ['powertools-spa','powertools-standalone']`, con `powertools-epdvisualization-spa` aggiunto con `unshift` se EPD (projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L13-L28). `defaultB2bOccConfig`, unica "recipe" di `@spartacus/setup`, contribuisce solo `backend.occ.endpoints` B2B (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L38-L49).

Il site context che consuma `context.*` è descritto in 13 — Site context e theme; l'i18n in 37 — i18n; la PWA in 40 — SEO e PWA.

SpartacusFeaturesModule. Importa eager auth (`AuthModule.forRoot()`, `LogoutModule`, `LoginRouteModule`), 16 moduli CMS di base, User/AnonymousConsents/Product core+UI, `CostCenterOccModule` e i moduli di page event (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L176-L242). Una sezione "Opt-in features" aggiunge `ExternalRoutesModule.forRoot()` e `JsonLdBuilderModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L244-L247). La sezione "Feature libraries" elenca 24 `*FeatureModule` sempre presenti più `...featureModules` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L249-L285). Nei provider:

- `USE_MY_ACCOUNT_V2_CONSENT` e `USE_MY_ACCOUNT_V2_NOTIFICATION_PREFERENCE` valgono `environment.myAccountV2`, "because consents feature is not code-splitted ... and not lazy-loaded" (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L287-L296);
- `provideFeatureTogglesFactory` restituisce un oggetto tipizzato `Required<FeatureToggles>` con 107 chiavi, tutte `true` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L297-L408).

BaseStorefrontModule è la base delle librerie (layout, CMS, routing, ecc.) e viene trattato nei capitoli della Parte II dedicati a quei sottosistemi.

3.5 Il pattern feature module + wrapper

Ogni libreria opzionale entra con un `*FeatureModule` sotto `features/`. Il pattern canonico (es. ASM) è (projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts:L18-L41):

1. importare **eager** il `*RootModule` della libreria;
2. fornire `provideConfig({ featureModules: { [KEY]: { module: () => import(...) } } });`
3. fornire un `provideConfig` i18n con risorse en/ja/de/zh, `chunks` e spesso `fallbackLang: 'en'`.

La lista dei componenti CMS gestiti da una feature **non** sta nella demo: è il `*RootModule` di libreria a fornirla come default. Per esempio `CartBaseRootModule` mappa 6 componenti su `CART_BASE_FEATURE`, `MiniCartComponent` su `MINI_CART_FEATURE`, `ProductAddToCartComponent` su `ADD_TO_CART_FEATURE` e dichiara l'alias `CART_BASE_CORE_FEATURE → CART_BASE_FEATURE` (feature-libs/cart/base/root/cart-base-root.module.ts:L26-L57, feature-libs/cart/base/root/cart-base-root.module.ts:L77-L83). Il `deepMerge` per chiave unisce `cmsComponents` (default di libreria) e `module` (root config dell'app). `CmsFeaturesService` mappa un componente sulla sua feature **solo** se la feature ha sia `module` sia `cmsComponents` non vuoto (core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76). `FeatureModulesService.resolveFeature` risolve gli alias stringa, rifiuta le feature senza `module` ("No module defined for Feature Module ..."), risolve le `dependencies` e carica il modulo una sola volta con `shareReplay` (core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73). `FeatureModuleConfig` ha tre campi: `module`, `dependencies`, `cmsComponents` (core-libs/core/src/cms/config/cms-config.ts:L109-L129). Il resto del meccanismo è nel capitolo 12.

Quando una libreria ha **estensioni** da altre librerie, la demo non punta `module` direttamente alla libreria ma a un **wrapper module locale**, che importa il modulo della libreria più le estensioni abilitate dai flag. Esempi:

WRAPPER	BASE	ESTENSIONI CONDIZIONALI	RIFERIMENTO
CartBaseWrapperModule	CartBaseModule	EstimatedDeliveryDateModule (estimatedDeliveryDate), CpqQuoteModule (cpq)	projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25
CheckoutWrapperModule	CheckoutModule	CheckoutB2BModule + CheckoutScheduledReplenishmentModule (b2b), DigitalPaymentsModule (digitalPayments), S4ServiceCheckoutModule (s4Service)	projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L16-L33
OrderWrapperModule	OrderModule	S4ServiceOrderModule, OmfOrderModule, OpfOrderModule	projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27
UserAccountWrapperModule, UserProfileWrapperModule, AdministrationWrapperModule, OrganizationUserRegistrationWrapperModule	modulo lib	modulo CDC corrispondente (cdc)	projects/storefrontapp/src/app/spartacus/features/user/user-account-wrapper.module.ts:L13-L22 e analoghi
CustomerTicketingWrapperModule	modulo lib	CdpCustomerTicketingModule (cdp)	projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts:L13-L22
RulebasedConfiguratorWrapperModule	modulo lib	RulebasedCpqConfiguratorModule (cpq)	projects/storefrontapp/src/app/spartacus/features/product-configurator/rulebased-configurator-wrapper.module.ts:L12-L21

I wrapper sono 9 e sono raggiunti solo con `import()` dinamico da `featureModules` (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L29-L32, projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L81-L84). Il flag `cdp` viene usato **solo** nel wrapper del customer ticketing: non esiste alcun feature module CDP dedicato (projects/storefrontapp/src/app/spartacus/features/customer-ticketing/customer-ticketing-wrapper.module.ts:L13-L22).

Varianti degne di nota:

- `CartBaseFeatureModule` registra tre feature in tre `provideConfig` separati: `CART_BASE_FEATURE` verso il wrapper, `MINI_CART_FEATURE` e `ADD_TO_CART_FEATURE` verso entry point secondari di `@spartacus/cart/base/components` (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L55).
- `CheckoutFeatureModule` importa eager anche `CheckoutB2BRootModule` e `CheckoutScheduledReplenishmentRootModule` se `b2b` e imposta `checkout.guest: true` (projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L38-L107).
- `WishListV2FeatureModule` importa `WishListFeatureModule` (V1) e ri-registra solo `CART_WISH_LIST_FEATURE` con `WishListWithV2Module`. Poiché i suoi provider vengono dopo quelli del modulo importato, l'override su `module` vince, mentre `ADD_TO_WISHLIST_FEATURE` e le traduzioni di V1 restano (projects/storefrontapp/src/app/spartacus/features/cart/wish-list-v2-feature.module.ts:L12-L27, projects/storefrontapp/src/app/spartacus/features/cart/wish-list-feature.module.ts:L32-L63).
- `CdsFeatureModule` **non** è lazy: importa `CdsModule.forRoot()` e fornisce `provideConfigFactory(cdsConfigFactory, [WindowRef])` (projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts:L84-L102). `cdsConfigFactory` restituisce `cds1` sul server; nel browser sceglie la prima config il cui `baseSite` compare in `windowRef.location.href`, con fallback `cds1` (projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts:L30-L82).
- `OpfFeatureModule` importa 9 root module OPF e registra 9 feature lazy in un solo `provideConfig` (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L92-L152); la config di checkout dipende dal canale (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L73-L91).
- `PersonalizationFeatureModule` importa eager `BaseTmsModule.forRoot()`, `GtmModule`, `AepModule` e configura gli eventi inoltrati ai tag manager (projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts:L22-L52).
- Alcuni feature module sono "solo root": `OmfFeatureModule`, `OppsFeatureModule`, `SegmentRefsFeatureModule`, `ProductMultiDimensionalListFeatureModule`; altri importano il root module e aggiungono solo le traduzioni, senza `featureModules` (`CpqQuoteFeatureModule`, `EstimatedDeliveryDateFeatureModule`, `S4OMFeatureModule`) (projects/storefrontapp/src/app/spartacus/features/omf/omf-feature.module.ts:L10-L13, projects/storefrontapp/src/app/spartacus/features/cpq-quote/cpq-quote-feature.module.ts:L18-L35). `OmfRootModule` è addirittura un NgModule vuoto: il codice OMF utile arriva solo con `OmfOrderModule` nel wrapper degli ordini (integration-libs/omf/root/omf-root.module.ts:L9-L10).
- `DigitalPaymentsFeatureModule` non importa alcun root module e fornisce solo traduzioni; `DigitalPaymentsModule` entra soltanto nel wrapper lazy del checkout (projects/storefrontapp/src/app/spartacus/features/digital-payments/digital-payments-feature.module.ts:L17-L33, projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L22-L24).

3.6 privateProviders e config e2e guidata da cookie

`privateProviders` è un `makeEnvironmentProviders([...])` (projects/storefrontapp/src/app/private/private.providers.ts:L14-L19) che:

- imposta `backend.occ.baseUrl/prefix` da `environment.occBaseUrl/occApiPrefix` e `backend.media.baseUrl/prefix` da `environment.mediaBaseUrl/mediaApiPrefix` (projects/storefrontapp/src/app/private/private.providers.ts:L20-L31). Nessuno dei due file environment valorizza i campi media, e non esiste una variabile `CX_*` per essi (projects/storefrontapp/src/environments/environment.ts:L22-L44, projects/storefrontapp/src/environments/models/build.process.env.d.ts:L13-L33);
- ridefinisce la rotta `product` con due path e `paramsMapping: { name: 'slug' }` "for e2e testing" (projects/storefrontapp/src/app/private/private.providers.ts:L33-L41);
- imposta `features: { level: '*' }`, ma `features.level` è `@deprecated` (projects/storefrontapp/src/app/private/private.providers.ts:L43-L44, core-libs/core/src/features-config/config/features-config.ts:L48-L70);
- imposta `googleMaps.apiKey` a `GOOGLE_MAPS_DEVELOPMENT_KEY_CONFIG`, cioè `'cx-development'` (projects/storefrontapp/src/app/private/private.providers.ts:L46-L49, feature-libs/storefinder/root/config/constants.ts:L7-L7);
- importa `TestOutletModule`, `TestConfigModule.forRoot({ cookie: 'cxConfigE2E' })` e, solo se `!environment.production`, `StoreDevtoolsModule.instrument()` (projects/storefrontapp/src/app/private/private.providers.ts:L52-L56).

`TestConfigModule.forRoot` contribuisce il contenuto di `TEST_CONFIG` due volte: come `FeatureTogglesChunk` (campo `features`) e come `ConfigChunk`, con eslint-disable espliciti ("deliberately providing a high priority FeatureConfig") (core-libs/core/src/config/test-config.module.ts:L83-L106). Nel browser `TEST_CONFIG` legge da `document.cookie` il cookie nominato da `TEST_CONFIG_COOKIE_NAME` e fa `JSON.parse(decodeURIComponent(...))`, restituendo `{}` in caso di errore (core-libs/core/src/config/test-config.module.ts:L42-L63). Sul server lo stesso token viene ridefinito da `TestConfigServerModule.forRoot()`, che legge l'header `Cookie` dalla `REQUEST` express (core-libs/setup/ssr/testing/test-config-server.module.ts:L26-L58). Il JSDoc ricorda il limite di 4096 byte dei cookie (core-libs/core/src/config/test-config.module.ts:L76-L82).

I `FeatureTogglesChunk` sono fusi con `Object.assign` superficiale, quindi vince l'ultimo: prima i 107 toggle dell'app, poi quelli dal cookie (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L15-L24, core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L48-L58).

3.7 Entry point server

`app.config.server.ts` esporta `config = mergeApplicationConfig(appConfig, serverConfig)`, dove `serverConfig` aggiunge `provideServerRendering()`, `importProvidersFrom(AppServerModule)` e `TestConfigServerModule.forRoot()` ("DO NOT USE IN CUSTOMERS APPS") (projects/storefrontapp/src/app/app.config.server.ts:L17-L30). `mergeApplicationConfig` concatena i `providers` nell'ordine degli argomenti: i provider server vengono dopo quelli browser e vincono negli override "last wins" (angular:packages/core/src/application/application_config.ts:L31-L38).

`AppServerModule` fornisce `...provideServer({ serverRequestOrigin: process.env['SERVER_REQUEST_ORIGIN'] })` (projects/storefrontapp/src/app/app.module.server.ts:L10-L17). `provideServer` registra `SERVER_REQUEST_ORIGIN`, `SERVER_REQUEST_URL`, un `LoggerService` server e un `MULTI_ERROR_HANDLER` (core-libs/setup/ssr/providers/ssr-providers.ts:L27-L47). La factory dell'origin dà precedenza all'opzione esplicita, poi al valore dell'express engine, altrimenti lancia un errore che suggerisce `SERVER_REQUEST_ORIGIN=... yarn prerender` (core-libs/setup/ssr/providers/server-request-origin.ts:L14-L40). Il JSDoc spiega che in SSR l'origin è risolta automaticamente da express, mentre per il prerendering è *mandatory* (core-libs/setup/ssr/providers/model.ts:L10-L22).

`main.server.ts` esporta di default `bootstrap(context: BootstrapContext)`, che inoltra il `context` a `bootstrapApplication(AppComponent, config, context)` (projects/storefrontapp/src/main.server.ts:L19-L26).

3.8 server.ts: Express e engine decorato

A livello di modulo, `server.ts` costruisce `ssrOptions` con `timeout = Number(process.env['SSR_TIMEOUT'] ?? defaultSsrOptimizationOptions.timeout)` e `cache = process.env['SSR_CACHE'] === 'true'` (projects/storefrontapp/src/server.ts:L22-L27). Poi crea l'engine decorato, cioè una factory: `NgExpressEngineDecorator.get(engine, ssrOptions)`, dove `engine` è `ngExpressEngine` di `@spartacus/setup/ssr` (projects/storefrontapp/src/server.ts:L8-L15, projects/storefrontapp/src/server.ts:L29-L29). L'istanza di `CxCommonEngine`/`OptimizedSsrEngine` nasce a ogni chiamata di `app()`, quando la factory viene invocata con `{ bootstrap }` in `server.engine('html', ...)`; nella demo `app()` è chiamata una sola volta da `run()` (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L57-L73, projects/storefrontapp/src/server.ts:L51-L56).

- I default sono `timeout: 3_000`, `cache: false`, `concurrency: 10`, `forcedSsrTimeout: 60_000`, `maxRenderTime: 300_000`, `reuseCurrentRendering: true` e altri (core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L220-L237). Il costruttore di `OptimizedSsrEngine` li fonde con le opzioni parziali (core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L59-L74), quindi passare solo `timeout` e `cache` è sufficiente.
- `decorateExpressEngine` antepone `getServerRequestProviders()` ai provider dell'utente e, se ci sono opzioni di ottimizzazione, avvolge l'istanza in `OptimizedSsrEngine` (core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L74).
- `ngExpressEngine` crea un `CxCommonEngine` (sottoclasse di `CommonEngine`) con `bootstrap`, `providers`, `enablePerformanceProfiler`, `allowedHosts` e calcola l'URL da protocollo, host, `baseUrl` e `url` (core-libs/setup/ssr/engine/ng-express-engine.ts:L82-L106, core-libs/setup/ssr/engine/cx-common-engine.ts:L20-L22).

La funzione `app()`, "exported so that it can be used by serverless Functions" (projects/storefrontapp/src/server.ts:L31-L32):

1. risolve `browserDistFolder = ../browser` da `import.meta.url` e legge **sincronamente** `index.server.html` (projects/storefrontapp/src/server.ts:L32-L37);
2. imposta `trust proxy = 'loopback'` (projects/storefrontapp/src/server.ts:L39-L39);
3. installa `getOriginValidationMiddleware({ allowedOrigins: process.env['SSR_ALLOWED_ORIGINS'] })` (projects/storefrontapp/src/server.ts:L45-L49). Il middleware accetta array o CSV; se la lista è vuota è un no-op, altrimenti risponde `400 Bad Request` con `Cache-Control: no-store` alle origin non ammesse (core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L51-L67). Il confronto è case-insensitive e `*` corrisponde a esattamente un'etichetta host (core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L84-L99). L'origin usa `X-Forwarded-Host` solo se `trust proxy fn` considera fidato il remoteAddress (core-libs/setup/ssr/express-utils/express-request-origin.ts:L9-L26);
4. registra l'engine `html` con `ngExpressEngine({ bootstrap })` e imposta `views` su `browserDistFolder` (projects/storefrontapp/src/server.ts:L51-L59);
5. serve come statico ogni path che contiene un punto (`/.*\..*/`), con `maxAge: '1y'` (projects/storefrontapp/src/server.ts:L61-L67);
6. per tutte le altre rotte chiama `res.render(indexHtml, { req, providers: [{ provide: APP_BASE_HREF, useValue: req.baseUrl }] })` (projects/storefrontapp/src/server.ts:L69-L75);
7. chiude con `defaultExpressErrorHandlers(indexHtmlContent)` (projects/storefrontapp/src/server.ts:L77-L77), che, se gli header non sono già partiti, risponde con il documento CSR, `Cache-Control: no-store` e status 404 per `CmsPageNotFoundOutboundHttpError` oppure 500 per il resto (core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32).

`run()` legge `PORT` (default 4000), chiama `listen` ed è invocata incondizionatamente in fondo al file (projects/storefrontapp/src/server.ts:L82-L97).

`server.ts` **non** passa `allowedHosts` all'engine (projects/storefrontapp/src/server.ts:L51-L56): gli script npm impostano `NG_ALLOWED_HOSTS=localhost` (package.json:L152-L155), che `CommonEngine` legge dall'ambiente con precedenza sull'opzione (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L111). Se l'host non è ammesso e non c'è alcun allowed host configurato, `CommonEngine.render` logga un errore e restituisce il documento non renderizzato ("Falling back to client side rendering") (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L113-L130).

Cosa succede *dentro* `OptimizedSsrEngine` (code di rendering, timeout, fallback CSR, cache) è nel capitolo 39.

3.9 Target di build, script SSR e tsconfig

Opzioni di `build`: `outputPath: dist/storefrontapp`, `browser: src/main.ts`, `server: src/main.server.ts`, `ssr.entry: src/server.ts`, `prerender: false`, `polyfills: ["zone.js"]`, `aot: true` (projects/storefrontapp/project.json:L14-L24, projects/storefrontapp/project.json:L142-L147). La configurazione `production` attiva `optimization`, `outputHashing: all`, niente source map, `serviceWorker: false` e `tsConfig: tsconfig.app.prod.json` (projects/storefrontapp/project.json:L150-L171). Esistono anche `development`, `noSsr` e `withPrerender` (con `routesFile: prerender.txt`); il default è `production` (projects/storefrontapp/project.json:L172-L188). `prerender.txt` elenca 11 URL esplicite con baseSite, lingua e valuta nel path (projects/storefrontapp/prerender.txt:L1-L11).

I target `build-csr` e `prerender` sono `concat` che rimandano a `build` con `noSsr` o `withPrerender`; `serve` usa **sempre** `noSsr` (projects/storefrontapp/project.json:L190-L229).

In sviluppo `tsconfig.app.json` estende il `tsconfig.json` di root, dove `@spartacus/*` punta ai sorgenti; in produzione `tsconfig.app.prod.json` rimappa 314 path `@spartacus/*` su `dist/*` (projects/storefrontapp/tsconfig.app.json:L1-L5, tsconfig.json:L795-L796, projects/storefrontapp/tsconfig.app.prod.json:L508-L509). `tsconfig.app.json` disattiva `strict` (con un TODO) e `strictTemplates`, e abilita `skipLibCheck` per la dipendenza `@sapui5/ts-types-esm` di EPD (projects/storefrontapp/tsconfig.app.json:L4-L20).

Script SSR (package.json:L150-L155):

- `serve:ssr` → `NG_ALLOWED_HOSTS=localhost node dist/storefrontapp/server/server.mjs`;
- `serve:ssr:watch`/`serve:ssr:dev` → `SSR_TIMEOUT=30000` e `NODE_TLS_REJECT_UNAUTHORIZED=0`; `serve:ssr:ci` → `SSR_TIMEOUT=0`;
- `dev:ssr` → cancella `dist/storefrontapp/server` e lancia in parallelo `watch` e `serve:ssr:watch`.

3.10 Stili e test outlet

`styles.scss` importa, nell'ordine, `styles-config`, il core Spartacus, i parziali Bootstrap, `@spartacus/styles` e infine `cms-themes`, che deve venire dopo per sovrascrivere le custom property di `:root` (projects/storefrontapp/src/styles.scss:L1-L29). `styles-config.scss` imposta `$useLatestStyles: true` (projects/storefrontapp/src/styles-config.scss:L1-L2), sovrascrivendo il default `false !default` (core-libs/styles/scss/_versioning.scss:L46-L46). `cms-themes.scss` definisce le classi `.santorini`, `.lambda`, `.sparta` che ridefiniscono 24 custom property `--cx-color-*` (projects/storefrontapp/src/cms-themes.scss:L1-L80). Gli stili delle librerie sono 23 entry con `bundleName` distinto (`styles` + 22 `lib-*`) (projects/storefrontapp/project.json:L43-L141). Il resto è nel capitolo 38.

`TestOutletModule` importa tre moduli di pagina di test e `ConfigModule.withConfigFactory(testOutletPagesCmsContentConfig)` (projects/storefrontapp/src/test-outlets/test-

outlet.module.ts:L14-L22). La config definisce in `cmsStructure` tre pagine statiche con `ignoreBackend: true` (projects/storefrontapp/src/test-outlets/test-outlet-cms-page.config.ts:L33-L76); ogni modulo registra una rotta `test/outlet/*` con `canActivate: [CmsPageGuard]` (projects/storefrontapp/src/test-outlets/test-outlet-component/test-outlet-component.module.ts:L17-L32). I template dichiarano `ng-template [cxOutletRef]` in posizione `before`, `after` e replace (projects/storefrontapp/src/test-outlets/test-outlet-component/test-outlet-component.component.html:L1-L13). Sono un buon esempio minimo d'uso degli outlet (vedi 17 — Outlet).

---

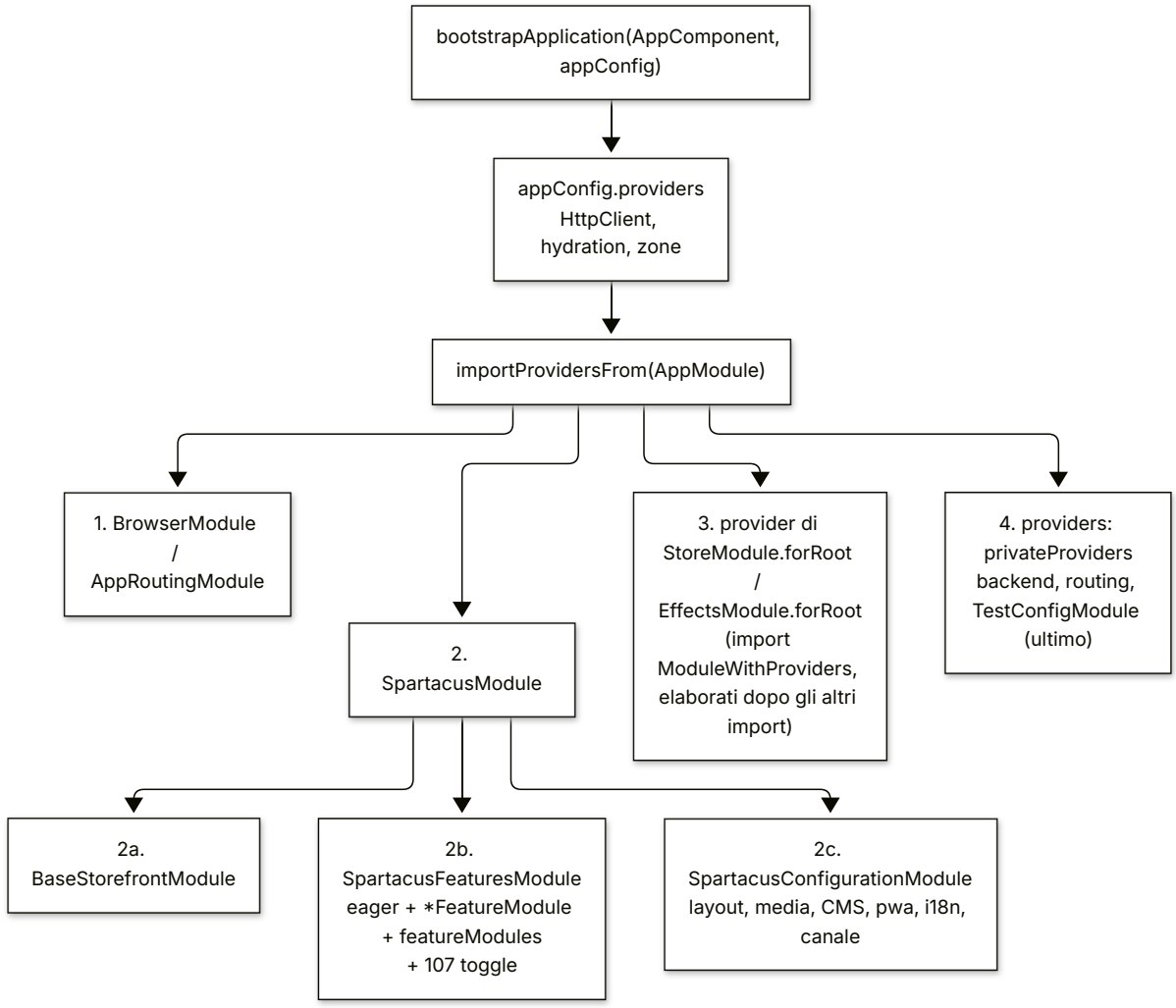
4. Diagrammi

4.1 Dai flag di build alla struttura dell'app

■ **TAVOLA 47.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: package.json:L16-L16, .env-cmdrc:L2-L4, projects/storefrontapp/project.json:L9-L13, projects/storefrontapp/esbuild/plugins.ts:L10-L31, projects/storefrontapp/project.json:L157-L162, projects/storefrontapp/src/environments/environment.ts:L22-L44, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L115-L175, projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L40-L42, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25, projects/storefrontapp/src/app/private/private.providers.ts:L20-L31.

4.2 Albero dei moduli e ordine dei provider



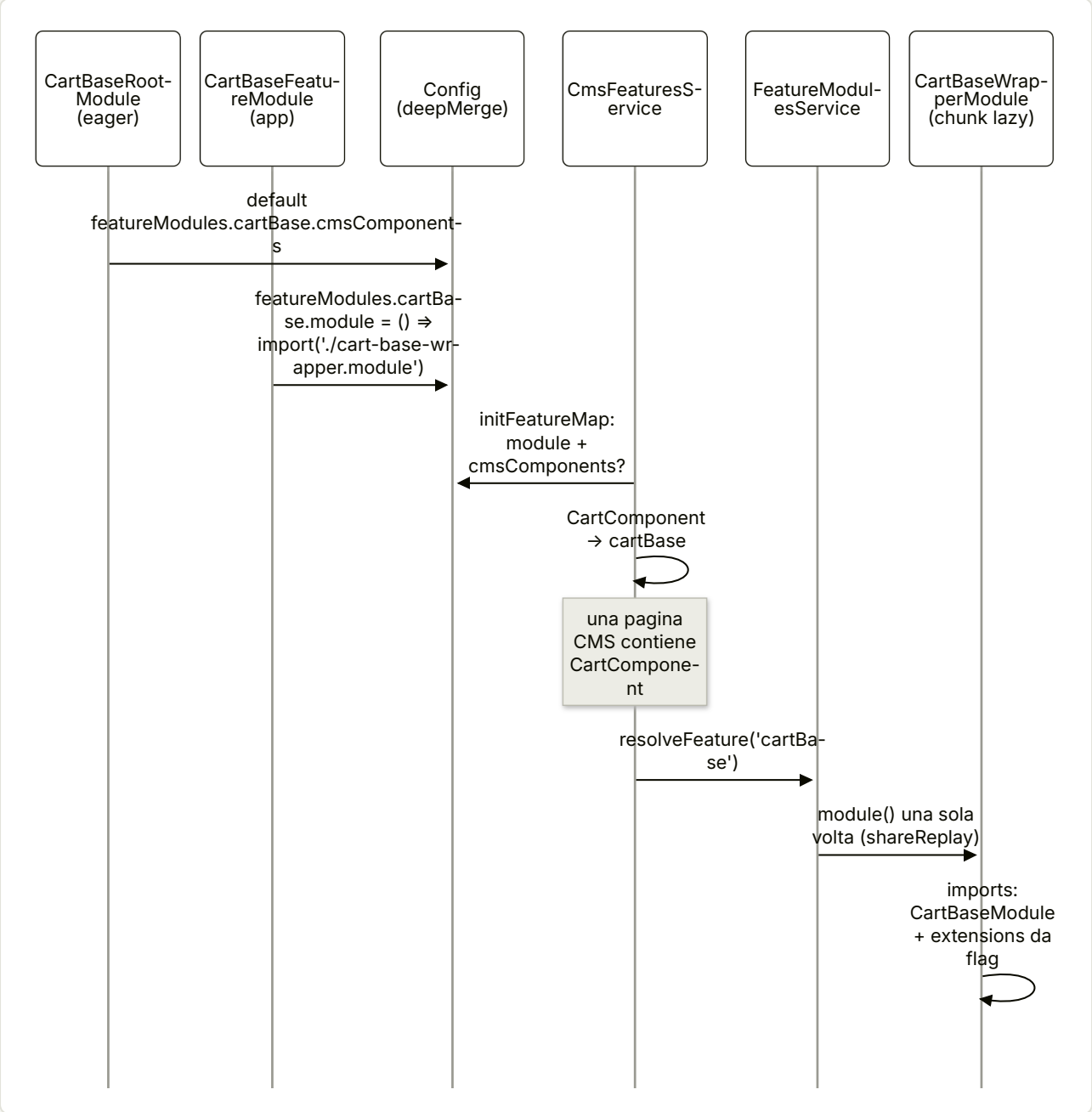
L'ordine numerico è l'ordine in cui i provider entrano nella lista (prima gli `imports` semplici, poi i provider degli import `forRoot()`, poi i `providers`): projects/storefrontapp/src/app/app.module.ts:L15-L25, projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L20, angular:packages/core/src/di/provider_collection.ts:L293-L354, angular:packages/core/src/di/provider_collection.ts:L316-L321.

4.3 Richiesta SSR

■ **TAVOLA 47.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: projects/storefrontapp/src/server.ts:L45-L77, core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L51-L67, core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L74, deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L130, projects/storefrontapp/src/main.server.ts:L23-L24, core-libs/setup/ssr/error-handling/express-error-handlers/express-error-handlers.ts:L21-L32.

4.4 Lazy loading di una feature con wrapper (cart)



Riferimenti: `feature-libs/cart/base/root/cart-base-root.module.ts:L26-L57`, `projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L35`, `core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76`, `core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73`, `projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25`.

5. Estratti di codice reale

5.1 Il plugin che porta i flag nel bundle

projects/storefrontapp/esbuild/plugins.ts:L10-L31

```ts
const resolveEnvPlugin: Plugin = {
  name: 'resolve-env-plugin',
  setup(build) {
    const env: Record<string, string | boolean | undefined> = {};
    Object.keys(process.env).forEach((key) => {
      if (key.startsWith('CX_')) {
        if (process.env[key] === 'true') {
          env[key] = true;
        } else if (process.env[key] === 'false') {
          env[key] = false;
        } else {
          env[key] = process.env[key];
        }
      }
    });
    build.initialOptions.define = {
      ...build.initialOptions.define,
      'buildProcess.env': JSON.stringify(env),
    };
    console.log('env=', env); // eslint-disable-line no-console
  },
};
```

5.2 L'oggetto environment di sviluppo

projects/storefrontapp/src/environments/environment.ts:L22-L30

```ts
export const environment: Environment = {
  production: false,
  occBaseUrl: buildProcess.env.CX_BASE_URL,
  occApiPrefix: '/occ/v2/',
  cds: buildProcess.env.CX_CDS ?? false,
  b2b: buildProcess.env.CX_B2B ?? false,
  cdc: buildProcess.env.CX_CDC ?? false,
  cdp: buildProcess.env.CX_CDP ?? false,
  cpq: buildProcess.env.CX_CPQ ?? false,
  // ...
};
```

5.3 La configurazione applicativa browser e server

projects/storefrontapp/src/app/app.config.ts:L26-L35

```ts
export const appConfig: ApplicationConfig = {
  providers: [
    provideHttpClient(withFetch(), withInterceptorsFromDi()),
    provideClientHydration(withEventReplay(), withNoHttpTransferCache()),
    provideZoneChangeDetection({ eventCoalescing: true }),
    provideBrowserGlobalErrorListeners(),

    importProvidersFrom(AppModule),
  ],
};
```

projects/storefrontapp/src/app/app.config.server.ts:L17-L30

```ts
const serverConfig: ApplicationConfig = {
  providers: [
    provideServerRendering(),

    importProvidersFrom(AppServerModule),

    importProvidersFrom(
      // DO NOT USE IN CUSTOMERS APPS:
      TestConfigServerModule.forRoot() // Injects config dynamically from e2e tests for SSR.
    ),
  ],
};

export const config = mergeApplicationConfig(appConfig, serverConfig);
```

projects/storefrontapp/src/main.server.ts:L23-L26

```ts
const bootstrap = (context: BootstrapContext) =>
  bootstrapApplication(AppComponent, config, context);

export default bootstrap;
```

5.4 AppModule: imports prima, privateProviders dopo

projects/storefrontapp/src/app/app.module.ts:L15-L25

```ts
@NgModule({
  imports: [
    BrowserModule,
    StoreModule.forRoot({}),
    AppRoutingModule,
```

```
    EffectsModule.forRoot([]),
    SpartacusModule,
  ],
  providers: [privateProviders],
})
export class AppModule {}
```

projects/storefrontapp/src/app/private/private.providers.ts:L52-L56

```ts
  importProvidersFrom(
    TestOutletModule, // Custom usages of cxOutletRef only for e2e testing
    TestConfigModule.forRoot({ cookie: 'cxConfigE2E' }), // Injects config dynamically from e2e tests. Should be imported after other config modules.
    ...(environment.production ? [] : [StoreDevtoolsModule.instrument()]) // Enable Redux devtools only in non-production build
  ),
```

5.5 Feature module scelti a import-time

projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L115-L133

```ts
const featureModules = [];

if (environment.b2b) {
  featureModules.push(
    AdministrationFeatureModule,
    AccountSummaryFeatureModule,
    BulkPricingFeatureModule,
    OrderApprovalFeatureModule,
    OrganizationUserRegistrationFeatureModule,
    UnitOrderFeatureModule,
    FutureStockFeatureModule
  );
} else {
  featureModules.push(PickupInStoreFeatureModule);
}

if (environment.cdc) {
  featureModules.push(CdcFeatureModule);
}
```

projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L297-L300

```ts
    provideFeatureTogglesFactory(() => {
      const appFeatureToggles: Required<FeatureToggles> = {
        useAdvancedGoogleMarkers: true,
        useGoogleMapsAsyncLoading: true,
```

5.6 Scelta del canale

projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L40-L42

```ts
const spartacusChannelSpecificConfigurationProviders = environment.b2b
  ? spartacusB2bConfigurationProviders
  : spartacusB2cConfigurationProviders;
```

5.7 Feature module e wrapper

projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L35

```ts
@NgModule({
  imports: [CartBaseRootModule],
  providers: [
    provideConfig({
      featureModules: {
        [CART_BASE_FEATURE]: {
          module: () =>
            import('./cart-base-wrapper.module').then(
              (m) => m.CartBaseWrapperModule
            ),
        },
      },
    }),
```

projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25

```ts
const extensions: Type<any>[] = [];

if (environment.estimatedDeliveryDate) {
  extensions.push(EstimatedDeliveryDateModule);
}
if (environment.cpq) {
  extensions.push(CpqQuoteModule);
}
@NgModule({
  imports: [CartBaseModule, ...extensions],
})
export class CartBaseWrapperModule {}
```

5.8 server.ts

projects/storefrontapp/src/server.ts:L22-L37

```ts
const ssrOptions: SsrOptimizationOptions = {
  timeout: Number(
    process.env['SSR_TIMEOUT'] ?? defaultSsrOptimizationOptions.timeout
```

```
  ),
  cache: process.env['SSR_CACHE'] === 'true',
};

const ngExpressEngine = NgExpressEngineDecorator.get(engine, ssrOptions);

// The Express app is exported so that it can be used by serverless Functions.
export function app(): express.Express {
  const server = express();
  const serverDistFolder = dirname(fileURLToPath(import.meta.url));
  const browserDistFolder = resolve(serverDistFolder, '../browser');
  const indexHtml = join(serverDistFolder, 'index.server.html');
  const indexHtmlContent = readFileSync(indexHtml, 'utf-8');
```

projects/storefrontapp/src/server.ts:L61-L77

```ts
  // Serve static files from /browser
  server.get(
    /.*\..*/,
    express.static(browserDistFolder, {
      maxAge: '1y',
    })
  );

  // All regular routes use the Universal engine
  server.get(/.*/, (req, res) => {
    res.render(indexHtml, {
      req,
      providers: [{ provide: APP_BASE_HREF, useValue: req.baseUrl }],
    });
  });

  server.use(defaultExpressErrorHandlers(indexHtmlContent));
```

5.9 Engine decorato (libreria)

core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L59-L72

```ts
    const engineInstance = ngExpressEngine({
      ...setupOptions,
      providers: [
        // add spartacus related providers
        ...getServerRequestProviders(),
        ...(setupOptions.providers ?? []),
      ],
    });

    // apply optimization wrapper if optimization options were defined
```

```
    return optimizationOptions
      ? new OptimizedSsrEngine(engineInstance, optimizationOptions)
          .engineInstance
      : engineInstance;
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`CX_BASE_URL` (build)	string	per ambiente (dev: `https://40.76.109.9:9002`)	`.env-cmdrc:L2-L4`	`projects/storefrontapp/src/environments/environment.ts:L24-L24` → `projects/storefrontapp/src/app/private/private.providers.ts:L24-L24`
`CX_B2B`, `CX_CDS`, `CX_CDC`, `CX_CDP`, `CX_CPQ`, `CX_OPF`, `CX_S4OM`, … (18 flag)	boolean (da `'true'`/`'false'`)	dev `false`; prod `undefined` (tranne `myAccountV2` = `false`)	`.env-cmdrc`, `projects/storefrontapp/esbuild/plugins.ts:L14-L24`	`projects/storefrontapp/src/environments/environment.ts:L26-L43`, `projects/storefrontapp/src/environments/environment.prod.ts:L13-L30`
`SPA_ENV` (shell)	string	vuoto	—	`package.json:L16-L16`
`SSR_TIMEOUT` (runtime)	number (ms)	`3000`	`core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L226-L226`	`projects/storefrontapp/src/server.ts:L23-L25`
`SSR_CACHE` (runtime)	`'true'` → boolean	`false`	`core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L221-L221`	`projects/storefrontapp/src/server.ts:L26-L26`
`SSR_ALLOWED_ORIGINS` (runtime)	stringa CSV	assente → no-op	—	`projects/storefrontapp/src/server.ts:L45-L49`
`PORT` (runtime)	number/string	`4000`	—	`projects/storefrontapp/src/server.ts:L83-L83`
`SERVER_REQUEST_ORIGIN` (runtime)	string	assente → origin da express	—	`projects/storefrontapp/src/app/app.module.server.ts:L13-L13`
`NG_ALLOWED_HOSTS` (runtime)	stringa	`localhost` negli script	`package.json:L152-L155`	`deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L10-L11`

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cookie `cxConfigE2E`	JSON URI-encoded	assente → `{}`	`projects/storefrontapp/src/app/private/private.providers.ts:L54-L54`	`core-libs/core/src/config/test-config.module.ts:L42-L63`, `core-libs/setup/ssr/testing/test-config-server.module.ts:L42-L54`
`backend.occ.baseUrl` / `prefix`	string	`environment.occBaseUrl` / `'/occ/v2/'`	`projects/storefrontapp/src/app/private/private.providers.ts:L22-L26`	config globale (cap. 21)
`backend.media.baseUrl` / `prefix`	string	`undefined`	`projects/storefrontapp/src/app/private/private.providers.ts:L27-L30`	`core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L95-L99`
`routing.routes.product`	paths + paramsMapping	2 path, `name→slug`	`projects/storefrontapp/src/app/private/private.providers.ts:L34-L41`	routing (cap. 14)
`features.level`	string	`'*'` (deprecato)	`projects/storefrontapp/src/app/private/private.providers.ts:L44-L44`	`core-libs/core/src/features-config/utils/feature-config-utils.ts:L15-L21`
`googleMaps.apiKey`	string	`'cx-development'`	`projects/storefrontapp/src/app/private/private.providers.ts:L49-L49`	storefinder
`context.baseSite` / `urlParameters`	string[]	B2C 8 siti / B2B 2 siti; `['baseSite','language','currency']`	`projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L26-L31`, `projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L22-L27`	site context (cap. 13)
`cart.validation.enabled`, `cart.selectiveCart.enabled`	boolean	`true` (solo B2C)	`projects/storefrontapp/src/app/spartacus/spartacus-b2c-configuration.providers.ts:L32-L39`	cart (cap. 25)
`pwa.enabled` / `addToHomeScreen`	boolean	`false` / `true`	`projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L49-L54`	PWA (cap. 40)
`i18n.resources` / `chunks` / `fallbackLang`	oggetti	en/ja/de/zh, `translationChunksConfig`, `'en'`	`projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L55-L67` + ogni feature module	i18n (cap. 37)
`checkout.guest`	boolean	`true`	`projects/storefrontapp/src/app/spartacus/features/checkout/checkout-feature.module.ts:L99-L103`	checkout

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`userAccount.enableUserCurrencySync`	boolean	`false`	`projects/storefrontapp/src/app/spartacus/features/user/user-feature.module.ts:L42-L46`	user account
`featureModules[*].module`	`() => Promise`	per feature	ogni `*-feature.module.ts`	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73`, `core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76`
`USE_MY_ACCOUNT_V2_*` (token)	boolean	`environment.myAccountV2`	`projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L289-L296`, `projects/storefrontapp/src/app/spartacus/features/order/order-feature.module.ts:L34-L37`, `projects/storefrontapp/src/app/spartacus/features/user/user-feature.module.ts:L79-L90`	componenti my-account
FeatureToggles (107)	boolean	tutti `true`	`projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L297-L408`	`FeatureToggles` (cap. 42)
`$useLatestStyles` (SCSS)	boolean	`true` (default lib `false`)	`projects/storefrontapp/src/styles-config.scss:L2-L2`	`core-libs/styles/scss/_versioning.scss:L46-L46`

7. Commenti e note notevoli dal sorgente

- "It's just an example application file. This message is not crucial to be logged using any special logger." — `projects/storefrontapp/src/main.ts:L18-L22`, ripetuto in `projects/storefrontapp/src/server.ts:L88-L92`.
- "Load `$localize` onto the global scope - used if i18n tags appear in Angular templates." — `projects/storefrontapp/src/main.server.ts:L7-L9`. È un **residuo**: il file non importa `@angular/localize` (`projects/storefrontapp/src/main.server.ts:L7-L17`).
- "// All regular routes use the Universal engine" — `projects/storefrontapp/src/server.ts:L69-L69`: riferimento storico ad Angular Universal; oggi l'engine è `CxCommonEngine` su `@angular/ssr`.
- "Validates the resolved request origin against an allowlist … when unset or empty the middleware is a no-op." — `projects/storefrontapp/src/server.ts:L41-L44`.
- "DO NOT USE IN CUSTOMERS APPS" — `projects/storefrontapp/src/app/app.config.server.ts:L24-L25`; "CAUTION: IT IS NOT MEANT FOR CUSTOMERS' APPLICATIONS!" — `projects/storefrontapp/src/app/private/private.providers.ts:L17-L17`; "CAUTION: DON'T USE IT IN PRODUCTION! IT HASN'T BEEN REVIEWED FOR SECURITY ISSUES." — `core-libs/core/src/config/test-config.module.ts:L81-L81`, `core-libs/setup/ssr/testing/test-config-server.module.ts:L34-L34`.

- "Should be imported after other config modules." — projects/storefrontapp/src/app/private/private.providers.ts:L54-L54: è vero per costruzione, grazie all'ordine imports→providers (§3.3).
- "LogoutModule, // will be come part of auth package" — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L180-L181.
- "ExternalRoutesModule.forRoot(), // to opt-in explicitly, is added by default schematics" — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L246-L246.
- "Adding the provider here because consents feature is not code-splitted to separate library and not lazy-loaded" — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L288-L288.
- "we bring in static translations to be up and running soon right away" — projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L56-L56.
- JSDoc di `WishListFeatureModule`: API legacy V1 basata su SavedCart; per nuove installazioni V2 (endpoint `/wishlists`, da SAP Commerce Cloud 2211-jdk21.9) — projects/storefrontapp/src/app/spartacus/features/cart/wish-list-feature.module.ts:L22-L31.
- "Only differences to the default cds config, they are merged together." — projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts:L23-L28.
- Il commento in testa a `environment.ts` cita `ng build --env=prod` e `angular.json`: è **obsoleto**, oggi la sostituzione avviene con `fileReplacements` in `project.json` — projects/storefrontapp/src/environments/environment.ts:L7-L10, projects/storefrontapp/project.json:L157-L162.
- "This import should be commented out in production mode..." su `zone.js/plugins/zone-error` — projects/storefrontapp/src/environments/environment.ts:L12-L19.
- "TODO: Fix strict errors and remove this option" — projects/storefrontapp/tsconfig.app.json:L5-L5.
- "Angular 21 introduced a change that causes NG0100 errors in Karma tests ... we must manually provide zone change detection." — projects/storefrontapp/src/test.ts:L30-L46.
- "CMS-driven base-site theme rules (must come after Spartacus styles so the CSS custom properties override the defaults in `:root`)." — projects/storefrontapp/src/styles.scss:L27-L29.
- La regola ESLint `no-storefrontapp-false-feature-toggles` spiega che tutti i toggle devono essere `true` "so that developers exercise new code paths daily" — tools/eslint-rules/rules/no-storefrontapp-false-feature-toggles.ts:L33-L45.

---

8. Extension point

La demo è soprattutto un **catalogo di esempi** degli extension point pubblici. Ecco quelli che vale la pena copiare in un'app cliente, e quelli da non copiare.

Da imitare (API pubbliche):

SIMBOLO	USO NELLA DEMO	DEFINIZIONE	RICETTA
`provideConfig`, `provideConfigFactory`	tutti i chunk di config dell'app	core-libs/core/src/config/config-providers.ts:L18-L48	50 — Ricette: configurazione, DI, interceptor, eventi
`provideFeatureTogglesFactory`, `FeatureToggles`	107 toggle	core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L44-L52	50, 54 — Upgrade-safe
`CmsConfig.featureModules` / `FeatureModuleConfig` + `*RootModule`	ogni `*FeatureModule`	core-libs/core/src/cms/config/cms-config.ts:L109-L129	53 — Feature lib lazy, carrello/checkout, stili, traduzioni

SIMBOLO	USO NELLA DEMO	DEFINIZIONE	RICETTA
Wrapper module lazy (modulo lib + estensioni)	9 wrapper	projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25	53
`ConfigModule.withConfigFactory` + `cmsStructure` statica	test outlet	core-libs/core/src/config/config.module.ts:L44-L53, projects/storefrontapp/src/test-outlets/test-outlet-cms-page.config.ts:L33-L76	51 — Componenti CMS, outlet, route e layout
`cxOutletRef` before/after/replace	test outlet	projects/storefrontapp/src/test-outlets/test-outlet-component/test-outlet-component.component.html:L1-L13	51
`routing.routes.<page>.paths/paramsMapping`	rotta product	projects/storefrontapp/src/app/private/private.providers.ts:L33-L41	51
`defaultB2bOccConfig` (recipe)	canale B2B	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L38-L49	52 — OCC, adapter, converter, endpoint
`NgExpressEngineDecorator`, `SsrOptimizationOptions`, `getOriginValidationMiddleware`, `defaultExpressErrorHandlers`, `provideServer`	`server.ts`, `AppServerModule`	core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L51-L74, core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L51-L67, core-libs/setup/ssr/providers/ssr-providers.ts:L27-L47	39 — SSR ottimizzato
i18n per feature (`resources`, `chunks`, `fallbackLang`)	ogni feature module	projects/storefrontapp/src/app/spartacus/features/asm/asm-feature.module.ts:L18-L41	53
Override SCSS via ordine degli import + custom property di tema	`styles.scss`, `cms-themes.scss`	projects/storefrontapp/src/styles.scss:L1-L29	53
Meta tag `occ-backend-base-url` / `media-backend-base-url`	presenti ma inerti (placeholder)	core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L18-L41	52

Da non imitare (solo repo): `TestConfigModule` e `TestConfigServerModule` (config da cookie, mai revisionata per la sicurezza), `features.level: '*'` (deprecato), il plugin esbuild `buildProcess.env` e l'`if` sui flag per scegliere moduli (un'app cliente sceglie le librerie a `ng add`), `Required<FeatureToggles>` con tutti i toggle `true` (in un'app cliente accendere un toggle è una decisione consapevole, vedi 42). La checklist completa è in 54 — API pubblica vs interna, upgrade, anti-pattern.

9. Trappole, casi limite, comportamenti controintuitivi

1. **L'ordine dei moduli è l'ordine degli override.** `TestConfigModule` vince solo perché sta nei `providers` di `AppModule`, processati dopo gli `imports`. Spostarlo in un modulo importato prima di `SpartacusModule` lo renderebbe inefficace (projects/storefrontapp/src/app/app.module.ts:L16-L23, angular:packages/core/src/di/provider_collection.ts:L293-L354). Lo stesso vale per `WishListV2FeatureModule`, che vince su V1 perché ne importa il modulo e poi aggiunge i propri provider (projects/storefrontapp/src/app/spartacus/features/cart/wish-list-v2-feature.module.ts:L12-L27).

2. **Gli array vengono sostituiti, non uniti.** `context.baseSite`, `cdc[]`, `tagManager.*.events` sono array: un `provideConfig` successivo li rimpiazza interamente (core-libs/core/src/config/utils/deep-merge.ts:L7-L9).
3. `backend.media.baseUrl: undefined` **sovrascrive.** `privateProviders` scrive esplicitamente `undefined` per i campi media, e in `deepMerge` un `undefined` sostituisce i valori precedenti, compreso un eventuale meta tag `media-backend-base-url`. Il normalizer delle immagini ripiega su `occ.baseUrl` (projects/storefrontapp/src/app/private/private.providers.ts:L27-L30, core-libs/core/src/config/utils/deep-merge.ts:L36-L48, core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L95-L99). Lo stesso vale per `occ.baseUrl` rispetto a un meta tag `occ-backend-base-url` reale (core-libs/core/src/base-core.module.ts:L39-L39).
4. `environment.prod.ts` **senza** `?? false`. Con flag non impostati i valori sono `undefined`: falsy, ma non del tipo `boolean` dichiarato (projects/storefrontapp/src/environments/environment.prod.ts:L13-L30).
5. **Solo le variabili** `CX_*` **arrivano al bundle.** `SSR_*`, `PORT`, `SERVER_REQUEST_ORIGIN`, `NG_ALLOWED_HOSTS` sono lette a runtime da Node, non a build (projects/storefrontapp/esbuild/plugins.ts:L15-L15, projects/storefrontapp/src/server.ts:L22-L27).
6. **I flag sono decisi a build time.** Cambiare `CX_B2B` richiede un nuovo build: `featureModules` ed `extensions` sono calcolati a import-time e congelati dal decoratore `@NgModule` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L115-L129, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L285-L285).
7. **Import statici nei wrapper.** I moduli di estensione (`CpqQuoteModule`, `DigitalPaymentsModule`, ...) sono importati staticamente anche quando il flag è falso (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L10-L12). DA VERIFICARE se finiscano o meno nel chunk: dipende dal tree-shaking di esbuild, e il build non è eseguibile qui.
8. `Required<FeatureToggles>` **rompe la compilazione** a ogni nuovo toggle in core finché non viene aggiunto alla demo, e la regola lint impone che sia `true` (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L298-L298, projects/storefrontapp/eslint.config.mjs:L24-L30).
9. **Feature senza** `module` **non vengono mappate ai componenti CMS.** `PRODUCT_MULTI_DIMENSIONAL_LIST_FEATURE` dichiara `cmsComponents` ma l'app non fornisce `module`; resta attivo solo l'outlet `ProductListOutlets.ITEM_DETAILS` in `REPLACE` (feature-libs/product-multi-dimensional/list/root/product-multi-dimensional-list-root.module.ts:L19-L44, core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76).
10. **CDS sceglie la config per sottostringa dell'URL.** `location.href.includes(baseSite)` e, sul server, sempre `cds1`: SSR e browser possono divergere (projects/storefrontapp/src/app/spartacus/features/cds/cds-feature.module.ts:L30-L82).
11. **Placeholder non funzionanti** nella config demo: `JS_SDK_URL_PLACEHOLDER` (CDC), `A_CDS_*` (CDS 2), `ADD_COMMERCE_CLOUD_PUBLIC_KEY_HERE` (OPF) (projects/storefrontapp/src/app/spartacus/features/cdc/cdc-feature.module.ts:L20-L33, projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L186-L193).
12. **Statici per qualunque path con un punto** (`/.*\..*/`): anche URL applicative con un punto passano prima da `express.static` (projects/storefrontapp/src/server.ts:L61-L67). DA VERIFICARE il fallthrough di `express.static` verso il catch-all di rendering quando il file non esiste: il pacchetto `express` non è disponibile in `/tmp/deps`.
13. `index.server.html` **è letto con** `readFileSync` **all'avvio.** Se manca (dist incompleto o non coerente, per esempio un `server/` vecchio), `app()` lancia prima di `listen` (projects/storefrontapp/src/server.ts:L36-L37). Con un build `noSsr` (`ssr: false`) non c'è proprio un server da avviare (projects/storefrontapp/project.json:L177-L180).
14. **Nessun** `allowedHosts` **in codice.** Senza `NG_ALLOWED_HOSTS` (e senza `allowedHosts`), `CommonEngine` logga un errore e degrada a CSR. Se invece `NG_ALLOWED_HOSTS` è impostata (come in tutti gli script `serve:ssr*`) e

l'host della richiesta non vi compare, `render` lancia l'errore `URL with hostname "..." is not allowed.`, senza fallback (projects/storefrontapp/src/server.ts:L51-L56, deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L119-L133, deps:angular-ssr-21.2.23/package/fesm2022/_validation-chunk.mjs:L15-L22).

15. `trust proxy = 'loopback'`. Dietro un reverse proxy non loopback `X-Forwarded-Host` viene ignorato e l'origin (usata dall'allowlist e da `SERVER_REQUEST_ORIGIN`) è l'header `Host` (projects/storefrontapp/src/server.ts:L39-L39, core-libs/setup/ssr/express-utils/express-request-origin.ts:L9-L26).
16. **Il prerender richiede** `SERVER_REQUEST_ORIGIN`, altrimenti `serverRequestOriginFactory` lancia (core-libs/setup/ssr/providers/server-request-origin.ts:L14-L40).
17. **Il build production compila contro** `dist/`. `tsconfig.app.prod.json` mappa `@spartacus/*` su `dist/*`, quindi `build`/`build:ssr` (default `production`) richiedono le librerie già buildate, mentre `start` usa i sorgenti (projects/storefrontapp/tsconfig.app.prod.json:L508-L509, projects/storefrontapp/project.json:L172-L188).
18. `serve` **non fa mai SSR.** Entrambe le configurazioni usano `noSsr`; per SSR in sviluppo serve `dev:ssr` (projects/storefrontapp/project.json:L190-L229, package.json:L151-L151).
19. **Stili non bundlati.** `lib-customer-360.scss` e `lib-estimated-delivery-date.scss` esistono ma non sono elencati in `project.json` (projects/storefrontapp/src/styles/lib-customer-360.scss:L1-L2, projects/storefrontapp/project.json:L43-L141). DA VERIFICARE se gli stili ASM 360 ed EDD arrivino per altra via.
20. `lib-customer-ticketing`, `lib-s4om`, `lib-subscription-billing` **non importano** `styles-config`, quindi per quelle librerie `$useLatestStyles` resta al default `false` (projects/storefrontapp/src/styles/lib-customer-ticketing.scss:L1-L1, core-libs/styles/scss/_versioning.scss:L46-L46).
21. `"types": ["node"]` **fuori da** `compilerOptions` in `tsconfig.app.json`, dove TypeScript non lo considera un'opzione di compilazione (projects/storefrontapp/tsconfig.app.json:L3-L11).
22. **Script non uniformi.** `build:ssr:ci` e `test:ssr` non usano `--no-override` (package.json:L54-L57, package.json:L156-L157); `start:opf`/`start:punchout` non includono né `dev` né `b2c` (package.json:L137-L141), e l'ambiente `punchout` non definisce `CX_BASE_URL` (.env-cmdrc:L101-L104).
23. `.env-cmdrc` **contiene IP e host di ambienti interni** (.env-cmdrc:L3-L3). Gli script `serve:ssr:watch`, `serve:ssr:ci`, `serve:ssr:dev` e `prerender` disabilitano la verifica TLS con `NODE_TLS_REJECT_UNAUTHORIZED=0` (DA VERIFICARE il motivo: nessun file del repo lo spiega; presumibilmente i certificati degli ambienti in `.env-cmdrc`, ad es. un host raggiunto per IP). `serve:ssr` non la imposta (package.json:L152-L155, package.json:L158-L158). Non va riprodotto in produzione.

---

10. Come fare debug

- **Quali flag sono finiti nel bundle?** Il plugin stampa `env=` con l'oggetto `CX_*` all'inizio del build (projects/storefrontapp/esbuild/plugins.ts:L29-L29). Se un flag non compare, controllare quali ambienti `env-cmd` sono stati passati e se la shell ha già la variabile (con `--no-override`).
- **Quale config ha vinto?** Iniettare `Config` (o `FeatureToggles`) in un componente o dalla console e confrontare con l'ordine del §3.3. Per un override veloce senza rebuild, sfruttare il cookie `cxConfigE2E` con JSON URI-encoded (core-libs/core/src/config/test-config.module.ts:L42-L63): funziona anche in SSR grazie a `TestConfigServerModule` (core-libs/setup/ssr/testing/test-config-server.module.ts:L26-L58). Attenzione al limite di 4096 byte.
- **NgRx DevTools** sono attivi solo nei build non production (projects/storefrontapp/src/app/private/private.providers.ts:L55-L55).

- **Stack trace leggibili**: in sviluppo `environment.ts` importa `zone.js/plugins/zone-error` (projects/storefrontapp/src/environments/environment.ts:L12-L19).
- **Una feature non si carica?** Verificare che la chiave in `featureModules` abbia sia `cmsComponents` (dal root module) sia `module` (dall'app); altrimenti `CmsFeaturesService` non mappa il componente (core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76) e `FeatureModulesService` lancia "No module defined for Feature Module" (core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73). Nel pannello Network del browser si vede il chunk lazy richiesto al primo componente.
- **SSR che restituisce sempre CSR?** Guardare la console di Node: senza alcun allowed host configurato (né `NG_ALLOWED_HOSTS` né `allowedHosts`) compare l'errore "Falling back to client side rendering"; se invece `NG_ALLOWED_HOSTS` è impostata ma non contiene l'host richiesto, la richiesta fallisce con `URL with hostname "..." is not allowed.` senza fallback (deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L113-L133). Altrimenti aumentare `SSR_TIMEOUT` (gli script di sviluppo usano 30000, package.json:L153-L155) e consultare i log dell'engine ottimizzato (cap. 39).
- **400 Bad Request su tutte le richieste SSR**: `SSR_ALLOWED_ORIGINS` è impostata e l'origin calcolata non corrisponde; ricordare `trust proxy = 'loopback'` (projects/storefrontapp/src/server.ts:L39-L49).
- **Ciclo di sviluppo SSR**: `npm run dev:ssr` ricompila in watch e riavvia Node quando cambia `dist/storefrontapp/server/` (package.json:L151-L153).
- **Test Karma della demo con NG0100**: vedere il workaround `ZoneChangeDetectionModule` in projects/storefrontapp/src/test.ts:L30-L46.

---

11. Domande di autoverifica

1. **Perché `TestConfigModule` riesce a sovrascrivere qualsiasi config dell'app?** Perché sta nei `providers` di `AppModule` (tramite `privateProviders`), che Angular include dopo i provider di tutti gli `imports`; il `deepMerge` dei `ConfigChunk` fa vincere l'ultimo. — projects/storefrontapp/src/app/app.module.ts:L15-L25, angular:packages/core/src/di/provider_collection.ts:L293-L354, core-libs/core/src/config/config-tokens.ts:L37-L39
2. **Come arriva `CX_B2B=true` dalla shell a `environment.b2b`?** `env-cmd` carica l'ambiente da `.env-cmdrc`; il plugin esbuild `resolve-env-plugin` raccoglie le `CX_*`, converte `'true'` in `true` e le definisce come `buildProcess.env`; `environment.ts` legge `buildProcess.env.CX_B2B ?? false`. — .env-cmdrc:L51-L57, projects/storefrontapp/esbuild/plugins.ts:L10-L31, projects/storefrontapp/src/environments/environment.ts:L27-L27
3. **Si può passare da B2C a B2B impostando una variabile d'ambiente sul server Node già buildato?** No: i flag sono iniettati a build time e usati a import-time per costruire `featureModules` e i provider di canale. Solo `SSR_*`, `PORT`, `SERVER_REQUEST_ORIGIN`, `SSR_ALLOWED_ORIGINS`, `NG_ALLOWED_HOSTS` sono runtime. — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L115-L129, projects/storefrontapp/src/server.ts:L22-L27
4. **Perché la demo punta `featureModules[CART_BASE_FEATURE].module` a un wrapper locale e non a `@spartacus/cart/base`?** Perché il wrapper importa `CartBaseModule` più le estensioni (EDD, CPQ) abilitate dai flag, nello stesso chunk lazy. — projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L35, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25
5. **Dove è definita la lista dei componenti CMS che fanno scattare il caricamento lazy del carrello?** Nel `CartBaseRootModule` della libreria, come default; l'app aggiunge solo `module`, e `CmsFeaturesService` mappa i componenti solo se esistono entrambi. — feature-libs/cart/base/root/cart-base-root.module.ts:L26-L57, core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76

6. **Cosa succede se avvio** `serve:ssr` **senza** `NG_ALLOWED_HOSTS`**?** `server.ts` non passa `allowedHosts`, quindi `CommonEngine` non ha host ammessi: logga un errore e restituisce il documento non renderizzato (fallback CSR). — projects/storefrontapp/src/server.ts:L51-L56, deps:angular-ssr-21.2.23/package/fesm2022/node.mjs:L108-L130
7. **Quali opzioni di ottimizzazione SSR configura la demo, e da dove prende le altre?** Solo `timeout` (da `SSR_TIMEOUT`, default 3000) e `cache` (da `SSR_CACHE === 'true'`). Le altre vengono da `defaultSsrOptimizationOptions`, fuse nel costruttore di `OptimizedSsrEngine`. — projects/storefrontapp/src/server.ts:L22-L27, core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L220-L237, core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L59-L74
8. **Perché aggiungere un nuovo feature toggle in** `@spartacus/core` **rompe la compilazione della demo?** Perché i toggle dell'app sono tipizzati `Required<FeatureToggles>`, quindi ogni chiave dell'interfaccia deve comparire; inoltre una regola ESLint impone il valore `true`. — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L298-L298, projects/storefrontapp/eslint.config.mjs:L24-L30
9. **Perché i meta tag** `occ-backend-base-url` **di** `index.html` **non hanno effetto nella demo?** Contengono il placeholder, che le factory ignorano; e comunque `privateProviders`, registrati dopo `BaseCoreModule`, sovrascriverebbero `backend.occ.baseUrl` (e, con `undefined`, anche `media.baseUrl`). — projects/storefrontapp/src/index.html:L71-L82, core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L18-L41, projects/storefrontapp/src/app/private/private.providers.ts:L20-L31
10. **Che differenza c'è tra** `npm start` **e** `npm run dev:ssr` **rispetto all'SSR?** `start` usa il target `serve`, che applica sempre `noSsr`; `dev:ssr` lancia il build in watch e Node su `dist/storefrontapp/server/server.mjs`. — projects/storefrontapp/project.json:L214-L229, package.json:L150-L153

TAVOLA 47.1 **4.1 Dai flag di build alla struttura dell'app**

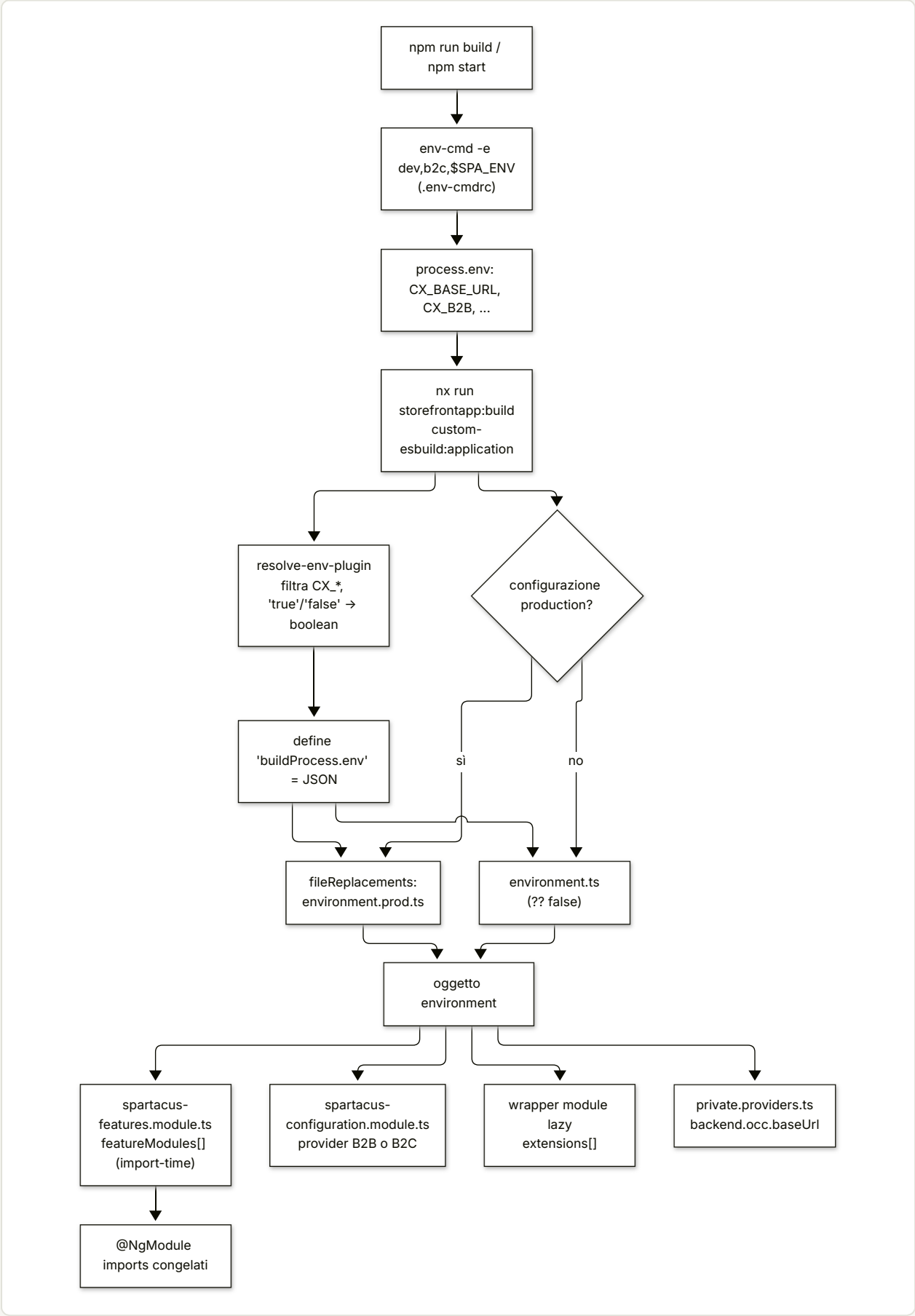
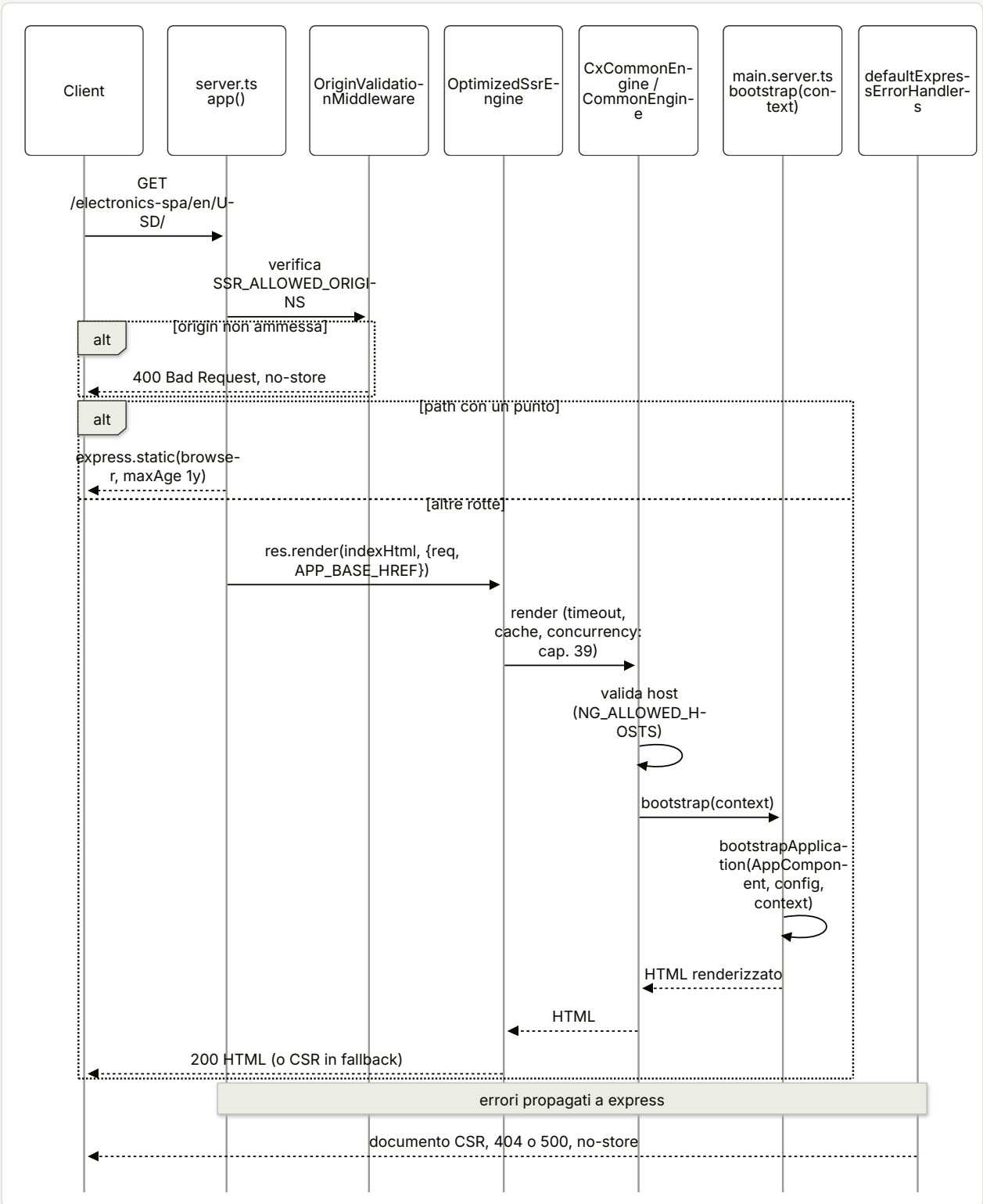


TAVOLA 47.2 **4.3 Richiesta SSR**



PARTE III

Ricette

50 Ricette: configurazione, DI, interceptor, eventi

51 Ricette: componenti CMS, outlet, route e layout

52 Ricette: OCC, adapter, converter, endpoint, meta resolver

53 Ricette: feature lib lazy, carrello/checkout, stili, traduzioni

54 API pubblica vs interna, upgrade, anti-pattern, checklist upgrade-safe

50

Ricette: configurazione, DI, interceptor, eventi

Override di config, sostituzione di servizi e facade, interceptor ed error handler, eventi custom.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Questo capitolo raccoglie le personalizzazioni "trasversali" di uno storefront Spartacus, quelle che non riguardano un componente o una feature specifica ma il modo in cui l'applicazione viene assemblata:

- **configurazione**: sovrascrivere i default delle librerie, calcolare config con una factory, definire una config tipizzata per il proprio codice, leggerla nel momento giusto;
- **dependency injection**: sostituire un servizio root caricato subito (eager) o l'implementazione di una facade caricata in lazy loading;
- **HTTP**: aggiungere un interceptor e personalizzare la gestione degli errori HTTP;
- **eventi**: definire, emettere e ascoltare eventi custom, anche derivati da altri eventi o da action NgRx.

Ogni ricetta (sezione 3) segue lo stesso schema: quando usarla, un esempio di codice che importa **solo** simboli esportati dai `public_api.ts` dei pacchetti `@spartacus/*` (oltre ad `@angular/*` e `rxjs`), i file sorgente che rendono possibile il meccanismo, i limiti. La regola "solo API pubblica" non è estetica: la superficie pubblica di `@spartacus/core` è esattamente quella che `core-libs/core/public_api.ts` riesporta dai barrel delle singole aree (config, event, lazy-loading, state, util, http, occ, auth, global-message, window, logger) — `core-libs/core/public_api.ts:L10-L36`. Un simbolo che non compare in quella catena (per esempio `configFactory`, `MergingSubject` o `CONFIG_INITIALIZER_FORROOT_GUARD`) è interno e può cambiare senza preavviso — `core-libs/core/src/config/index.ts:L7-L19`, `core-libs/core/src/event/index.ts:L7-L8`, `core-libs/core/src/config/config-initializer/index.ts:L7-L14`. Il tema è approfondito in 54 — API pubblica vs interna.

Le ricette presuppongono i meccanismi spiegati nella Parte I e II:

- DI Angular (multi provider, `providedIn`, `useExisting`/`useClass`, ordine di elaborazione dei provider): 01 — Dependency Injection;
- `HttpClient` e interceptor Angular: 06 — HttpClient e interceptor;
- config di Spartacus (chunk, `deepMerge`, initializer, validator, config unificata): 11 — Configurazione;
- feature module lazy, `featureModules`, `UnifiedInjector`: 12 — Feature module e lazy loading;
- facade, `facadeFactory`, `EventService`, `StateEventService`, query/command: 20 — Facade, command/query, eventi;
- endpoint OCC e `OccEndpointsService`: 21 — OCC;
- `AuthInterceptor` e `AuthHttpHeaderService`: 22 — Autenticazione;
- `HttpErrorHandler` e global message: 41 — Errori e global message.

2. Mappa dei file

File sorgente che abilitano le ricette (non file da modificare: le ricette si applicano nel codice dell'applicazione).

PATH	RUOLO
`core-libs/core/public_api.ts`	Superficie pubblica di `@spartacus/core`; blocco AUGMENTABLE_TYPES con `Config` (L38-L40)

PATH	RUOLO
core-libs/core/src/config/index.ts	Barrel config: provider helper, token, `ConfigModule`, `ConfigurationService`, initializer, validator
core-libs/core/src/config/config-tokens.ts	`Config`, `DefaultConfig`, `RootConfig`, `ConfigChunk`, `DefaultConfigChunk` e factory di merge
core-libs/core/src/config/config-providers.ts	`provideConfig`, `provideConfigFactory`, `provideDefaultConfig`, `provideDefaultConfigFactory`
core-libs/core/src/config/utils/deep-merge.ts	Algoritmo di merge (oggetti ricorsivi, array sostituiti)
core-libs/core/src/config/services/configuration.service.ts	Config unificata con i chunk dei moduli lazy (`unifiedConfig$`)
core-libs/core/src/config/config-initializer/config-initializer.service.ts	`getStable(...scopes)`
core-libs/core/src/config/config-validator/config-validator.ts	`provideConfigValidator`, `ConfigValidator`
core-libs/core/src/features-config/config/features-config.ts	Flag deprecato `disableConfigUpdates`
core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts	Proxy delle facade lazy: risoluzione dell'implementazione dall'injector del modulo lazy
core-libs/core/src/lazy-loading/feature-modules.service.ts	`isConfigured`, risoluzione degli alias di feature
core-libs/core/src/lazy-loading/lazy-modules.service.ts	Creazione dell'injector del modulo lazy (parent = root o dipendenze)
core-libs/core/src/lazy-loading/unified-injector.ts	`getMulti`: multi provider di root + moduli lazy
core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts	Servizio root delegato da `AuthInterceptor` (esempio di override eager)
core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts	Interceptor auth che delega a `AuthHttpHeaderService`
core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts	Sceglie l'`HttpErrorHandler` applicabile tra root + lazy
core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts	Classe base astratta degli handler di errore
core-libs/core/src/util/applicable.ts	`Priority`, `resolveApplicable`
core-libs/core/src/occ/services/occ-endpoints.service.ts	`getBaseUrl()` per riconoscere le chiamate OCC
core-libs/core/src/event/event.service.ts	`register`, `get`, `dispatch`
core-libs/core/src/event/cx-event.ts	Classe base `CxEvent`
core-libs/core/src/event/utils/merging-subject.ts	Sottoscrizione lazy alle sorgenti
core-libs/core/src/state/event/state-event.service.ts	Action NgRx → eventi
core-libs/core/src/util/create-from.ts	`createFrom(type, data)`
core-libs/storefront/events/navigation/navigation.event.ts	`NavigationEvent` (base per eventi di pagina derivati)

PATH	RUOLO
angular:packages/core/src/di/r3_injector.ts	Multi provider (push) vs single provider (sostituzione); record esplicito vs `providedIn`
angular:packages/core/src/di/provider_collection.ts	Ordine: provider degli `imports` prima dei `providers` del modulo
angular:packages/core/src/render3/di.ts	~~e~~`getInheritedFactory`: sottoclassi senza costruttore
angular:packages/common/http/src/interceptor.ts	Catena degli interceptor DI, ordine, cache
angular:packages/common/http/src/backend.ts	`HttpInterceptorHandler`: catena costruita dall'injector root
projects/storefrontapp/src/app/app.config.ts	`provideHttpClient(withFetch(), withInterceptorsFromDi())`
projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts	Esempi di `provideConfig`/`provideConfigFactory` nell'app demo
projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts	Wrapper module lazy con estensioni (modello per override di facade)

3. Come funziona passo per passo

Le ricette sono undici. Per ogni simbolo `@spartacus/*` usato negli esempi è indicata la catena di export dal `public_api.ts` del pacchetto fino alla definizione. Tutti gli esempi sono **illustrativi e non presenti nel repo**: sono modellati sugli usi reali citati nella sezione 5. Nomi di classi/file inventati per gli esempi sono scritti in corsivo nel testo.

3.1 Ricetta 1 — Sovrascrivere la configurazione di Spartacus con `provideConfig`

Quando usarlo

Quando si vuole cambiare un valore di default di una libreria (URL del backend, route, `featureModules`, flag `pwa`, ecc.) dal codice dell'applicazione. È lo strumento standard: le librerie forniscono i propri valori come *default config*, l'applicazione come *root config*, e la root vince sempre — core-libs/core/src/config/config-tokens.ts:L10-L12, core-libs/core/src/config/config-tokens.ts:L49-L62.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/spartacus/custom-config.module.ts (esempio illustrativo, non presente nel repo)
import { NgModule } from '@angular/core';
import { CmsConfig, provideConfig } from '@spartacus/core';
import { ORDER_FEATURE } from '@spartacus/order/root';

@NgModule({
  providers: [
    // 1. override puntuale: deepMerge ricorsivo sugli oggetti
    provideConfig({
      backend: {
        occ: {
          baseUrl: 'https://api.example.com',
```

```
        },
      },
    }),
    // 2. override di una chiave di featureModules (cast tipizzato, come nell'app demo)
    provideConfig(<CmsConfig>{
      featureModules: {
        [ORDER_FEATURE]: {
          module: () =>
            import('./order/custom-order-wrapper.module').then(
              (m) => m.CustomOrderWrapperModule
            ),
        },
      },
    }),
  ],
})
export class CustomConfigModule {}
```

Il modulo va importato **dopo** i moduli feature che configurano le stesse chiavi (o i provider vanno messi nei `providers` dell'`AppModule`, elaborati dopo tutti gli import).

Catena di export:

- `provideConfig`: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → definizione core-libs/core/src/config/config-providers.ts:L18-L27.
- `CmsConfig`: core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L8-L8 → core-libs/core/src/cms/config/index.ts:L7-L7 → definizione core-libs/core/src/cms/config/cms-config.ts:L128-L129.
- `ORDER_FEATURE`: feature-libs/order/root/public_api.ts:L9-L9 → definizione feature-libs/order/root/feature-name.ts:L7-L7.

Perché funziona (file sorgente)

- `provideConfig(config)` restituisce un provider `multi: true` su `ConfigChunk` — core-libs/core/src/config/config-providers.ts:L18-L27.
- Angular aggiunge (`push`) ogni provider multi al record del token nell'ordine in cui li elabora — angular:packages/core/src/di/r3_injector.ts:L497-L521 — e per un NgModule elabora prima i provider degli `imports`, poi i propri `providers` — angular:packages/core/src/di/provider_collection.ts:L293-L321, angular:packages/core/src/di/provider_collection.ts:L345-L353.
- `RootConfig` fonde tutti i `ConfigChunk` nell'ordine dell'array, `DefaultConfig` tutti i `DefaultConfigChunk` — core-libs/core/src/config/config-tokens.ts:L23-L47 — e `Config = deepMerge({}, DefaultConfig, RootConfig)` — core-libs/core/src/config/config-tokens.ts:L10-L21.
- `deepMerge` ricorre sugli oggetti plain e sostituisce tutto il resto, array compresi — core-libs/core/src/config/utils/deep-merge.ts:L7-L49.
- Esempio reale di override per ordine: `WishListV2FeatureModule` importa `WishListFeatureModule` e ridefinisce nei propri `providers` la stessa chiave `featureModules[CART_WISH_LIST_FEATURE]` — projects/storefrontapp/src/app/spartacus/features/cart/wish-list-v2-feature.module.ts:L12-L27, projects/storefrontapp/src/app/spartacus/features/cart/wish-list-feature.module.ts:L35-L40.

Limiti e trappole

- Gli **array sono sostituiti**, non concatenati: un `provideConfig` con un array rimpiazza l'intero array di default — core-libs/core/src/config/utils/deep-merge.ts:L43-L45.
- Tra due `ConfigChunk` sulla stessa chiave vince l'**ultimo elaborato**; un modulo importato prima del modulo feature viene sovrascritto da quest'ultimo — angular:packages/core/src/di/provider_collection.ts:L293-L353. L'app demo lo dichiara esplicitamente per `TestConfigModule`: "Should be imported after other config modules" — projects/storefrontapp/src/app/private/private.providers.ts:L54-L54.
- Un `provideConfig` dentro un modulo **lazy** non può battere un `provideConfig` della root: nella config unificata `rootConfig` è applicato per ultimo — core-libs/core/src/config/services/configuration.service.ts:L92-L99.
- Non usare `ConfigChunk` direttamente: il JSDoc lo vieta — core-libs/core/src/config/config-tokens.ts:L49-L52.
- `ConfigModule.withConfig(...)` è equivalente (è solo un wrapper `ModuleWithProviders` di `provideConfig`) — core-libs/core/src/config/config.module.ts:L13-L66; ma i `ModuleWithProviders` sono elaborati dopo gli import semplici dello stesso modulo — angular:packages/core/src/di/provider_collection.ts:L293-L321.

3.2 Ricetta 2 — Config calcolata con `provideConfigFactory` senza mutare i default

Quando usarlo

Quando la config non è un letterale statico ma va derivata da un'altra config (tipicamente: partire dal layout di default e aggiungere/rimuovere slot) o richiede dipendenze iniettate. L'app demo fornisce così il layout: `provideConfigFactory(layoutConfigFactory)` — projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L44-L72.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/spartacus/custom-layout-config.ts (esempio illustrativo, non presente nel repo)
import { NgModule } from '@angular/core';
import { provideConfigFactory } from '@spartacus/core';
import {
  LayoutConfig,
  SlotConfig,
  layoutConfigFactory,
} from '@spartacus/storefront';

export function customLayoutConfigFactory(): LayoutConfig {
  // layoutConfigFactory() restituisce già una copia profonda della config statica
  const config = layoutConfigFactory();
  const landing = config.layoutSlots?.LandingPage2Template as
    | SlotConfig
    | undefined;
  if (landing?.slots) {
    // l'array verrà SOSTITUITO nel deepMerge: va fornito completo
    landing.slots = [...landing.slots, 'MyPromoSlot'];
  }
```

```
  return config;
}

@NgModule({
  providers: [provideConfigFactory(customLayoutConfigFactory)],
})
export class CustomLayoutConfigModule {}
```

Nell'app demo questo provider prenderebbe il posto di `provideConfigFactory(layoutConfigFactory)`; se convivono, vince quello elaborato per ultimo.

Catena di export:

- `provideConfigFactory`: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → definizione core-libs/core/src/config/config-providers.ts:L37-L47.
- `LayoutConfig`, `SlotConfig`: core-libs/storefront/public_api.ts:L26-L26 → core-libs/storefront/layout/index.ts:L9-L9 → core-libs/storefront/layout/config/index.ts:L8-L8 → definizioni core-libs/storefront/layout/config/layout-config.ts:L36-L46, core-libs/storefront/layout/config/layout-config.ts:L90-L102.
- `layoutConfigFactory`: core-libs/storefront/public_api.ts:L27-L27 → core-libs/storefront/recipes/index.ts:L7-L7 → core-libs/storefront/recipes/config/index.ts:L8-L8 → definizione core-libs/storefront/recipes/config/layout-config.ts:L190-L198.

Perché funziona (file sorgente)

- `provideConfigFactory(factory, deps?, defaultConfig = false)` produce un `FactoryProvider` multi su `ConfigChunk` — core-libs/core/src/config/config-providers.ts:L37-L47; `ConfigFactory` è `(...props: any[]) => Config` — core-libs/core/src/config/config-factory.ts:L9-L9.
- La factory viene eseguita quando `RootConfig` inietta tutti i `ConfigChunk` — core-libs/core/src/config/config-tokens.ts:L37-L39.
- `layoutConfigFactory` clona la config statica con `JSON.parse(JSON.stringify(...))` prima di modificarla — core-libs/storefront/recipes/config/layout-config.ts:L190-L198 — quindi modificare il risultato non tocca l'oggetto `layoutConfig` condiviso.

Limiti e trappole

- `deepMerge` **muta il target** e il `Config` globale viene ulteriormente modificato in place quando arrivano moduli lazy: non mutare oggetti statici condivisi, clonali — core-libs/core/src/config/services/configuration.service.ts:L102-L105, core-libs/storefront/recipes/config/layout-config.ts:L190-L191.
- La costante `layoutConfig` è `@deprecated` a favore di `layoutConfigFactory` — core-libs/storefront/recipes/config/layout-config.ts:L13-L27.
- In una **libreria** usare `provideDefaultConfigFactory`: il JSDoc lo raccomanda e la regola ESLint `use-provide-default-config-factory` del monorepo segnala qualsiasi chiamata a `provideConfigFactory`, anche con `defaultConfig = true` (il selettore guarda solo il nome della funzione) — core-libs/core/src/config/config-providers.ts:L29-L36, tools/eslint-rules/rules/use-provide-default-config-factory.ts:L41-L47.
- La factory gira dentro un contesto di injection durante la creazione di `Config`: iniettare (anche indirettamente) `Config` dalla factory crea una dipendenza circolare. DA VERIFICARE con test: deduzione da core-libs/core/src/config/config-tokens.ts:L10-L47 (nessun caso nel repo).

3.3 Ricetta 3 — Config tipizzata per il proprio codice (augmentation + default + validator)

Quando usarlo

Quando un proprio servizio/feature ha parametri configurabili e si vuole lo stesso modello delle librerie: una classe astratta iniettabile che è un alias di `Config`, un default sovrascrivibile dall'app, e un controllo in dev mode.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/shop-headers/shop-headers-config.ts (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import {
  Config,
  ConfigValidator,
  provideConfigValidator,
  provideDefaultConfig,
} from '@spartacus/core';

@Injectable({
  providedIn: 'root',
  useExisting: Config,
})
export abstract class ShopHeadersConfig {
  shopHeaders?: {
    headerName?: string;
    value?: string;
  };
}

declare module '@spartacus/core' {
  interface Config extends ShopHeadersConfig {}
}

export const defaultShopHeadersConfig: ShopHeadersConfig = {
  shopHeaders: {
    headerName: 'X-Shop-Channel',
    value: 'web',
  },
};

export const shopHeadersConfigValidator: ConfigValidator = (config) => {
  if (!config.shopHeaders?.headerName) {
    return 'Please configure shopHeaders.headerName';
  }
};

@NgModule({
  providers: [
```

```
    provideDefaultConfig(defaultShopHeadersConfig),
    provideConfigValidator(shopHeadersConfigValidator),
  ],
})
export class ShopHeadersConfigModule {}
```

L'applicazione può poi scrivere `provideConfig({ shopHeaders: { value: 'kiosk' } })` con tipizzazione completa.

Catena di export:

- `Config`: core-libs/core/public_api.ts:L40-L40 (blocco AUGMENTABLE_TYPES) e core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L10-L16 → definizione core-libs/core/src/config/config-tokens.ts:L17-L21.
- `provideDefaultConfig`: core-libs/core/src/config/index.ts:L9-L9 → definizione core-libs/core/src/config/config-providers.ts:L55-L61.
- `ConfigValidator`, `provideConfigValidator`: core-libs/core/src/config/index.ts:L17-L17 → core-libs/core/src/config/config-validator/index.ts:L7-L7 → definizioni core-libs/core/src/config/config-validator/config-validator.ts:L20-L20, core-libs/core/src/config/config-validator/config-validator.ts:L27-L35.

Perché funziona (file sorgente)

- Pattern identico a `CartConfig`: classe astratta `providedIn: 'root', useExisting: Config` + `declare module '@spartacus/core' { interface Config extends CartConfig {} }` — feature-libs/cart/base/root/config/cart-config.ts:L12-L32; e a `SegmentRefsConfig` con il suo default — integration-libs/segment-refs/root/config/segment-refs-config.ts:L10-L22, integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13.
- `Config` è esportato nel blocco dei tipi "augmentable" proprio per consentire la module augmentation — core-libs/core/public_api.ts:L38-L40.
- `provideDefaultConfig` punta sempre a `DefaultConfigChunk`, che perde contro qualsiasi `ConfigChunk` — core-libs/core/src/config/config-providers.ts:L55-L61, core-libs/core/src/config/config-tokens.ts:L10-L12.
- `provideConfigValidator(fn)` registra un multi provider su `ConfigValidatorToken`; `ConfigValidator = (config) => string | void` — core-libs/core/src/config/config-validator/config-validator.ts:L11-L35.

Limiti e trappole

- I validator girano **solo in dev mode**, dopo `getStable()`, e producono solo `logger.warn`: non bloccano l'app — core-libs/core/src/config/config-validator/config-validator.module.ts:L23-L37, core-libs/core/src/config/config-validator/config-validator.ts:L37-L48.
- Se il modulo che fornisce il default è **lazy**, il default arriva nel `Config` iniettato solo grazie al merge in place, disattivabile con `features.disableConfigUpdates` — core-libs/core/src/config/services/configuration.service.ts:L102-L105, core-libs/core/src/features-config/config/features-config.ts:L101-L116. È il caso reale di `OmfOrderModule` + `OmfConfig` — integration-libs/omf/order/omf-order.module.ts:L15-L26, integration-libs/omf/order/config/omf-config.ts:L11-L21.
- La module augmentation è globale: due augmentation con la stessa chiave e tipi incompatibili producono errori TypeScript.

3.4 Ricetta 4 — Leggere la config al momento giusto (`getStable`, `unifiedConfig$`)

Quando usarlo

- Codice che gira durante il bootstrap (un `APP_INITIALIZER`, il costruttore di un servizio istanziato presto) e che deve vedere la config **dopo** i `CONFIG_INITIALIZER` asincroni: usare `ConfigInitializerService.getStable(...)`.
- Codice che deve reagire alla config contribuita da moduli **lazy** (o che deve funzionare anche con `disableConfigUpdates` attivo): usare `ConfigurationService.unifiedConfig$`.

Esempio

Esempio illustrativo, non presente nel repo (usa l'augmentation della Ricetta 3).

```ts
// app/shop-headers/shop-headers-reader.service.ts (esempio illustrativo, non presente nel repo)
import { Injectable, inject } from '@angular/core';
import { ConfigInitializerService, ConfigurationService } from '@spartacus/core';
import { Observable } from 'rxjs';
import { distinctUntilChanged, map } from 'rxjs/operators';
import './shop-headers-config'; // porta in scope l'augmentation di Config

@Injectable({ providedIn: 'root' })
export class ShopHeadersReaderService {
  protected configInitializer = inject(ConfigInitializerService);
  protected configuration = inject(ConfigurationService);

  /** Valore alla fine dell'inizializzazione (una sola emissione). */
  readonly stableHeaderName$: Observable<string | undefined> =
    this.configInitializer
      .getStable('shopHeaders')
      .pipe(map((config) => config.shopHeaders?.headerName));

  /** Valore che si aggiorna se un modulo lazy contribuisce config. */
  readonly liveHeaderName$: Observable<string | undefined> =
    this.configuration.unifiedConfig$.pipe(
      map((config) => config.shopHeaders?.headerName),
      distinctUntilChanged()
    );
}
```

Catena di export:

- `ConfigInitializerService`: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L8-L8 → core-libs/core/src/config/config-initializer/index.ts:L14-L14 → definizione core-libs/core/src/config/config-initializer/config-initializer.service.ts:L21-L24.
- `ConfigurationService`: core-libs/core/src/config/index.ts:L19-L19 → definizione core-libs/core/src/config/services/configuration.service.ts:L21-L24.

Perché funziona (file sorgente)

- `getStable(...scopes)` è documentato come "Recommended way to get config for code that can run before app will finish initialization"; se la config è già stabile emette subito il `Config`, altrimenti attende che

gli scope richiesti non siano più in corso — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L46-L70.

- `ConfigInitializerModule.forRoot()` trattiene la navigazione iniziale finché la config non è stabile — core-libs/core/src/config/config-initializer/config-initializer.module.ts:L41-L68 — ed è già importato da `BaseCoreModule` — core-libs/core/src/base-core.module.ts:L30-L32.
- `unifiedConfig$` è un `BehaviorSubject` inizializzato con il `Config` globale che riemette `default < ambientDefault < ambient < root` quando un modulo lazy porta nuovi chunk — core-libs/core/src/config/services/configuration.service.ts:L25-L30, core-libs/core/src/config/services/configuration.service.ts:L48-L49, core-libs/core/src/config/services/configuration.service.ts:L57-L106.
- Modelli reali: `getStable('context')` in un `MODULE_INITIALIZER` del cart — feature-libs/cart/base/core/cart-persistence.module.ts:L19-L31; `CmsComponentsService` copia `cmsComponents` dalla config stabile — core-libs/storefront/cms-structure/services/cms-components.service.ts:L61-L68.

Limiti e trappole

- `getStable()` emette una volta sola: non vede config arrivata in seguito da moduli lazy — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L62-L69.
- Leggere `inject(Config)` una volta nel costruttore e memorizzare il valore perde i contributi lazy successivi; con `disableConfigUpdates` attivo il `Config` iniettato non cambia mai e il JSDoc impone `unifiedConfig$` — core-libs/core/src/features-config/config/features-config.ts:L109-L113.
- `disableConfigUpdates` è disattivo per default (`isFeatureEnabled` → `false` se assente) — core-libs/core/src/features-config/utils/feature-config-utils.ts:L48-L65.
- `initialize()` di `ConfigInitializerService` è `@internal`: non chiamarlo — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L129.

3.5 Ricetta 5 — Sostituire un servizio root eager (`useExisting` + sottoclasse)

Quando usarlo

Quando si vuole cambiare il comportamento di un servizio `providedIn: 'root'` che non è dietro una facade lazy, per esempio `AuthHttpHeaderService` (usato da `AuthInterceptor` per decidere a quali richieste aggiungere il token). Il codice stesso indica questo punto come quello da personalizzare: l'interceptor "only hooks into request send/received events" — core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L18-L27.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/auth/custom-auth-http-header.service.ts (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { AuthHttpHeaderService } from '@spartacus/core';

@Injectable({ providedIn: 'root' })
export class CustomAuthHttpHeaderService extends AuthHttpHeaderService {
  // Nessun costruttore: Angular usa la factory ereditata da AuthHttpHeaderService.

  /**
   * Tratta come "OCC" anche un secondo backend fidato, così riceve
```

```
   * il token e la gestione del refresh.
   */
  protected override isOccUrl(url: string): boolean {
    return super.isOccUrl(url) || url.startsWith('https://api2.example.com/');
  }
}

@NgModule({
  providers: [
    {
      provide: AuthHttpHeaderService,
      useExisting: CustomAuthHttpHeaderService,
    },
  ],
})
export class CustomAuthModule {}
```

CustomAuthModule va importato in un modulo **eager** (es. l'`AppModule`), non in un modulo lazy.

Catena di export:

- `AuthHttpHeaderService`: core-libs/core/public_api.ts:L11-L11 → core-libs/core/src/auth/index.ts:L9-L9 → core-libs/core/src/auth/user-auth/index.ts:L20-L20 → core-libs/core/src/auth/user-auth/services/index.ts:L10-L10 → definizione core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L87-L90.

Perché funziona (file sorgente)

- Un record esplicito nell'injector ha precedenza sulla definizione `@Injectable({providedIn})`: quest'ultima è usata solo se `records.get(token)` è `undefined` — angular:packages/core/src/di/r3_injector.ts:L347-L366.
- `useExisting` è un alias: chi inietta `AuthHttpHeaderService` riceve la stessa istanza di *CustomAuthHttpHeaderService* — angular:packages/core/src/di/r3_injector.ts:L657-L677.
- `isOccUrl` è `protected` ed è l'unico punto usato da `shouldCatchError`, `shouldAddAuthorizationHeader` e `alterRequest` — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L175-L207.
- Una sottoclasse `@Injectable` senza costruttore usa la factory del padre (`ɵɵgetInheritedFactory`) — angular:packages/core/src/render3/di.ts:L918-L945; lo stesso schema è usato nel repo da `InternalServerErrorHandler` — core-libs/core/src/global-message/http-interceptors/handlers/internal-server/internal-server.handler.ts:L13-L29.
- Modello reale: `AsmRootModule` sostituisce `AuthStorageService`, `AuthService`, `AuthHttpHeaderService` con `useExisting` verso sottoclassi `providedIn: 'root'`, nel root module eager — feature-libs/asm/root/asm-root.module.ts:L65-L87, feature-libs/asm/root/services/asm-auth-http-header.service.ts:L26-L60.

Limiti e trappole

- **Conflitto con ASM e Punchout**: anche `AsmRootModule` e `PunchoutRootModule` ridefiniscono `AuthHttpHeaderService` in root (Punchout con una `useFactory` che, secondo il toggle `enableExpiredRefreshTokenHandlers`, restituisce `PunchoutAuthHttpHeaderService` oppure il servizio fornito sotto `DELEGATED_AUTH_HTTP_HEADER_SERVICE`, con fallback su `DEFAULT_AUTH_HTTP_HEADER_SERVICE`); vince il provider elaborato per ultimo e gli altri vengono ignorati — feature-libs/asm/root/asm-root.module.ts:L73-L76, integration-libs/punchout/root/punchout.root.module.ts:L49-L66, integration-libs/

punchout/root/public_api.ts:L11-L11, angular:packages/core/src/di/r3_injector.ts:L513-L521. I token DEFAULT_AUTH_HTTP_HEADER_SERVICE/DELEGATED_AUTH_HTTP_HEADER_SERVICE esistono proprio per coordinare questi override e sono marcati come temporanei — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L52-L82. Se serve il comportamento ASM, la sottoclasse può estendere AsmAuthHttpHeaderService, che è pubblica — feature-libs/asm/root/public_api.ts:L14-L14 → feature-libs/asm/root/services/index.ts:L7-L7.

- L'override in un modulo **lazy** è visibile solo ai consumer risolti da quell'injector; AuthInterceptor (root) continuerebbe a usare l'istanza root. DA VERIFICARE a runtime: deduzione dalla gerarchia degli injector — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L85-L107, angular:packages/core/src/di/r3_injector.ts:L347-L366.
- Se si dichiara un costruttore, bisogna rilanciare a super(...) tutte e sette le dipendenze nell'ordine esatto, come fa ASM, anche se la sottoclasse le ottiene con inject() — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L157-L165, feature-libs/asm/root/services/asm-auth-http-header.service.ts:L33-L52. Per aggiungere dipendenze senza accoppiarsi alla firma, non dichiarare un costruttore e usare field inject() nella sottoclasse (protected x = inject(X)), lasciando ad Angular la factory ereditata — angular:packages/core/src/render3/di.ts:L918-L945.
- Estendere isOccUrl invia il token OAuth a un altro host: farlo solo per backend fidati.
- useClass invece di useExisting creerebbe un'istanza distinta da quella registrata sul token della sotto-classe — angular:packages/core/src/di/r3_injector.ts:L657-L677.

3.6 Ricetta 6 — Sostituire l'implementazione di una facade lazy (wrapper module)

Quando usarlo

Quando si vuole cambiare il comportamento di una facade astratta come OrderHistoryFacade o ActiveCartFacade, la cui implementazione vive nel chunk lazy della feature. Fornire la facade nell'injector root disattiverebbe il lazy loading (vedi trappole): l'override va fatto **dentro** il modulo lazy, con un wrapper module.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/spartacus/features/order/custom-order-wrapper.module.ts (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { OrderModule } from '@spartacus/order';
import { OrderHistoryService } from '@spartacus/order/core';
import { OrderHistoryFacade } from '@spartacus/order/root';

@Injectable()
export class CustomOrderHistoryService extends OrderHistoryService {
  // Nessun costruttore: factory ereditata da OrderHistoryService.

  override loadOrderList(
    pageSize: number,
    currentPage?: number,
    sort?: string
  ): void {
```

```
    super.loadOrderList(Math.min(pageSize, 20), currentPage, sort);
  }
}

@NgModule({
  // 1. prima il modulo feature (registra OrderHistoryFacade -> OrderHistoryService)
  imports: [OrderModule],
  // 2. poi i propri provider, elaborati DOPO gli imports: vincono
  providers: [
    CustomOrderHistoryService,
    {
      provide: OrderHistoryFacade,
      useExisting: CustomOrderHistoryService,
    },
  ],
})
export class CustomOrderWrapperModule {}
```

Il wrapper si collega con `featureModules[ORDER_FEATURE].module` come nella Ricetta 1, al posto di `OrderWrapperModule` dell'app demo.

Catena di export:

- `OrderModule`: feature-libs/order/public_api.ts:L16-L16 → definizione feature-libs/order/order.module.ts:L12-L15.
- `OrderHistoryService`: feature-libs/order/core/public_api.ts:L8-L8 → feature-libs/order/core/facade/index.ts:L7-L7 → definizione feature-libs/order/core/facade/order-history.service.ts:L29-L37.
- `OrderHistoryFacade`: feature-libs/order/root/public_api.ts:L8-L8 → feature-libs/order/root/facade/index.ts:L7-L7 → definizione feature-libs/order/root/facade/order-history.facade.ts:L45-L49.

Perché funziona (file sorgente)

- La facade root è una classe astratta `providedIn: 'root'` con `useFactory: orderHistoryFacadeFactory`, che crea un proxy con `facadeFactory({ feature: ORDER_CORE_FEATURE, methods: [...], async: true })`; `loadOrderList` è tra i `methods` — feature-libs/order/root/facade/order-history.facade.ts:L19-L49.
- Il proxy risolve la feature e prende l'implementazione con `moduleRef.injector.get(facadeClass)`, cioè dall'injector del modulo lazy — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57.
- `ORDER_CORE_FEATURE` è alias di `ORDER_FEATURE`, che dipende da `CART_BASE_FEATURE` — feature-libs/order/root/order-root.module.ts:L68-L80; gli alias stringa sono risolti da `FeatureModulesService` — core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100.
- `OrderModule` importa `OrderCoreModule`, che registra `OrderHistoryService` e `{ provide: OrderHistoryFacade, useExisting: OrderHistoryService }` — feature-libs/order/order.module.ts:L12-L15, feature-libs/order/core/order-core.module.ts:L17-L29, feature-libs/order/core/facade/facade-providers.ts:L33-L37. I `providers` del wrapper sono elaborati dopo gli import e sostituiscono il record (single provider) — angular:packages/core/src/di/provider_collection.ts:L345-L353, angular:packages/core/src/di/r3_injector.ts:L513-L521.

- Modelli reali: `OrderWrapperModule` importa `OrderModule` e poi le estensioni, fra cui `OmfOrderModule`, che ridefinisce `OrderHistoryFacade` con `useClass: OmfOrderHistoryService` — projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27, integration-libs/omf/order/omf-order.module.ts:L15-L26; dentro una libreria, `WishListWithV2Module = imports: [WishListModule, WishListV2Module]` — feature-libs/cart/wish-list/wish-list-with-v2.module.ts:L11-L14, feature-libs/cart/wish-list/wish-list-v2.module.ts:L20-L50.

Limiti e trappole

- **Non** fornire `OrderHistoryFacade` nell'injector root: il record esplicito disattiva `useFactory` (niente proxy), la classe finisce nel bundle iniziale e le sue dipendenze (store della feature) non sono registrate in root — angular:packages/core/src/di/r3_injector.ts:L347-L366. Il repo non contiene esempi di questo approccio.
- Il proxy espone **solo** i membri elencati in `methods`/`properties`: un metodo nuovo o un metodo non elencato non è raggiungibile dai consumer root; metodi con ritorno non-Observable restituiscono `EMPTY`. Esempio reale: `getQueryParams` non è in `methods` ed è `@deprecated` per questo, ma è proprio il metodo ridefinito da OMF — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L84-L131, feature-libs/order/root/facade/order-history.facade.ts:L144-L148, integration-libs/omf/order/omf-order-history.service.ts:L12-L24.
- Ordine degli `imports`: un'estensione che ridefinisce la facade deve venire **dopo** il modulo feature; se nel wrapper si importano anche `OmfOrderModule` o altre estensioni che ridefiniscono `OrderHistoryFacade`, vince l'ultimo elaborato (i `providers` del wrapper restano comunque dopo tutti gli import) — projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L24-L26.
- `useClass` vs `useExisting`: OMF usa `useClass` e registra anche la classe, ottenendo due istanze distinte; `useExisting` (cart, wish-list, questo esempio) condivide l'istanza — integration-libs/omf/order/omf-order.module.ts:L17-L23.
- Chiamare un metodo della facade scarica subito il chunk (`connect()`), anche senza subscribe — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L92.

3.7 Ricetta 7 — Interceptor HTTP custom

Quando usarlo

Quando si deve modificare ogni richiesta (header, query param) o osservare ogni risposta, in modo indipendente dagli adapter OCC. Per cambiare solo *quali* richieste ricevono il token, preferire la Ricetta 5; per reagire agli errori, la Ricetta 8.

Esempio

Esempio illustrativo, non presente nel repo (usa la config della Ricetta 3).

```ts
// app/shop-headers/shop-headers.interceptor.ts (esempio illustrativo, non presente nel repo)
import {
  HTTP_INTERCEPTORS,
  HttpEvent,
  HttpHandler,
  HttpInterceptor,
  HttpRequest,
} from '@angular/common/http';
import { Injectable, NgModule, Provider, inject } from '@angular/core';
```

```
import { OccEndpointsService, WindowRef } from '@spartacus/core';
import { Observable } from 'rxjs';
import { ShopHeadersConfig } from './shop-headers-config';

@Injectable({ providedIn: 'root' })
export class ShopHeadersInterceptor implements HttpInterceptor {
  protected config = inject(ShopHeadersConfig);
  protected occEndpoints = inject(OccEndpointsService);
  protected winRef = inject(WindowRef);

  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    const headerName = this.config.shopHeaders?.headerName;
    const value = this.config.shopHeaders?.value;
    if (
      this.winRef.isBrowser() &&
      headerName &&
      value &&
      request.url.includes(this.occEndpoints.getBaseUrl())
    ) {
      request = request.clone({ setHeaders: { [headerName]: value } });
    }
    return next.handle(request);
  }
}

export const shopHeadersInterceptors: Provider[] = [
  {
    provide: HTTP_INTERCEPTORS,
    useExisting: ShopHeadersInterceptor,
    multi: true,
  },
];

@NgModule({
  providers: [...shopHeadersInterceptors],
})
export class ShopHeadersModule {}
```

ShopHeadersModule va importato in un modulo eager e l'app deve avere `provideHttpClient(withInterceptorsFromDi())`.

Catena di export:

- OccEndpointsService: core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L16-L16 → core-libs/core/src/occ/services/index.ts:L7-L7 → definizione core-libs/core/src/occ/services/occ-endpoints.service.ts:L34-L34.

- `WindowRef`: core-libs/core/public_api.ts:L36-L36 → core-libs/core/src/window/index.ts:L7-L7 → definizione core-libs/core/src/window/window-ref.ts:L16-L16.
- `HTTP_INTERCEPTORS`: `@angular/common/http`, definito in angular:packages/common/http/src/interceptor.ts:L195-L197.

Perché funziona (file sorgente)

- L'app demo configura `provideHttpClient(withFetch(), withInterceptorsFromDi())` — projects/storefrontapp/src/app/app.config.ts:L26-L35 — e lo schematic `add-spartacus` aggiunge la stessa riga — core-libs/schematics/src/add-spartacus/update-app-config.ts:L96-L100. `withInterceptorsFromDi` include gli interceptor class-based del multi provider `HTTP_INTERCEPTORS` — angular:packages/common/http/src/provider.ts:L170-L194.
- La funzione legacy legge `HTTP_INTERCEPTORS` una volta e avvolge la catena right-to-left, così il primo interceptor registrato è il più esterno — angular:packages/common/http/src/interceptor.ts:L227-L240; la catena è costruita e memorizzata da `HttpInterceptorHandler` con il proprio `EnvironmentInjector` — angular:packages/common/http/src/backend.ts:L57-L66, angular:packages/common/http/src/backend.ts:L101-L122.
- Modello reale: `OccSegmentRefsInterceptor` (`providedIn: 'root'`, filtro OCC con `getBaseUrl()`, `request.clone({ setHeaders })`, guardia `winRef.isBrowser()`) registrato con `useExisting` in un array di provider spalmato nel root module — integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L18-L86, integration-libs/segment-refs/root/http-interceptors/index.ts:L11-L17, integration-libs/segment-refs/root/segment-refs-root.module.ts:L12-L20.

Limiti e trappole

- Interceptor forniti in un modulo **lazy** non vengono applicati: la catena viene dall'injector root ed è in cache dalla prima richiesta. Tutti gli interceptor del repo sono in moduli root/eager — angular:packages/common/http/src/backend.ts:L101-L122, angular:packages/common/http/src/interceptor.ts:L227-L240.
- Senza `withInterceptorsFromDi()` non funzionano né gli interceptor Spartacus né quelli class-based custom; Angular avverte che il supporto DI "may be phased out" — angular:packages/common/http/src/provider.ts:L170-L174.
- Ordine: un interceptor nei `providers` dell'`AppModule` è elaborato dopo quelli dei moduli importati, quindi più interno (vede la richiesta già modificata, es. con l'header di autenticazione) — angular:packages/core/src/di/r3_injector.ts:L497-L512, angular:packages/common/http/src/interceptor.ts:L237-L240. Nella demo l'ordine di registrazione è [`HttpErrorInterceptor`, ..., `AuthInterceptor`, ..., custom]: `HttpErrorInterceptor` arriva da `GlobalMessageModule.forRoot()` in `BaseCoreModule` (via `BaseStorefrontModule`), importato prima di `SpartacusFeaturesModule` che contiene `AuthModule.forRoot()` — core-libs/core/src/global-message/global-message.module.ts:L27-L30, core-libs/core/src/base-core.module.ts:L35-L35, projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L13-L16, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L179-L179. Quindi `HttpErrorInterceptor` è il più esterno dei tre.
- `getBaseUrl()` per default include `baseUrl + prefix + baseSite`: richieste OCC senza base site (es. `baseSites`) non passano il filtro — core-libs/core/src/occ/services/occ-endpoints.service.ts:L85-L101.
- Gli interceptor girano anche in SSR: proteggere il codice browser-only con `winRef.isBrowser()` — integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L53-L59, integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L73-L77.
- Non tutto è estendibile: `@spartacus/segment-refs/root` esporta solo `SegmentRefsRootModule`, quindi `OccSegmentRefsInterceptor` non è importabile dall'app — integration-libs/segment-refs/root/public_api.ts:L7-L7.

- Per un parametro "per richiesta" Spartacus usa `HttpContextToken` (`OCC_HTTP_TOKEN`, `HTTP_TIMEOUT_CONFIG`) letti dall'interceptor con `request.context.get(...)` — core-libs/core/src/occ/utils/occ-http-token.ts:L9-L20, feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L29-L84.

3.8 Ricetta 8 — Gestione custom di un errore HTTP con `HttpErrorHandler`

Quando usarlo

Quando si vuole cambiare la reazione a uno status HTTP (messaggio, redirect, logout) per tutte le chiamate. A differenza di un interceptor, un `HttpErrorHandler` funziona anche se fornito in un modulo lazy, e si integra con la scelta per priorità degli handler core.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/errors/custom-conflict.handler.ts (esempio illustrativo, non presente nel repo)
import { HttpErrorResponse, HttpRequest } from '@angular/common/http';
import { Injectable, NgModule } from '@angular/core';
import {
  GlobalMessageType,
  HttpErrorHandler,
  HttpResponseStatus,
  Priority,
} from '@spartacus/core';

@Injectable({ providedIn: 'root' })
export class CustomConflictHandler extends HttpErrorHandler {
  // Nessun costruttore: factory ereditata (GlobalMessageService, PLATFORM_ID).
  responseStatus = HttpResponseStatus.CONFLICT;

  getPriority(): Priority {
    // ConflictHandler di core usa LOW, ma alcune feature lazy (organization)
    // registrano handler 409 NORMAL: HIGH prevale in modo affidabile
    return Priority.HIGH;
  }

  hasMatch(errorResponse: HttpErrorResponse): boolean {
    return (
      super.hasMatch(errorResponse) && !this.isSsr()
    );
  }

  handleError(_request: HttpRequest<any>, _response: HttpErrorResponse): void {
    this.globalMessageService.add(
      { key: 'myErrors.conflict' },
      GlobalMessageType.MSG_TYPE_ERROR
    );
  }
}
```

```
@NgModule({
  providers: [
    {
      provide: HttpErrorHandler,
      useExisting: CustomConflictHandler,
      multi: true,
    },
  ],
})
export class CustomErrorHandlingModule {}
```

Catena di export (tutti da core-libs/core/public_api.ts:L21-L21 → core-libs/core/src/global-message/index.ts, salvo Priority):

- HttpErrorHandler: core-libs/core/src/global-message/index.ts:L10-L10 → core-libs/core/src/global-message/http-interceptors/handlers/index.ts:L12-L12 → definizione core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L12-L19.
- GlobalMessageType: core-libs/core/src/global-message/index.ts:L12-L12 → definizione core-libs/core/src/global-message/models/global-message.model.ts:L9-L13.
- HttpResponseStatus: core-libs/core/src/global-message/index.ts:L13-L13 → definizione core-libs/core/src/global-message/models/response-status.model.ts:L7-L17.
- Priority: core-libs/core/public_api.ts:L35-L35 → core-libs/core/src/util/index.ts:L7-L7 → definizione core-libs/core/src/util/applicable.ts:L7-L24.

Perché funziona (file sorgente)

- HttpErrorInterceptor raccoglie gli handler con UnifiedInjector.getMulti(HttpErrorHandler) (root + moduli lazy) — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L22-L28, core-libs/core/src/lazy-loading/unified-injector.ts:L75-L93 — e sceglie con resolveApplicable(handlers, [response]) — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L59-L65.
- resolveApplicable filtra per hasMatch e sceglie la priorità più alta; a parità vince il provider registrato per ultimo — core-libs/core/src/util/applicable.ts:L47-L87. Gli handler core hanno priorità LOW (UnknownErrorHandler: FALLBACK) — core-libs/core/src/global-message/http-interceptors/handlers/conflict/conflict.handler.ts:L26-L28, core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L48-L50.
- Per prevalere in modo affidabile serve Priority.HIGH: OrganizationUserRegistrationConflictHandler (che non ridefinisce hasMatch, quindi accetta ogni 409) e OrganizationConflictHandler usano Priority.NORMAL e sono registrati nei moduli core di organization, caricati lazy; a parità di priorità vince l'ultimo registrato e UnifiedInjector.getMulti accoda gli handler lazy dopo quelli root — feature-libs/organization/user-registration/core/http-interceptors/conflict/conflict.handler.ts:L20-L39, feature-libs/organization/user-registration/core/user-registration-core.module.ts:L23-L23, feature-libs/organization/administration/core/http-interceptors/conflict/conflict.handler.ts:L95-L95, core-libs/core/src/lazy-loading/unified-injector.ts:L82-L92, core-libs/core/src/util/applicable.ts:L57-L57.
- HttpErrorHandler fornisce hasMatch per status, isSsr() e globalMessageService — core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L12-L72.

- Modello reale importato solo da `@spartacus/core`: `BadCartRequestHandler`, registrato nel modulo lazy `CartBaseCoreModule` — feature-libs/cart/base/core/http-interceptors/handlers/bad-cart-request.handler.ts:L9-L38, feature-libs/cart/base/core/cart-base-core.module.ts:L37-L41.

Limiti e trappole

- `UnifiedInjector.getMulti` lancia un errore se lo stesso token mescola multi e single provider: fornire sempre `HttpErrorHandler` con `multi: true` — core-libs/core/src/lazy-loading/unified-injector.ts:L75-L93.
- Un handler in un modulo lazy è considerato solo dopo il caricamento del modulo — core-libs/core/src/lazy-loading/unified-injector.ts:L75-L93.
- L'interceptor non intercetta l'errore: lo osserva con `tap({ error })` e lo lascia propagare al chiamante — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L30-L43.
- `getPriority()` è chiamato senza argomenti (l'interceptor passa solo `matchParams`) — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L62-L64.
- Le chiavi di traduzione usate in `handleError` devono esistere nelle risorse i18n (vedi 53).

3.9 Ricetta 9 — Evento custom: definire, emettere, ascoltare

Quando usarlo

Per notificare fatti "una tantum" tra parti disaccoppiate dell'app (analytics, tracking, reload di query) senza creare dipendenze dirette tra servizi. Per stato persistente usare facade/query, non eventi.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/events/promo.events.ts + listener (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule, OnDestroy, inject } from '@angular/core';
import { CxEvent, EventService, LogoutEvent } from '@spartacus/core';
import { Subscription, merge } from 'rxjs';

export class PromoBannerClickedEvent extends CxEvent {
  static readonly type = 'PromoBannerClickedEvent';
  bannerId!: string;
}

/** Produttore: da un componente o servizio. */
export function trackPromoClick(eventService: EventService, bannerId: string) {
  eventService.dispatch({ bannerId }, PromoBannerClickedEvent);
}

/** Consumer: servizio root attivato da un modulo eager. */
@Injectable({ providedIn: 'root' })
export class PromoEventListener implements OnDestroy {
  protected eventService = inject(EventService);
  protected subscriptions = new Subscription();

  constructor() {
```

```
    this.subscriptions.add(
      merge(
        this.eventService.get(PromoBannerClickedEvent),
        this.eventService.get(LogoutEvent)
      ).subscribe((event) => {
        // es. invio a un sistema di analytics
        console.log(event);
      })
    );
  }

  ngOnDestroy(): void {
    this.subscriptions.unsubscribe();
  }
}

@NgModule({})
export class PromoEventModule {
  constructor(_promoEventListener: PromoEventListener) {}
}
```

Catena di export:

- `CxEvent`, `EventService`: `core-libs/core/public_api.ts:L18-L18` → `core-libs/core/src/event/index.ts:L7-L8` → definizioni `core-libs/core/src/event/cx-event.ts:L12-L17`, `core-libs/core/src/event/event.service.ts:L44-L47`.
- `LogoutEvent`: `core-libs/core/public_api.ts:L11-L11` → `core-libs/core/src/auth/index.ts:L9-L9` → `core-libs/core/src/auth/user-auth/index.ts:L10-L10` → `core-libs/core/src/auth/user-auth/events/index.ts:L9-L9` → definizione `core-libs/core/src/auth/user-auth/events/user-auth.events.ts:L15-L15`.

Perché funziona (file sorgente)

- `CxEvent` è la base astratta con `static readonly type` — `core-libs/core/src/event/cx-event.ts:L7-L17`.
- `dispatch(event, type)`: se `event` non è istanza del tipo la crea con `createFrom`, poi fa `next` su un `Subject` di input creato al primo uso e registrato come sorgente — `core-libs/core/src/event/event.service.ts:L105-L128`; `createFrom = Object.assign(new type(), data)` — `core-libs/core/src/util/create-from.ts:L15-L17`.
- `get(type)` restituisce l'`output$` del `MergingSubject`, con validazione in dev mode — `core-libs/core/src/event/event.service.ts:L90-L96`; ascoltare un padre riceve anche i sottotipi — `core-libs/core/src/event/event.service.ts:L144-L159`.
- Modello reale: eventi minimali `extends CxEvent` — `feature-libs/subscription-billing/root/events/subscription-billing.events.ts:L7-L23`; listener root che combina eventi core, dispatcha con `dispatch({}, Tipo)` e chiude la `Subscription` — `feature-libs/subscription-billing/root/events/subscription-billing-event.listener.ts:L19-L45`; attivato da un modulo che lo inietta nel costruttore — `feature-libs/subscription-billing/root/events/subscription-billing-event.module.ts:L8-L13`.
- Un evento può essere trigger di una query: `QueryNotifier = Observable<unknown> | Type<CxEvent>` e `reloadOn: [Tipo]` usa `eventService.get(trigger)` — `core-libs/core/src/util/command-query/query.service.ts:L31-L31`, `core-libs/core/src/util/command-query/query.service.ts:L144-L154`; esempio

reale in `SubscriptionService` — feature-libs/subscription-billing/core/facade/subscription.service.ts:L33-L57.

Limiti e trappole

- Il bus non fa replay: un evento senza consumer attivi va perso — core-libs/core/src/event/event.service.ts:L120-L128, core-libs/core/src/event/utils/merging-subject.ts:L45-L80. Il listener deve essere istanziato prima che l'evento sia emesso (da qui il modulo che lo inietta nel costruttore).
- `createFrom` chiama `new Tipo()` senza argomenti: niente costruttori con parametri negli eventi — core-libs/core/src/util/create-from.ts:L15-L17.
- Un evento che non estende `CxEvent` produce solo un warning in dev mode — core-libs/core/src/event/event.service.ts:L166-L197.
- `static readonly type` è una convenzione delle classi evento. `EventService` lo usa solo per riconoscere `CxEvent` lungo la catena dei prototipi (confronto con `CxEvent.type`): un evento custom non deve quindi dichiarare `type = 'CxEvent'`. I messaggi di validazione usano il nome della classe (`eventType.name`) e i tipi sono distinti per riferimento di classe, non per stringa — core-libs/core/src/event/event.service.ts:L183-L196, core-libs/core/src/event/event.service.ts:L133-L141.

3.10 Ricetta 10 — Evento derivato da un altro evento (`EventService.register`)

Quando usarlo

Quando un evento può essere calcolato da un evento esistente (tipicamente `NavigationEvent`) filtrando o arricchendo: si registra una **sorgente** invece di fare `dispatch` manuale, così lo stream è attivo solo se qualcuno ascolta.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/events/order-history-page-event.builder.ts (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule, inject } from '@angular/core';
import { CxEvent, EventService, createFrom } from '@spartacus/core';
import { NavigationEvent, NavigationEventModule } from '@spartacus/storefront';
import { Observable } from 'rxjs';
import { filter, map } from 'rxjs/operators';

export class OrderHistoryPageEvent extends CxEvent {
  static readonly type = 'OrderHistoryPageEvent';
  url!: string;
}

@Injectable({ providedIn: 'root' })
export class OrderHistoryPageEventBuilder {
  protected eventService = inject(EventService);

  constructor() {
    // servizio root: vive quanto l'app, il teardown non serve
    this.eventService.register(OrderHistoryPageEvent, this.build());
  }
```

```
  protected build(): Observable<OrderHistoryPageEvent> {
    return this.eventService.get(NavigationEvent).pipe(
      filter((navigation) => navigation.semanticRoute === 'orders'),
      map((navigation) =>
        createFrom(OrderHistoryPageEvent, { url: navigation.url })
      )
    );
  }
}

@NgModule({
  imports: [NavigationEventModule], // senza, NavigationEvent non ha sorgenti
})
export class OrderHistoryPageEventModule {
  constructor(_builder: OrderHistoryPageEventBuilder) {}
}
```

Catena di export:

- `createFrom`: core-libs/core/public_api.ts:L35-L35 → core-libs/core/src/util/index.ts:L12-L12 → definizione core-libs/core/src/util/create-from.ts:L15-L17.
- `CxEvent`, `EventService`: come Ricetta 9.
- `NavigationEvent`, `NavigationEventModule`: core-libs/storefront/public_api.ts:L25-L25 → core-libs/storefront/events/index.ts:L8-L8 → core-libs/storefront/events/navigation/index.ts:L8-L9 → definizione core-libs/storefront/events/navigation/navigation.event.ts:L13-L33.

Perché funziona (file sorgente)

- `register(type, source$)` aggiunge la sorgente al `MergingSubject` del tipo (con warning se già registrata) e restituisce il teardown — core-libs/core/src/event/event.service.ts:L54-L84.
- `MergingSubject.output$` si abbona alle sorgenti solo quando c'è almeno un consumer e se ne stacca a zero consumer — core-libs/core/src/event/utils/merging-subject.ts:L45-L80.
- `NavigationEvent` porta `context`, `semanticRoute?`, `url`, `params` — core-libs/storefront/events/navigation/navigation.event.ts:L13-L33; la route semantica `orders` è definita nella config di routing di default dell'order — feature-libs/order/root/config/default-order-routing-config.ts:L12-L12.
- Modello reale: `CartPageEventBuilder` registra `CartPageEvent` filtrando `NavigationEvent` su `semanticRoute === 'cart'` — feature-libs/cart/base/core/event/cart-page-event.builder.ts:L14-L35.

Limiti e trappole

- Senza `NavigationEventModule` importato, `NavigationEvent` non ha sorgenti e il builder non emette mai; l'app demo lo importa esplicitamente — projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L240-L242.
- Chiamare `build()` due volte e registrare due observable distinti crea due sorgenti ed eventi duplicati: la deduplicazione è per identità dell'observable — core-libs/core/src/event/event.service.ts:L70-L81.
- Se la sorgente vive meno dell'app (es. registrata in un componente) il teardown va chiamato in `ngOnDestroy` — core-libs/core/src/event/event.service.ts:L57-L59.

3.11 Ricetta 11 — Evento da action NgRx (`StateEventService`)

Quando usarlo

Quando un fatto interessante è già rappresentato da una action NgRx ma non esiste un evento corrispondente. Esempio: tracciare i messaggi globali mostrati all'utente, dato che nel repo nessun evento è mappato su `GlobalMessageActions.ADD_MESSAGE` (la stringa compare solo in effect e reducer) — core-libs/core/src/global-message/store/effects/global-message.effect.ts:L34-L34, core-libs/core/src/global-message/store/reducers/global-message.reducer.ts:L24-L24.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/events/global-message-shown.event.ts (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule, inject } from '@angular/core';
import {
  CxEvent,
  GlobalMessage,
  GlobalMessageActions,
  StateEventService,
  createFrom,
} from '@spartacus/core';

export class GlobalMessageShownEvent extends CxEvent {
  static readonly type = 'GlobalMessageShownEvent';
  message!: GlobalMessage;
}

@Injectable({ providedIn: 'root' })
export class GlobalMessageEventBuilder {
  protected stateEventService = inject(StateEventService);

  constructor() {
    this.stateEventService.register({
      action: GlobalMessageActions.ADD_MESSAGE,
      event: GlobalMessageShownEvent,
      // senza factory il payload (GlobalMessage) verrebbe copiato
      // direttamente nei campi dell'evento
      factory: (action: GlobalMessageActions.AddMessage) =>
        createFrom(GlobalMessageShownEvent, { message: action.payload }),
    });
  }
}

@NgModule({})
export class GlobalMessageEventModule {
  constructor(_builder: GlobalMessageEventBuilder) {}
}
```

Catena di export:

- StateEventService: core-libs/core/public_api.ts:L33-L33 → core-libs/core/src/state/index.ts:L8-L8 → core-libs/core/src/state/event/index.ts:L7-L8 → definizione core-libs/core/src/state/event/state-event.service.ts:L19-L22.
- GlobalMessageActions: core-libs/core/public_api.ts:L21-L21 → core-libs/core/src/global-message/index.ts:L14-L14 → core-libs/core/src/global-message/store/actions/index.ts:L7-L8 → core-libs/core/src/global-message/store/actions/global-message-group.actions.ts:L7-L7 → definizioni core-libs/core/src/global-message/store/actions/global-message.actions.ts:L13-L21.
- GlobalMessage: core-libs/core/src/global-message/index.ts:L12-L12 → definizione core-libs/core/src/global-message/models/global-message.model.ts:L17-L21.
- CxEvent, createFrom: come Ricette 9 e 10.

Perché funziona (file sorgente)

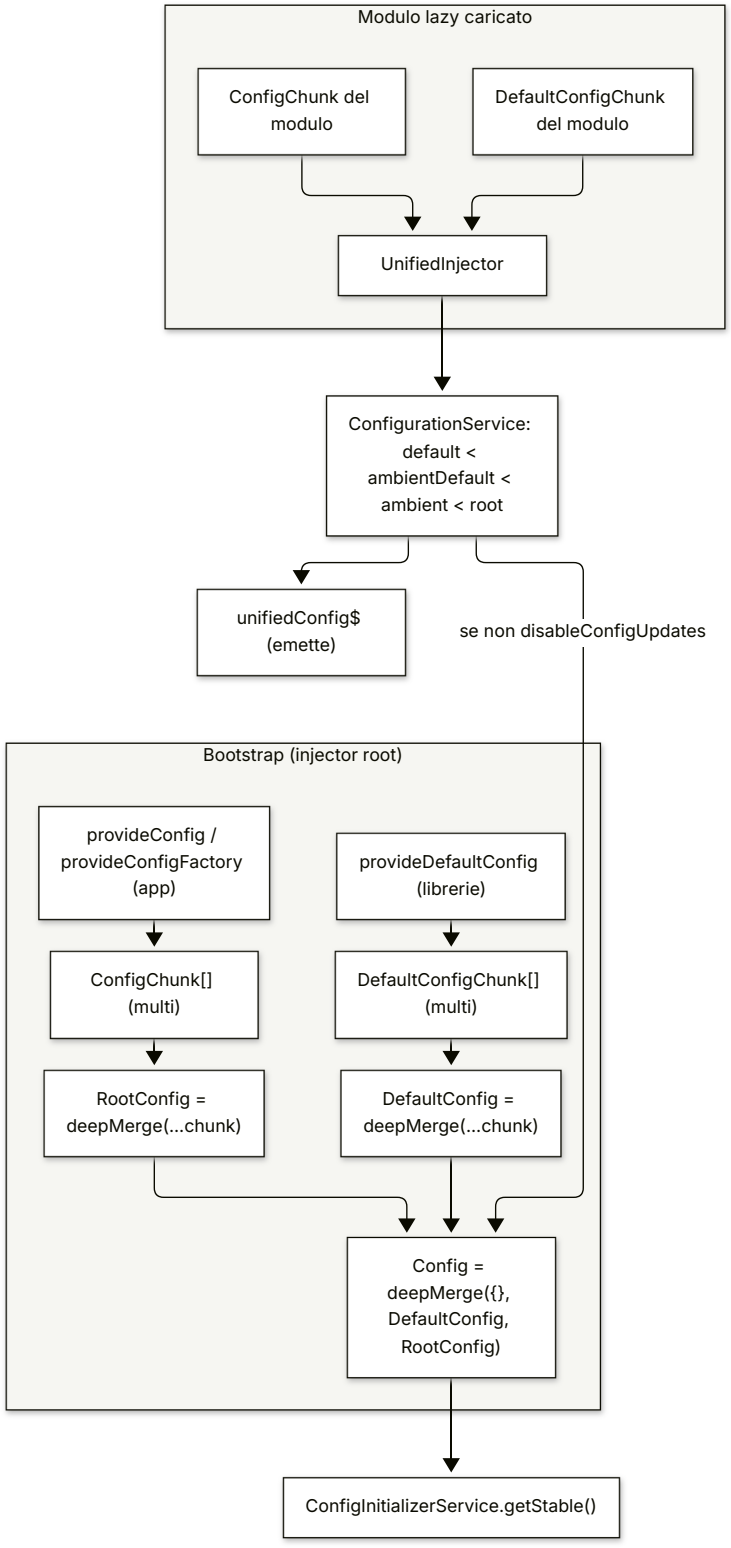
- StateEventService.register(mapping) registra su EventService lo stream actionsSubject.pipe(ofType(...), map(createEvent)); senza factory usa createFrom(eventType, action.payload ?? {}) — core-libs/core/src/state/event/state-event.service.ts:L36-L76.
- ActionToEventMapping<T> = { action: string | string[]; event: Type<T> | null; factory? } — core-libs/core/src/state/event/action-to-event-mapping.ts:L9-L24.
- GlobalMessageService.add dispatcha GlobalMessageActions.AddMessage con payload GlobalMessage — core-libs/core/src/global-message/facade/global-message.service.ts:L43-L54, core-libs/core/src/global-message/store/actions/global-message.actions.ts:L13-L21.
- Modelli reali: mapping senza factory in UserEventBuilder — core-libs/core/src/user/events/user-event.builder.ts:L16-L39; con factory in CartEventBuilder — feature-libs/cart/base/core/event/cart-event.builder.ts:L136-L145.

Limiti e trappole

- ActionsSubject di NgRx è un BehaviorSubject: il primo consumer di un evento mappato può ricevere subito l'evento se l'ultima action dispatchata corrisponde. DA VERIFICARE con test: deduzione da deps:ngrx-store-21.0.1/package/fesm2022/ngrx-store.mjs:L196-L200 e core-libs/core/src/state/event/state-event.service.ts:L47-L54.
- Mappare action di una feature lazy importando il suo pacchetto core (es. CartActions da @spartacus/cart/base/core) in un modulo eager porta quel codice nel bundle iniziale; il repo registra gli eventi di cart dentro il chunk lazy (CartEventModule in CartBaseCoreModule) — feature-libs/cart/base/core/event/cart-event.module.ts:L10-L15, feature-libs/cart/base/core/cart-base-core.module.ts:L22-L28. Le action lazy esistono comunque solo dopo il caricamento del chunk.
- Senza factory, action senza payload producono eventi vuoti — core-libs/core/src/state/event/state-event.service.ts:L73-L75.
- Le action sono un dettaglio dello store: preferire eventi già esistenti quando ci sono (es. CartAddEntrySuccessEvent da @spartacus/cart/base/root) — feature-libs/cart/base/root/public_api.ts:L10-L10, feature-libs/cart/base/root/events/index.ts:L8-L9.

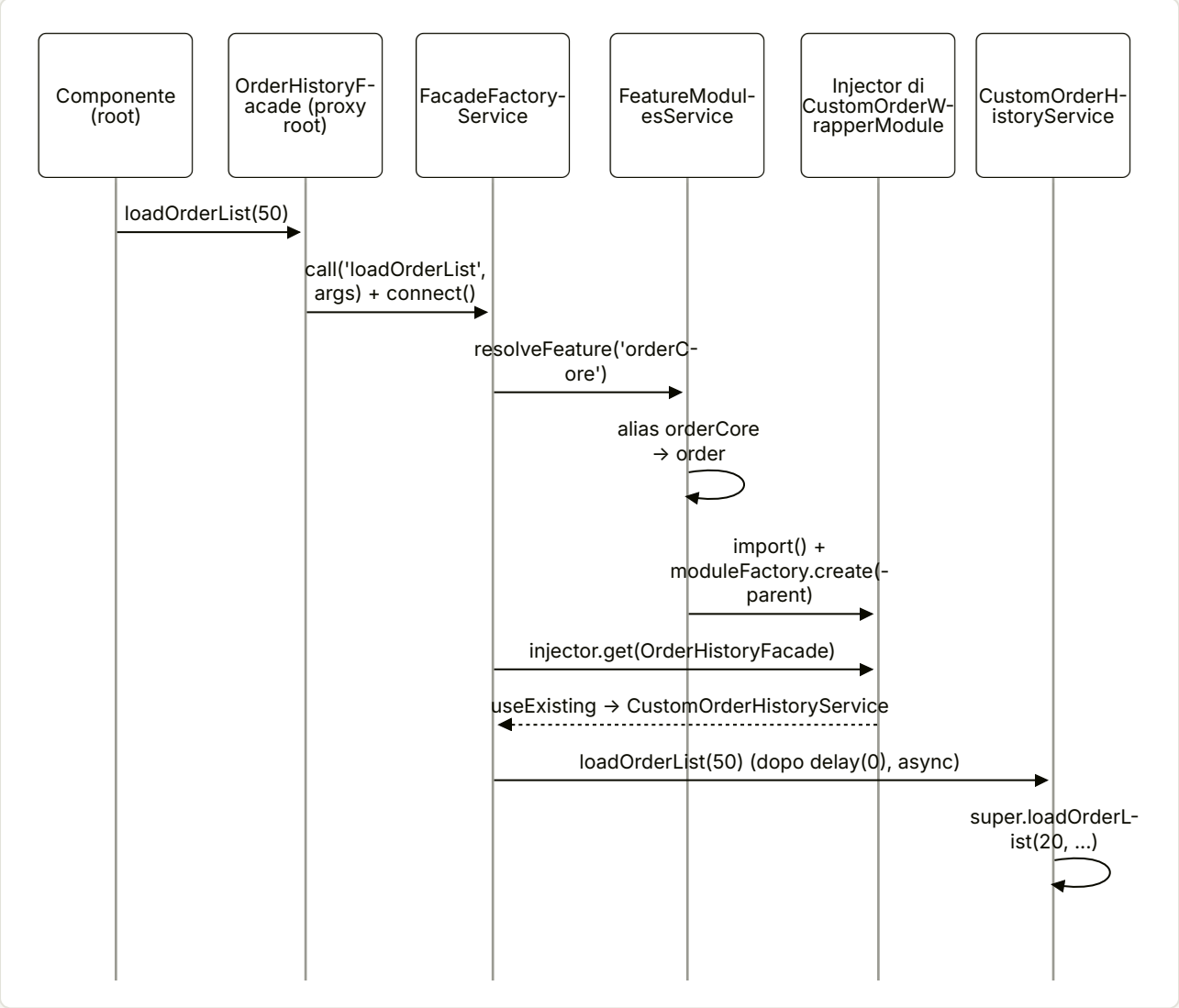
4. Diagrammi

4.1 Da dove arriva un valore di config (Ricette 1-4)



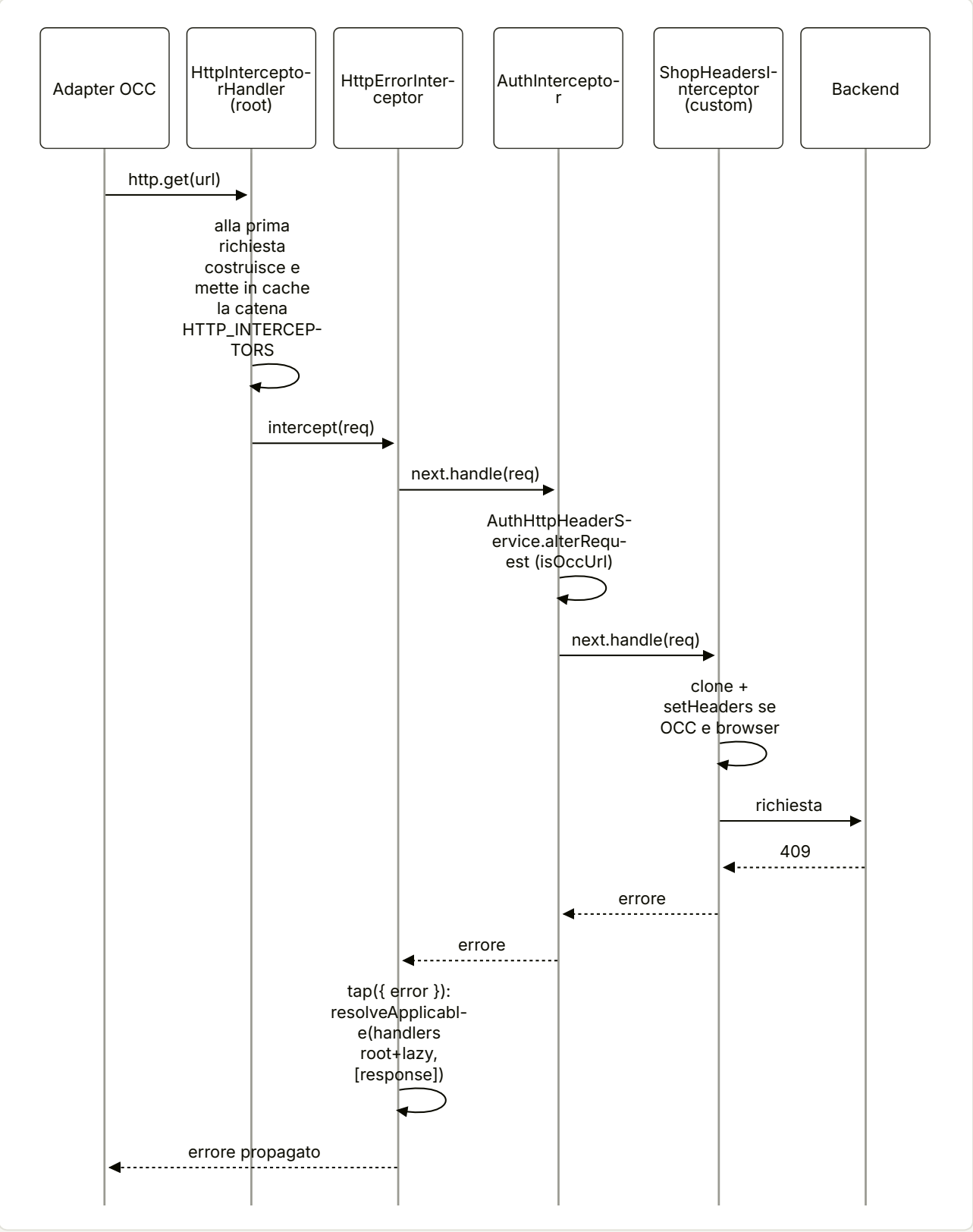
Riferimenti: core-libs/core/src/config/config-providers.ts:L18-L79, core-libs/core/src/config/config-tokens.ts:L10-L47, core-libs/core/src/config/services/configuration.service.ts:L57-L106, core-libs/core/src/config/config-initializer/config-initializer.service.ts:L58-L70.

4.2 Override di una facade lazy con wrapper module (Ricetta 6)



Riferimenti: feature-libs/order/root/facade/order-history.facade.ts:L19-L49, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L92, core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L85-L107.

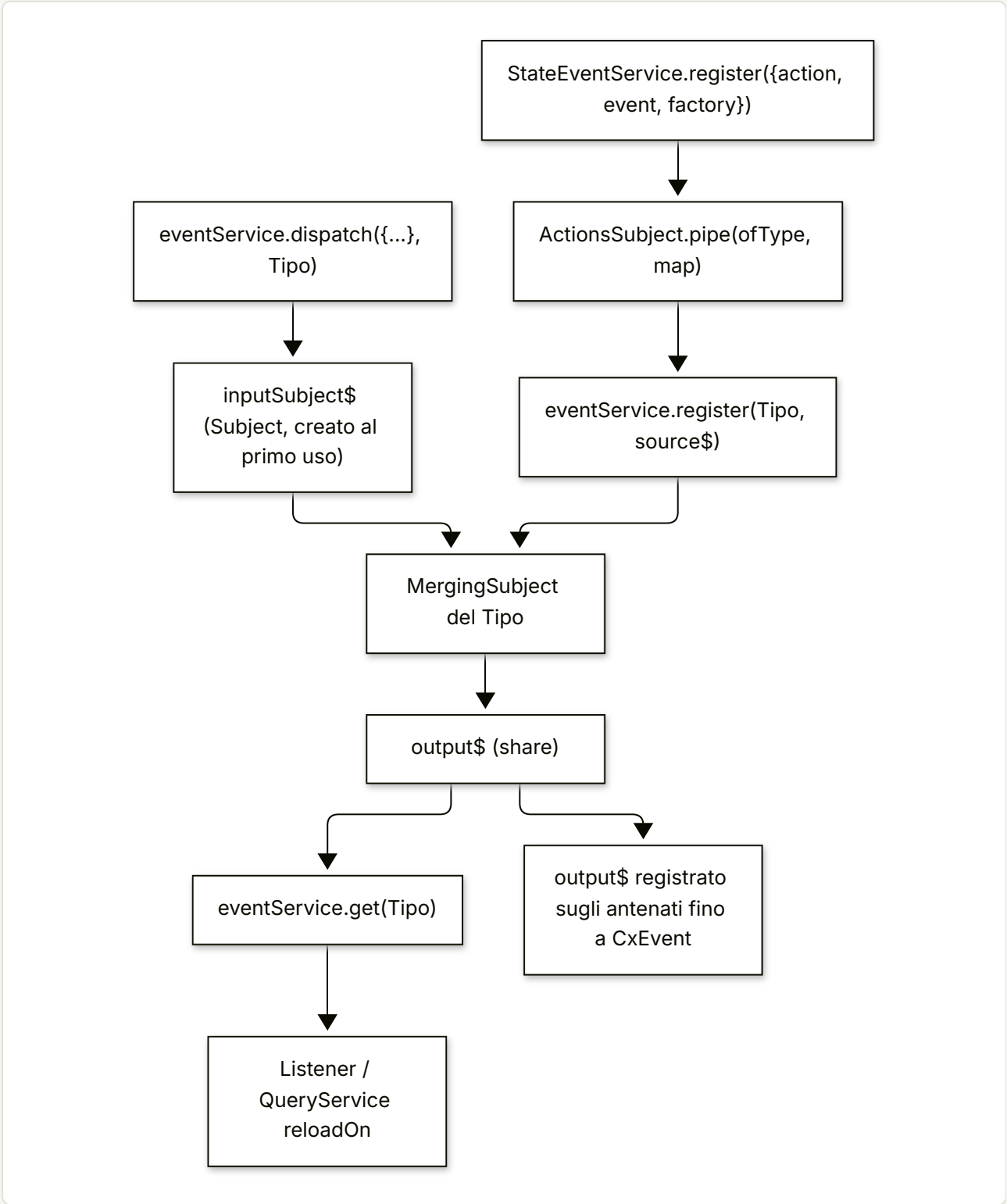
4.3 Richiesta HTTP: interceptor custom e handler di errore (Ricette 5, 7, 8)



L'ordine relativo degli interceptor dipende dall'ordine di registrazione dei multi provider: il primo registrato è il più esterno (`reduceRight`). Nella demo `HttpErrorInterceptor` (da `GlobalMessageModule.forRoot()` in `BaseCoreModule`) precede `AuthInterceptor` (da `AuthModule.forRoot()` in `SpartacusFeaturesModule`), e un interceptor nei `providers` dell'`AppModule` è elaborato per ultimo — `core-libs/core/src/global-message/global-`

message.module.ts:L27-L30, core-libs/core/src/base-core.module.ts:L35-L35, projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L13-L16, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L179-L179, angular:packages/common/http/src/interceptor.ts:L237-L240. Altri riferimenti: angular:packages/common/http/src/backend.ts:L101-L122, angular:packages/common/http/src/interceptor.ts:L227-L240, core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L188-L207, core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L30-L65.

4.4 Eventi custom: dispatch, sorgenti derivate, action (Ricette 9-11)



Riferimenti: core-libs/core/src/event/event.service.ts:L68-L159, core-libs/core/src/event/utils/merging-subject.ts:L45-L80, core-libs/core/src/state/event/state-event.service.ts:L36-L55, core-libs/core/src/util/command-query/query.service.ts:L144-L154.

---

5. Estratti di codice reale

Estratti da cui sono modellati gli esempi della sezione 3.

`provideConfig` **è solo un multi provider** — core-libs/core/src/config/config-providers.ts:L18-L27

```ts
export function provideConfig(
  config: Config = {},
  defaultConfig = false
): ValueProvider {
  return {
    provide: defaultConfig ? DefaultConfigChunk : ConfigChunk,
    useValue: config,
    multi: true,
  };
}
```

Default e root fusi separatamente, root per ultima — core-libs/core/src/config/config-tokens.ts:L10-L21

```ts
export function configFactory() {
  return deepMerge({}, inject(DefaultConfig), inject(RootConfig));
}

/**
 * Global Configuration, can be used to inject configuration to any part of the app
 */
@Injectable({
  providedIn: 'root',
  useFactory: configFactory,
})
export abstract class Config {}
```

Array sostituiti nel merge — core-libs/core/src/config/utils/deep-merge.ts:L31-L49

```ts
  for (const key in source) {
    if (isRestricted(key)) {
      continue;
    }

    if (source[key] instanceof Date) {
      target[key] = source[key];
    } else if (isObject(source[key])) {
```

```
      if (!isObject(target[key])) {
        target[key] = {};
      }
      deepMerge(target[key] as Record<string, unknown>, source[key]);
    } else {
      target[key] = source[key];
    }
  }

  return deepMerge(target, ...sources);
}
```

Config unificata e merge in place — core-libs/core/src/config/services/configuration.service.ts:L92-L106

```ts
  private emitUnifiedConfig(): void {
    const newConfig: Config = deepMerge(
      {},
      this.defaultConfig,
      this.ambientDefaultConfig,
      this.ambientConfig,
      this.rootConfig
    );
    (this.unifiedConfig$ as BehaviorSubject<Config>).next(newConfig);

    // compatibility mechanism, can be disabled with feature toggle
    if (!isFeatureEnabled(this.config, 'disableConfigUpdates')) {
      deepMerge(this.config as Record<string, unknown>, newConfig);
    }
  }
```

Override per ordine dei chunk nell'app demo — projects/storefrontapp/src/app/spartacus/features/cart/wish-list-v2-feature.module.ts:L12-L27

```ts
@NgModule({
  imports: [WishListFeatureModule],
  providers: [
    provideConfig({
      featureModules: {
        [CART_WISH_LIST_FEATURE]: {
          module: () =>
            import('@spartacus/cart/wish-list').then(
              (m) => m.WishListWithV2Module
            ),
        },
      },
    }),
```

```
  ],
})
export class WishListV2FeatureModule {}
```

Factory di layout che clona prima di modificare — core-libs/storefront/recipes/config/layout-config.ts:L190-L198

```ts
export function layoutConfigFactory(): LayoutConfig {
  const config: LayoutConfig = JSON.parse(JSON.stringify(layoutConfig));

  applyUnifiedHeaderSlots(config);

  applyWithoutPageFold(config);

  return config;
}
```

Config tipizzata con augmentation — feature-libs/cart/base/root/config/cart-config.ts:L12-L32

```ts
@Injectable({
  providedIn: 'root',
  useExisting: Config,
})
export abstract class CartConfig {
  cart?: {
// ...
declare module '@spartacus/core' {
  interface Config extends CartConfig {}
}
```

Override di servizi root eager (ASM) — feature-libs/asm/root/asm-root.module.ts:L65-L76

```ts
    {
      provide: AuthStorageService,
      useExisting: AsmAuthStorageService,
    },
    {
      provide: AuthService,
      useExisting: AsmAuthService,
    },
    {
      provide: AuthHttpHeaderService,
      useExisting: AsmAuthHttpHeaderService,
    },
```

Punto unico per "è una chiamata OCC?" — core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L179-L183, core-libs/core/src/auth/user-auth/services/auth-http-header.service.

ts:L205-L207

```ts
  public shouldAddAuthorizationHeader(request: HttpRequest<any>): boolean {
    const hasAuthorizationHeader = !!this.getAuthorizationHeader(request);
    const isOccUrl = this.isOccUrl(request.url);
    return !hasAuthorizationHeader && isOccUrl;
  }
// ...
  protected isOccUrl(url: string): boolean {
    return url.includes(this.occEndpoints.getBaseUrl());
  }
```

Wrapper module lazy con estensioni — projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27

```ts
const extensions: Type<any>[] = [];
if (environment.s4Service) {
  extensions.push(S4ServiceOrderModule);
}
if (environment.omf) {
  extensions.push(OmfOrderModule);
}
if (environment.opf) {
  extensions.push(OpfOrderModule);
}
@NgModule({
  imports: [OrderModule, ...extensions],
})
export class OrderWrapperModule {}
```

L'estensione OMF ridefinisce la facade nel modulo lazy — integration-libs/omf/order/omf-order.module.ts:L15-L26

```ts
@NgModule({
  providers: [
    OmfOrderHistoryService,
    provideDefaultConfig(defaultOmfConfig),
    { provide: OrderHistoryAdapter, useClass: OccOmfOrderHistoryAdapter },
    {
      provide: OrderHistoryFacade,
      useClass: OmfOrderHistoryService,
    },
  ],
})
export class OmfOrderModule {}
```

Il proxy prende l'implementazione dall'injector lazy — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L52

```ts
    if (!this.featureModules.isConfigured(feature)) {
      return throwError(
        () => new Error(`Feature ${feature} is not configured properly`)
      );
    }

    let facadeService$ = this.featureModules.resolveFeature(feature).pipe(
      map((moduleRef) => moduleRef.injector),
      map((injector) => injector.get(facadeClass))
    );
```

Registrazione di un interceptor — integration-libs/segment-refs/root/http-interceptors/index.ts:L11-L17

```ts
export const segmentRefsInterceptors: Provider[] = [
  {
    provide: HTTP_INTERCEPTORS,
    useExisting: OccSegmentRefsInterceptor,
    multi: true,
  },
];
```

Interceptor che filtra le chiamate OCC — integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L69-L86

```ts
  intercept(
    request: HttpRequest<any>,
    next: HttpHandler
  ): Observable<HttpEvent<any>> {
    if (
      this.winRef.isBrowser() &&
      this.requestHeader &&
      this.segmentRefs &&
      request.url.includes(this.occEndpoints.getBaseUrl())
    ) {
      request = request.clone({
        setHeaders: {
          [this.requestHeader]: this.segmentRefs,
        },
      });
    }
    return next.handle(request);
  }
```

Handler d'errore importato solo da `@spartacus/core` — feature-libs/cart/base/core/http-interceptors/handlers/bad-cart-request.handler.ts:L18-L38

```ts
@Injectable({
  providedIn: 'root',
})
export class BadCartRequestHandler extends HttpErrorHandler {
  responseStatus = HttpResponseStatus.BAD_REQUEST;

  getPriority(): Priority {
    return Priority.NORMAL;
  }

  hasMatch(errorResponse: HttpErrorResponse): boolean {
    return (
      super.hasMatch(errorResponse) &&
      this.getErrors(errorResponse).some(isCartError)
    );
  }

  handleError(request: HttpRequest<any>, response: HttpErrorResponse): void {
    this.handleCartNotFoundError(request, response);
    this.handleOtherCartErrors(request, response);
  }
```

Handler raccolti da root + lazy — core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L22-L28

```ts
@Injectable({ providedIn: 'root' })
export class HttpErrorInterceptor implements HttpInterceptor {
  constructor(protected unifiedInjector: UnifiedInjector) {}

  protected handlers$: Observable<HttpErrorHandler[]> = this.unifiedInjector
    .getMulti(HttpErrorHandler)
    .pipe(shareReplay(1));
```

`register` **e** `dispatch` — core-libs/core/src/event/event.service.ts:L68-L84, core-libs/core/src/event/event.service.ts:L105-L113

```ts
  register<T>(eventType: AbstractType<T>, source$: Observable<T>): () => void {
    const eventMeta = this.getEventMeta(eventType);
    if (eventMeta.mergingSubject.has(source$)) {
      if (isDevMode()) {
        this.logger.warn(
          `EventService: the event source`,
          source$,
          `has been already registered for the type`,
          eventType
        );
      }
    } else {
```

```
      eventMeta.mergingSubject.add(source$);
    }

    return () => eventMeta.mergingSubject.remove(source$);
  }
// ...
  dispatch<T extends object>(event: T, eventType?: Type<T>): void {
    if (!eventType) {
      eventType = event.constructor as Type<T>;
    } else if (!(event instanceof eventType)) {
      event = createFrom(eventType, event);
    }
    const inputSubject$ = this.getInputSubject(eventType);
    inputSubject$.next(event);
  }
```

Listener di eventi con `Subscription` chiusa in `ngOnDestroy` — feature-libs/subscription-billing/root/events/subscription-billing-event.listener.ts:L19-L45

```ts
@Injectable({
  providedIn: 'root',
})
export class SubscriptionBillingEventListener implements OnDestroy {
  protected subscriptions = new Subscription();
  protected eventService = inject(EventService);

  constructor() {
    this.onLanguageAndCurrencySetEvent();
  }

  protected onLanguageAndCurrencySetEvent(): void {
    this.subscriptions.add(
      merge(
        this.eventService.get(LanguageSetEvent),
        this.eventService.get(CurrencySetEvent)
      ).subscribe(() => {
        this.eventService.dispatch({}, GetSubscriptionByCodeReloadEvent);
        this.eventService.dispatch({}, GetSubscriptionListReloadEvent);
      })
    );
  }

  ngOnDestroy(): void {
    this.subscriptions.unsubscribe();
  }
}
```

Evento derivato da `NavigationEvent` — feature-libs/cart/base/core/event/cart-page-event.builder.ts:L17-L35

```ts
export class CartPageEventBuilder {
  constructor(protected eventService: EventService) {
    this.register();
  }

  protected register(): void {
    this.eventService.register(CartPageEvent, this.buildCartPageEvent());
  }

  protected buildCartPageEvent(): Observable<CartPageEvent> {
    return this.eventService.get(NavigationEvent).pipe(
      filter((navigationEvent) => navigationEvent.semanticRoute === 'cart'),
      map((navigationEvent) =>
        createFrom(CartPageEvent, {
          navigation: navigationEvent,
        })
      )
    );
  }
```

Action → evento con `factory` — feature-libs/cart/base/core/event/cart-event.builder.ts:L136-L145

```ts
    this.stateEventService.register({
      action: CartActions.DELETE_CART,
      event: DeleteCartEvent,
      factory: (action: CartActions.DeleteCart) =>
        createFrom(DeleteCartEvent, {
          ...action.payload,
          cartCode: action.payload.cartId,
        }),
    });
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
qualsiasi, via `provideConfig` / `provideConfigFactory`	`Config`	—	core-libs/core/src/config/config-providers.ts:L18-L47	core-libs/core/src/config/config-tokens.ts:L37-L39
qualsiasi, via `provideDefaultConfig` / `provideDefaultConfigFactory`	`Config`	—	core-libs/core/src/config/config-providers.ts:L55-L79	core-libs/core/src/config/config-tokens.ts:L23-L28

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`features.disableConfigUpdates`	`boolean` (deprecato)	assente → `false`	`core-libs/core/src/features-config/config/features-config.ts:L101-L116`	`core-libs/core/src/config/services/configuration.service.ts:L103-L103`
`featureModules[name].module`	`() => Promise<any>`	definito dall'app	`core-libs/core/src/cms/config/cms-config.ts:L128-L129`	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35`
`featureModules[alias]` (es. `orderCore`)	`string`	`ORDER_CORE_FEATURE → ORDER_FEATURE`	`feature-libs/order/root/order-root.module.ts:L68-L80`	`core-libs/core/src/lazy-loading/feature-modules.service.ts:L95-L100`
`layoutSlots`	`LayoutSlotConfig`	`layoutConfigFactory()` nell'app demo	`core-libs/storefront/layout/config/layout-config.ts:L90-L102`	(vedi 18)
`segmentRefs.httpHeaderName` (modello per la Ricetta 3)	`string`	`'Segmentrefs'`	`integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13`	`integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L53-L59`

Le chiavi `shopHeaders.*` delle Ricette 3, 4 e 7 sono inventate per l'esempio e non esistono nel repo.

---

7. Commenti e note notevoli dal sorgente

- "Should not be used directly, use `provideConfig` or import `ConfigModule.withConfig` instead." e "General rule is, that all config provided in libraries should be provided as default config." — `core-libs/core/src/config/config-tokens.ts:L49-L62`.
- "To provide default configuration in libraries provideDefaultConfig should be used instead." — `core-libs/core/src/config/config-providers.ts:L11-L17`.
- "Recommended way to get config for code that can run before app will finish initialization (APP_INITIALIZERS, selected service constructors)" — `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L46-L48`.
- `disableConfigUpdates`: "@deprecated" e "if any service would like to take into an account configuration from lazy loaded modules, it will have to use ConfigurationService.unifiedConfig$" — `core-libs/core/src/features-config/config/features-config.ts:L101-L116`.
- "Injects config dynamically from e2e tests. Should be imported after other config modules." — `projects/storefrontapp/src/app/private/private.providers.ts:L54-L54`.
- "Uses AuthHttpHeaderService for request manipulation and error handling. Interceptor only hooks into request send/received events." — `core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L18-L27`.
- "Overrides `AuthHttpHeaderService` to handle asm calls as well (not only OCC)" — `feature-libs/asm/root/services/asm-auth-http-header.service.ts:L26-L29`.
- "All methods should either return an Observable or void. Any return type that is not an Observable will be ignored." — `core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L20-L24`.
- JSDoc di `WishListV2Module`: override di `WishListFacade` e ordine `imports: [WishListModule, WishListV2Module]` — `feature-libs/cart/wish-list/wish-list-v2.module.ts:L20-L34`.

- "Prefer `withInterceptors` and functional interceptors instead, as support for DI-provided interceptors may be phased out in a later release." — `angular:packages/common/http/src/provider.ts:L170-L174`.
- "applicables with the same priority are sorted by order of providers, the applicable that was provided later wins" — `core-libs/core/src/util/applicable.ts:L47-L57`.
- "CAUTION: To avoid memory leaks, the returned teardown function should be called when the event source is no longer maintained by its creator" — `core-libs/core/src/event/event.service.ts:L57-L59`.
- Commento su `switchMap` invece di `withLatestFrom` per non abbonarsi agli stream del carrello prima dell'action — `feature-libs/cart/base/core/event/cart-event.builder.ts:L208-L220`.

8. Extension point

Riepilogo delle ricette, con il capitolo della Parte II che descrive il meccanismo e le ricette affini della Parte III.

EXTENSION POINT	RICETTA	MECCANISMO (PARTE II/I)	RICETTE AFFINI
`provideConfig`, `ConfigModule.withConfig`	3.1	11 — Configurazione	53 (config di feature e traduzioni)
`provideConfigFactory`, `layoutConfigFactory`	3.2	11, 18 — Layout	51 — CMS, outlet, layout
Config tipizzata (`useExisting: Config` + augmentation), `provideDefaultConfig`, `provideConfigValidator`	3.3	11	53
`ConfigInitializerService.getStable`, `ConfigurationService.unifiedConfig$`	3.4	11, 12 — Feature module lazy	—
Override di servizio root (`useExisting` su sottoclasse)	3.5	01 — DI, 22 — Auth	52 — OCC e dati (adapter/converter)
Override di facade lazy nel wrapper module	3.6	12, 20 — Facade	53
`HTTP_INTERCEPTORS` + `OccEndpointsService.getBaseUrl`	3.7	06 — HttpClient e interceptor, 21 — OCC	52
`HttpErrorHandler` + `Priority`	3.8	41 — Errori e global message	—
`CxEvent`, `EventService.dispatch/get`	3.9	20	—
`EventService.register` (eventi derivati)	3.10	20	—
`StateEventService.register`	3.11	20, 19 — State management	—
Scelta dei soli simboli pubblici	tutte	54 — API pubblica vs interna	54

9. Trappole, casi limite, comportamenti controintuitivi

1. **L'ordine degli import decide gli override.** Per i multi provider (config) l'ultimo chunk elaborato vince nel `deepMerge`; per i single provider (servizi, facade) l'ultimo record sostituisce i precedenti; gli `imports` di un NgModule sono elaborati prima dei suoi `providers` — `angular:packages/core/src/di/r3_injector.ts:L497-L521`, `angular:packages/core/src/di/provider_collection.ts:L293-L353`.

2. **Gli array di config si sostituiscono** — core-libs/core/src/config/utils/deep-merge.ts:L43-L45.
3. **Root batte lazy** nella config unificata, indipendentemente dall'ordine di caricamento — core-libs/core/src/config/services/configuration.service.ts:L92-L99.
4. **Il** `Config` **iniettato cambia dopo il bootstrap** (merge in place), salvo `disableConfigUpdates` — core-libs/core/src/config/services/configuration.service.ts:L102-L105.
5. **Fornire una facade lazy in root la rende eager** e disattiva il proxy — angular:packages/core/src/di/r3_injector.ts:L347-L366.
6. **Il proxy di facade espone solo i** `methods`**/**`properties` **del descrittore** — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L110-L131.
7. **Interceptor in moduli lazy ignorati**; `HttpErrorHandler` in moduli lazy invece funziona — angular:packages/common/http/src/backend.ts:L101-L122, core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L26-L28.
8. **Senza** `withInterceptorsFromDi()` **nessun interceptor class-based è attivo** — projects/storefrontapp/src/app/app.config.ts:L28-L28.
9. **Override concorrenti dello stesso servizio** (es. app + ASM + Punchout su `AuthHttpHeaderService`): vince l'ultimo provider elaborato, gli altri vengono silenziosamente ignorati — feature-libs/asm/root/asm-root.module.ts:L73-L76, integration-libs/punchout/root/punchout.root.module.ts:L49-L66.
10. **Eventi senza listener vanno persi**; i listener vanno istanziati presto (modulo che li inietta nel costruttore) — core-libs/core/src/event/utils/merging-subject.ts:L45-L80, feature-libs/subscription-billing/root/events/subscription-billing-event.module.ts:L8-L13.
11. **Eventi derivati da action di feature lazy esistono solo dopo il caricamento del chunk** — feature-libs/cart/base/core/event/cart-event.module.ts:L10-L15.
12. **Non tutto ciò che si vede nel repo è pubblico**: controllare sempre `public_api.ts` del pacchetto (es. `@spartacus/segment-refs/root`) — integration-libs/segment-refs/root/public_api.ts:L7-L7.

---

10. Come fare debug

- **Valore di config finale**: iniettare `Config` (o `ConfigurationService` e sottoscrivere `unifiedConfig$`) e loggare la chiave; confrontare prima e dopo il caricamento della feature lazy per vedere il merge in place — core-libs/core/src/config/services/configuration.service.ts:L92-L106.
- **Ordine dei chunk**: iniettare `ConfigChunk`/`DefaultConfigChunk` (token pubblici) in un componente di debug e ispezionare l'array: l'ordine è quello di elaborazione dei provider — core-libs/core/src/config/index.ts:L10-L16, core-libs/core/src/config/config-tokens.ts:L37-L39.
- **Validator**: in dev mode i warning di `provideConfigValidator` compaiono in console dopo `getStable()` — core-libs/core/src/config/config-validator/config-validator.module.ts:L23-L37.
- **Override di servizio/facade**: nel metodo sovrascritto mettere un breakpoint; se non scatta, verificare (a) che il provider sia nell'injector giusto (root per servizi eager, wrapper lazy per facade), (b) l'ordine degli import, (c) che il metodo sia nei `methods` del descrittore — feature-libs/order/root/facade/order-history.facade.ts:L19-L43.
- **Facade che fallisce subito**: il messaggio "Feature X is not configured properly" indica `featureModules[X].module` mancante (o alias non risolto) — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47.
- **Interceptor**: verificare nel tab Network che l'header compaia; se no, controllare `withInterceptorsFromDi()`, che il modulo sia eager e che il filtro `getBaseUrl()` corrisponda all'URL reale (base site incluso) — core-libs/core/src/occ/services/occ-endpoints.service.ts:L85-L101.

- **Eventi**: in dev mode `EventService` avvisa di sorgenti registrate due volte e di tipi che non estendono `CxEvent` — core-libs/core/src/event/event.service.ts:L70-L78, core-libs/core/src/event/event.service.ts:L166-L197. Per vedere se un evento viene emesso, sottoscrivere `eventService.get(CxEvent)` (riceve tutti i tipi, grazie alla registrazione sugli antenati) — core-libs/core/src/event/event.service.ts:L144-L159.

---

11. Domande di autoverifica

1. **Un `provideConfig` nell'`AppModule` può essere sovrascritto da un `provideDefaultConfig` di una libreria importata dopo?** No: `Config = deepMerge({}, DefaultConfig, RootConfig)`, i default sono sempre sotto la root — core-libs/core/src/config/config-tokens.ts:L10-L12.
2. **Perché un array fornito con `provideConfig` "cancella" i valori di default?** Perché `deepMerge` ricorre solo sugli oggetti plain e assegna tutto il resto, array inclusi — core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/config/utils/deep-merge.ts:L36-L45.
3. **Un servizio che legge la config nel costruttore vede i default forniti da un modulo lazy?** Solo se il modulo è già caricato; in seguito il `Config` viene aggiornato in place (salvo `disableConfigUpdates`), ma un valore copiato resta vecchio: usare `unifiedConfig$` — core-libs/core/src/config/services/configuration.service.ts:L92-L106, core-libs/core/src/features-config/config/features-config.ts:L109-L113.
4. **Perché per sostituire `OrderHistoryFacade` non basta fornirla nell'`AppModule`?** Perché il record esplicito in root disattiva `useFactory: orderHistoryFacadeFactory` e quindi il proxy lazy; l'implementazione va fornita nell'injector del modulo lazy, da cui il proxy la legge con `moduleRef.injector.get` — angular:packages/core/src/di/r3_injector.ts:L347-L366, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52.
5. **In un wrapper module, perché i `providers` vincono su quelli di `OrderModule` importato?** Perché Angular elabora prima i provider degli `imports` e poi quelli del modulo, e un single provider sostituisce il record — angular:packages/core/src/di/provider_collection.ts:L345-L353, angular:packages/core/src/di/r3_injector.ts:L513-L521.
6. **Una sottoclasse `@Injectable` di `AuthHttpHeaderService` senza costruttore riceve le dipendenze?** Sì, Angular usa la factory ereditata (`ɵɵgetInheritedFactory`) — angular:packages/core/src/render3/di.ts:L918-L945; lo stesso accade per `InternalServerErrorHandler` — core-libs/core/src/global-message/http-interceptors/handlers/internal-server/internal-server.handler.ts:L13-L29.
7. **Perché un interceptor fornito in un modulo lazy non viene eseguito, mentre un `HttpErrorHandler` sì?** La catena degli interceptor è costruita dall'injector root e messa in cache alla prima richiesta; gli `HttpErrorHandler` sono raccolti con `UnifiedInjector.getMulti`, che include gli injector lazy — angular:packages/common/http/src/backend.ts:L101-L122, core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L26-L28.
8. **Come fa un handler custom per il 409 a prevalere su `ConflictHandler`?** `ConflictHandler` ha priorità `LOW` e `resolveApplicable` sceglie la più alta, ma le feature lazy di organization registrano handler 409 `NORMAL` e a parità vince l'ultimo registrato (quello lazy): per prevalere in modo affidabile usare `Priority.HIGH` — core-libs/core/src/global-message/http-interceptors/handlers/conflict/conflict.handler.ts:L26-L28, core-libs/core/src/util/applicable.ts:L76-L86, feature-libs/organization/user-registration/core/http-interceptors/conflict/conflict.handler.ts:L20-L39, core-libs/core/src/lazy-loading/unified-injector.ts:L82-L92.
9. **Perché un evento custom con un costruttore `constructor(public id: string)` dà problemi con `dispatch({id}, Tipo)`?** `createFrom` fa `new type()` senza argomenti e poi `Object.assign` — core-libs/

core/src/util/create-from.ts:L15-L17.

10. **Cosa succede se si fa `dispatch` di un evento prima che il listener sia sottoscritto?** L'evento va perso: il `MergingSubject` inoltra i valori solo ai consumer presenti — core-libs/core/src/event/utils/merging-subject.ts:L45-L80, core-libs/core/src/event/event.service.ts:L120-L128.

51

Ricette: componenti CMS, outlet, route e layout

Sostituire o creare componenti CMS, `provideOutlet`, route e layout.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

Capitolo di **ricette** (Parte III). Ogni ricetta in sezione 3 è un extension point pubblico con esempio completo, meccanismo sorgente e trappole. Le spiegazioni di fondo sono nei capitoli della Parte II: 14 — Routing, 15 — CMS core, 16 — CMS rendering, 17 — Outlet, 18 — Layout, breakpoint e a11y, 40 — SEO e PWA (meta resolver), 45 — Componenti storefront (i componenti da estendere). Le ricette sulla configurazione e la DI in generale sono in 50 — Ricette: configurazione, DI, interceptor, eventi; quelle su OCC e meta resolver in 52 — Ricette: OCC, adapter, converter, endpoint, meta resolver.

1. A cosa serve e perché esiste

Uno storefront Spartacus è disegnato dal backend CMS: la pagina arriva come *template* + *slot* + *componenti tipizzati*, e lo storefront decide **quale componente Angular** disegnare per ogni tipo, **dove** mettere ogni slot e **quale route** porta a quale pagina. Ognuna di queste decisioni è guidata da configurazione o da registri globali, ed è quindi personalizzabile senza toccare il codice delle librerie:

BISOGNO	EXTENSION POINT	RICETTA
Disegnare diversamente un tipo CMS esistente	`cmsComponents[type].component` (mapping)	3.1
Riutilizzare la logica di un componente di libreria cambiandone la vista	ereditarietà di classe + `CmsComponentData`	3.2
Supportare un nuovo tipo CMS definito nel backend	nuovo componente + mapping (`data`, `deferLoading`, `guards`, `providers`)	3.3
Aggiungere UI in un punto nominato senza toccare i template	`provideOutlet` (`OutletPosition`)	3.4
Avvolgere/sostituire slot, componenti o pagine dal template	`[cxOutletRef]` + `cxOutletPos`	3.5
Dare a una content page CMS un URL "parlante"	`CmsRoute` + `RoutingConfig` (`cxRoute`, `pageLabel`)	3.6
Disporre gli slot di un template nuovo (ed eventualmente una pagina statica)	`LayoutConfig.layoutSlots` (+ `CmsStructureConfig`)	3.7

Tutte le ricette poggiano sullo stesso meccanismo di configurazione: `provideConfig` registra un chunk su `ConfigChunk`, `provideDefaultConfig` su `DefaultConfigChunk` (core-libs/core/src/config/config-providers.ts:L18-L27, core-libs/core/src/config/config-providers.ts:L55-L61), e la config finale è `deepMerge({}, DefaultConfig, RootConfig)`, quindi i chunk applicativi vincono sempre sui default di libreria (core-libs/core/src/config/config-tokens.ts:L10-L12). Il JSDoc lo dice esplicitamente: "To provide default configuration in libraries provideDefaultConfig should be used instead" (core-libs/core/src/config/config-providers.ts:L14-L14). Da qui la regola pratica di questo capitolo: **nell'app si usa** `provideConfig`, in una libreria riusabile `provideDefaultConfig`.

2. Mappa dei file

File sorgente che rendono possibili le ricette (tutti sotto il repo alla SHA indicata).

PATH	RUOLO
core-libs/core/public_api.ts	Barrel pubblico di @spartacus/core (cms, config, model, routing alle righe 14, 15, 26, 30)
core-libs/storefront/public_api.ts	Barrel pubblico di @spartacus/storefront (cms-components, cms-structure, layout, recipes alle righe 21, 23, 26, 27)
core-libs/core/src/config/config-providers.ts	provideConfig, provideDefaultConfig, varianti factory
core-libs/core/src/config/config-tokens.ts	Config, ConfigChunk, DefaultConfigChunk, merge default + root
core-libs/core/src/config/utils/deep-merge.ts	Merge ricorsivo: oggetti fusi, array/classi sostituiti
core-libs/core/src/cms/config/cms-config.ts	CmsConfig, CmsComponentMapping, DeferLoadingStrategy, FeatureModuleConfig
core-libs/core/src/cms/config/cms-structure.config.ts	CmsStructureConfig, CmsPageConfig, CmsPageSlotConfig (pagine statiche)
core-libs/core/src/cms/services/cms-structure-config.service.ts	Fusione struttura statica / backend
core-libs/core/src/cms/connectors/page/cms-page.connector.ts	ignoreBackend e merge della struttura statica
core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts	Calcolo del flexType (chiave di mapping)
core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts	pageLabelOrId per le content page
core-libs/core/src/routing/configurable-routes/config/routing-config.ts	RoutingConfig, RouteLoadStrategy
core-libs/core/src/routing/configurable-routes/routes-config.ts	RoutesConfig, RouteConfig
core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts	Applica paths/matchers alle route con data.cxRoute
core-libs/core/src/routing/models/cms-route.ts	CmsRoute, CmsRouteData (pageLabel, cxCmsRouteContext)
core-libs/core/src/routing/store/reducers/router.reducer.ts	CustomSerializer: PageContext, cmsRequired
core-libs/storefront/cms-structure/page/model/cms-component-data.ts	Token CmsComponentData
core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts	Istanzia il componente mappato
core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts	Injector per componente (CmsComponentData + providers)
core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts	data$ = dati statici + dati CMS
core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts	Handler per classi componente
core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts	Handler per () => import(...)
core-libs/storefront/cms-structure/services/cms-components.service.ts	Risoluzione mapping, guard, childRoutes, i18n, disableSSR

PATH	RUOLO
core-libs/storefront/cms-structure/guards/cms-page.guard.ts	CmsPageGuard
core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts	canActivatePage / not-found
core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts	Route '**' CMS aggiunta in coda
core-libs/storefront/cms-structure/routing/default-routing-config.ts	Route semantiche di default
core-libs/storefront/cms-structure/outlet/outlet.providers.ts	provideOutlet, ProvideOutletOptions
core-libs/storefront/cms-structure/outlet/outlet.module.ts	OutletModule.forRoot()/forChild(), registerOutletsFactory
core-libs/storefront/cms-structure/outlet/outlet.service.ts	Registro degli outlet
core-libs/storefront/cms-structure/outlet/outlet.directive.ts	[cxOutlet]: build BEFORE/REPLACE/AFTER, OutletContextData
core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts	[cxOutletRef]
core-libs/storefront/cms-structure/outlet/outlet.model.ts	OutletPosition, OutletContextData
core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html	Outlet di template/section + slot
core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts	Risoluzione degli slot da layoutSlots
core-libs/storefront/cms-structure/page/slot/page-slot.component.html	Outlet di slot e di componente
core-libs/storefront/cms-structure/utils/cms-structure.util.ts	provideCmsStructure
core-libs/storefront/layout/config/layout-config.ts	LayoutConfig, SlotConfig, SlotGroup, LayoutSlotConfig
core-libs/storefront/recipes/config/layout-config.ts	layoutConfigFactory (layout di riferimento)
core-libs/storefront/cms-components/product/product-outlets.model.ts	ProductDetailOutlets, ProductListOutlets

3. Come funziona passo per passo

Prima di ogni ricetta, il percorso comune "dal mapping al componente" (dettagli nel cap. 16):

1. PageSlotComponent crea, per ogni componente dello slot, un outlet con chiave component.flexType e dentro un cxComponentWrapper (core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L6-L16).

2. Il `flexType` è calcolato dal normalizer OCC: `CMSFlexComponent` → `component.flexType`, `JspIncludeComponent` → `component.uid`, altrimenti `typeCode` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L151-L158).
3. `ComponentWrapperDirective` risolve il mapping, verifica `shouldRender` e lancia il componente (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L67-L102), usando un injector creato da `CmsInjectorService.getInjector` che fornisce `CmsComponentData` e i `providers` del mapping (core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50).
4. Il componente è creato da un handler scelto con `resolveApplicable` (core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L49-L65): `DefaultComponentHandler` per le classi, `LazyComponentHandler` per `() => import(...)` (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L27-L34, core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L38-L52).

3.1 R1 — Sostituire un componente CMS (mapping `cmsComponents`)

Quando usarlo

Quando il backend continua a inviare un tipo CMS standard (es. `CMSParagraphComponent`) ma lo vuoi disegnare con un tuo componente, completamente nuovo, senza ereditare nulla dall'originale. È il modo più "pulito": il tuo componente dipende solo dal modello dati (`CmsParagraphComponent`) e dal token `CmsComponentData`, non dall'implementazione di libreria.

Esempio

> Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// custom-paragraph.module.ts — esempio illustrativo, non presente nel repo
import { AsyncPipe, NgIf } from '@angular/common';
import { ChangeDetectionStrategy, Component, NgModule } from '@angular/core';
import {
  CmsConfig,
  CmsParagraphComponent,
  provideConfig,
} from '@spartacus/core';
import { CmsComponentData } from '@spartacus/storefront';

@Component({
  selector: 'app-plain-paragraph',
  template: `
    <article *ngIf="component.data$ | async as data" class="app-paragraph">
      <h2 *ngIf="data.title">{{ data.title }}</h2>
      <div [innerHTML]="data.content"></div>
    </article>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush,
  imports: [NgIf, AsyncPipe],
})
export class PlainParagraphComponent {
  constructor(public component: CmsComponentData<CmsParagraphComponent>) {}
}
```

```
@NgModule({
  imports: [PlainParagraphComponent],
  providers: [
    // provideConfig (ConfigChunk) vince sempre sul provideDefaultConfig della libreria
    provideConfig(<CmsConfig>{
      cmsComponents: {
        CMSParagraphComponent: { component: PlainParagraphComponent },
      },
    }),
  ],
})
export class CustomParagraphModule {}
```

Il modulo va importato nell'app (nella demo la config applicativa vive in un `NgModule` importato da `AppModule`, a sua volta incluso con `importProvidersFrom(AppModule)` in `appConfig`: projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L44-L72, projects/storefrontapp/src/app/app.module.ts:L15-L25, projects/storefrontapp/src/app/app.config.ts:L26-L35).

Catene di export dei simboli `@spartacus/*`:

SIMBOLO	CATENA
provideConfig	core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L27
CmsConfig	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L8-L8 → core-libs/core/src/cms/config/index.ts:L7-L7 → core-libs/core/src/cms/config/cms-config.ts:L124-L134
CmsParagraphComponent	core-libs/core/public_api.ts:L26-L26 → core-libs/core/src/model/index.ts:L8-L8 → core-libs/core/src/model/cms.model.ts:L88-L92
CmsComponentData	core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L10-L10 → core-libs/storefront/cms-structure/page/index.ts:L8-L8 → core-libs/storefront/cms-structure/page/model/index.ts:L7-L7 → core-libs/storefront/cms-structure/page/model/cms-component-data.ts:L10-L13

Perché funziona (file sorgente)

- `CmsConfig` è risolto con `useExisting: Config`, quindi il chunk entra nella config globale (core-libs/core/src/cms/config/cms-config.ts:L124-L134); `CMSComponentConfig` è un dizionario `componentType → CmsComponentMapping` (core-libs/core/src/cms/config/cms-config.ts:L103-L107).
- Il mapping di libreria è un default: `CmsParagraphModule` usa `provideDefaultConfig` (core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L21-L32), e la config finale mette `RootConfig` sopra `DefaultConfig` (core-libs/core/src/config/config-tokens.ts:L10-L12).
- `deepMerge` sostituisce funzioni/classi per assegnazione (core-libs/core/src/config/utils/deep-merge.ts:L36-L45): la classe `component` dell'app rimpiazza quella di libreria.
- `CmsComponentsService.getMapping` legge `mappings[type]` o lo snapshot statico/`config.cmsComponents` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L165-L181); il wrapper crea il componente con quell'injector (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L81-L102).
- Modello reale: image-zoom rimappa `ProductImagesComponent` sul proprio componente (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom.module.ts:L53-L59); lo stesso

componente può servire più tipi (core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L21-L32).

Limiti e trappole

- `provideDefaultConfig` **nell'app è fragile**: sia il tuo chunk sia quello di libreria finirebbero in `DefaultConfigChunk`, dove vince l'ordine di iniezione (core-libs/core/src/config/config-tokens.ts:L23-L28); `provideConfig` evita il problema.
- **Il mapping è fuso, non sostituito**: cambiando solo `component` erediti `guards`, `childRoutes`, `providers`, `data` dell'originale; se invece ridefinisci un array (`guards`, `providers`) lo **sostituisci per intero** (core-libs/core/src/config/utils/deep-merge.ts:L36-L45, core-libs/core/src/cms/config/cms-config.ts:L60-L93).
- **Chiave =** `flexType`: per i `JspIncludeComponent` è l'`uid`, per i `CMSFlexComponent` il `flexType` (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L151-L158).
- **Mapping mancante = silenzio in produzione**: solo un warning dev, una volta per tipo (core-libs/storefront/cms-structure/services/cms-components.service.ts:L170-L178), e il wrapper esce senza renderizzare (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L86-L88).
- **Tipi coperti da feature lazy**: il mapping statico prevale su quello della feature (`deepMerge({}, featureComponentMapping, staticConfig)`, core-libs/storefront/cms-structure/services/cms-components.service.ts:L117-L141), ma la feature viene comunque caricata e il componente è creato con il suo `NgModuleRef` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L149-L153, core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L99-L101). `DA VERIFICARE` a runtime il caso image-zoom, dove `ProductImagesModule` mappa staticamente lo stesso tipo della feature lazy: da codice prevarrebbe il mapping statico (dossier T4/Q1; non eseguibile senza `node_modules`).
- **Snapshot statico**: `cmsComponents` aggiunti alla config globale dopo la sua stabilizzazione (es. da moduli lazy) non sono visti da `getMapping` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L60-L69).
- Il binding `[innerHTML]` è sanificato da Angular; l'originale invece usa `bypassSecurityTrustHtml` (core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L49-L51): stili/attributi inline del CMS possono quindi differire.

3.2 R2 — Estendere un componente CMS esistente

Quando usarlo

Quando la logica del componente di libreria va bene (servizi iniettati, stream, handler) e vuoi solo una vista diversa o l'override di qualche metodo `protected`/`public`. È lo schema usato dalla libreria stessa: `MyAccountV2UserComponent extends LoginComponent {}`, `BreadcrumbComponent extends PageTitleComponent`, `ProductImageZoomProductImagesComponent extends ProductImagesComponent`.

Esempio

> Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// boxed-paragraph.module.ts — esempio illustrativo, non presente nel repo
import { AsyncPipe, NgIf } from '@angular/common';
import { ChangeDetectionStrategy, Component, NgModule } from '@angular/core';
import { CmsConfig, provideConfig } from '@spartacus/core';
import { ParagraphComponent } from '@spartacus/storefront';
```

```
@Component({
  selector: 'app-boxed-paragraph',
  template: `
    <section *ngIf="component.data$ | async as data" class="boxed">
      <h3 *ngIf="data.title">{{ data.title }}</h3>
      <div [innerHTML]="bypassSecurityTrustHtml(data.content)"></div>
    </section>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush,
  imports: [NgIf, AsyncPipe],
})
// Nessun costruttore: Angular usa la factory ereditata da ParagraphComponent
// (CmsComponentData + Router) e la sottoclasse eredita sanitizer,
// bypassSecurityTrustHtml() e il listener click per i link interni.
export class BoxedParagraphComponent extends ParagraphComponent {}

@NgModule({
  imports: [BoxedParagraphComponent],
  providers: [
    provideConfig(<CmsConfig>{
      cmsComponents: {
        CMSTabParagraphComponent: { component: BoxedParagraphComponent },
      },
    }),
  ],
})
export class BoxedParagraphModule {}
```

Catene di export:

SIMBOLO	CATENA
`provideConfig`, `CmsConfig`	come in 3.1
`ParagraphComponent`	`core-libs/storefront/public_api.ts:L21-L21` → `core-libs/storefront/cms-components/index.ts:L8-L8` → `core-libs/storefront/cms-components/content/index.ts:L13-L13` → `core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L26-L52`

Variante con costruttore (quando servono servizi aggiuntivi): reiniettare **tutti** i parametri della base e chiamare `super(...)`, come `BreadcrumbComponent` (estratto in 5.3).

Perché funziona (file sorgente)

- `CmsComponentData` è una classe astratta usata come token DI (`core-libs/storefront/cms-structure/page/model/cms-component-data.ts:L10-L13`), fornita dall'injector del wrapper indipendentemente da quale classe venga creata (`core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L35-L49`).
- Senza costruttore proprio, Angular risale la catena dei prototipi per trovare la factory della superclasse (`angular:packages/core/src/render3/di.ts:L918-L935`): la sottoclasse riceve

`CmsComponentData<CmsParagraphComponent>` e `Router` come la base (core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L44-L47).

- I membri usati dal template sono accessibili: `sanitizer` è `protected`, `bypassSecurityTrustHtml` è `public` (core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L26-L52).
- Le host binding della superclasse vengono fuse in quelle della sottoclasse da `ɵɵInheritDefinitionFeature` (angular:packages/core/src/render3/features/inherit_definition_feature.ts:L84-L86, angular:packages/core/src/render3/features/inherit_definition_feature.ts:L215-L227), e il compilatore mette i listener host nella funzione `hostBindings` (angular:packages/compiler/src/render3/view/compiler.ts:L482-L522): il `@HostListener('click')` della base resta attivo.
- Modelli reali: `MyAccountV2UserComponent extends LoginComponent {}` con nuovo template e nuovi `imports` (feature-libs/user/account/components/my-account-v2-user/my-account-v2-user.component.ts:L13-L18); `ProductImageZoomProductImagesComponent` riusa `mainMediaContainer/product$` `protected` della base (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-product-images/product-image-zoom-product-images.component.ts:L36-L57, core-libs/storefront/cms-components/product/product-images/product-images.component.ts:L33-L55).

Limiti e trappole

- **I metadati `@Component` vanno ridichiarati**: selector, template e `imports` standalone (feature-libs/user/account/components/my-account-v2-user/my-account-v2-user.component.ts:L13-L17). Il template originale usa `SupplementHashAnchorsPipe` (core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L24-L24); l'esempio non la usa, quindi l'aggiunta del prefisso agli anchor `#` non avviene.
- **Membri `private` non estendibili**: es. `PageTitleComponent.setTitle` (core-libs/storefront/cms-components/navigation/page-header/page-title.component.ts:L42-L44).
- **Costruttore = API fragile**: se ne dichiari uno, devi ripetere tutti i parametri della base e chiamare `super` (core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.component.ts:L44-L50); un cambio della firma della base in un upgrade rompe la sottoclasse (Spartacus gestisce questi casi con le migrazioni "constructor deprecations", vedi cap. 43).
- **Stesso selector della base**: image-zoom riusa `cx-product-images` per ereditare gli stili (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-product-images/product-image-zoom-product-images.component.ts:L22-L22), ma la sostituzione avviene solo via mapping CMS, non dove la base è usata direttamente in un template.
- **Override di `CmsComponentData` nei `providers` del mapping**: i `providers` sono aggiunti dopo `CmsComponentData` nello stesso `Injector.create` (core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L36-L47), quindi un provider per lo stesso token lo sovrascrive (angular:packages/core/src/di/r3_injector.ts:L497-L521).
- Le stesse trappole di merge di R1 valgono anche qui, perché la registrazione passa ancora da `cmsComponents`.

3.3 R3 — Creare un nuovo componente CMS

Quando usarlo

Quando nel backend è stato definito un **nuovo tipo** di componente CMS (o un `CMSFlexComponent` con un nuovo `flexType`) e lo storefront deve disegnarlo. Servono: un'interfaccia dati che estende `CmsComponent`, un componente che inietta `CmsComponentData<T>`, un mapping con eventuali dati statici, strategia di defer, guard o provider.

Esempio

> Esempio illustrativo, non presente nel repo. *PromoBannerComponent* è un tipo CMS ipotetico definito nel backend.

TYPESCRIPT

```typescript
// promo-banner.module.ts — esempio illustrativo, non presente nel repo
import { AsyncPipe, NgIf } from '@angular/common';
import { ChangeDetectionStrategy, Component, NgModule } from '@angular/core';
import {
  CmsComponent,
  CmsComponentMapping,
  CmsConfig,
  DeferLoadingStrategy,
  provideConfig,
} from '@spartacus/core';
import { CmsComponentData } from '@spartacus/storefront';

/** Attributi del tipo CMS custom, come arrivano dal backend. */
export interface CmsPromoBannerComponent extends CmsComponent {
  headline?: string;
  ctaLabel?: string;
  ctaUrl?: string;
}

@Component({
  selector: 'app-promo-banner',
  template: `
    <aside *ngIf="component.data$ | async as data" class="promo">
      <!-- data$ emette prima i dati statici, poi quelli del CMS -->
      <strong>{{ data.headline }}</strong>
      <a *ngIf="data.ctaUrl" [href]="data.ctaUrl">{{ data.ctaLabel }}</a>
    </aside>
  `,
  changeDetection: ChangeDetectionStrategy.OnPush,
  imports: [NgIf, AsyncPipe],
})
export class PromoBannerComponent {
  constructor(public component: CmsComponentData<CmsPromoBannerComponent>) {}
}

const promoBannerMapping: CmsComponentMapping<CmsPromoBannerComponent> = {
  component: PromoBannerComponent,
  // dati statici: valori iniziali e proprietà assenti nell'API
  data: { ctaLabel: 'Scopri di più' },
  // il banner è sotto la piega: rendering differito (solo client-side, slot non prerenderizzati)
  deferLoading: DeferLoadingStrategy.DEFER,
};

@NgModule({
  imports: [PromoBannerComponent],
```

```
  providers: [
    provideConfig(<CmsConfig>{
      cmsComponents: {
        // chiave = flexType (qui uguale al typeCode del backend)
        PromoBannerComponent: promoBannerMapping,
      },
    }),
  ],
})
export class PromoBannerModule {}
```

Catene di export:

SIMBOLO	CATENA
CmsComponent	core-libs/core/public_api.ts:L26-L26 → core-libs/core/src/model/index.ts:L8-L8 → core-libs/core/src/model/cms.model.ts:L7-L13
CmsComponentMapping, DeferLoadingStrategy	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L8-L8 → core-libs/core/src/cms/config/index.ts:L7-L7 → core-libs/core/src/cms/config/cms-config.ts:L60-L93, core-libs/core/src/cms/config/cms-config.ts:L95-L101
CmsConfig, provideConfig, CmsComponentData	come in 3.1

Perché funziona (file sorgente)

- CmsComponentMapping<T> è generico sul tipo dei dati e accetta component, providers, childRoutes, disableSSR, i18nKeys, guards, data, deferLoading (core-libs/core/src/cms/config/cms-config.ts:L60-L93); il JSDoc di data elenca gli usi: dati iniziali, proprietà UI assenti nell'API, ghost design (core-libs/core/src/cms/config/cms-config.ts:L79-L85).
- ComponentDataProvider.get fonde {...static, ...cms} ed emette prima i dati statici con startWith (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L35-L59); la sorgente CMS è CmsService.getComponentData (core-libs/core/src/cms/facade/cms.service.ts:L74-L74).
- getDeferLoadingStrategy legge deferLoading dallo snapshot statico (core-libs/storefront/cms-structure/services/cms-components.service.ts:L195-L200) e lo slot lo passa come cxOutletDefer all'outlet del componente (core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L7-L13) non così com'è ma tramite PageSlotService.getComponentDeferOptions, che lo ignora (INSTANT) per gli slot prerenderizzati visibili (core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L63-L73); in SSR DeferLoaderService renderizza comunque subito (core-libs/storefront/layout/loading/defer-loader.service.ts:L55-L63). Quindi DeferLoadingStrategy.DEFER ha effetto solo nella navigazione client-side; il deferred loading è spiegato nel cap. 18.
- Canone di libreria: ParagraphComponent standalone, OnPush, public component: CmsComponentData<...> (core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L20-L47), template su component.data$ | async (core-libs/storefront/cms-components/content/paragraph/paragraph.component.html:L1-L4), mapping in provideDefaultConfig (core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L14-L35).
- Opzioni avanzate del mapping, tutte raccolte da CmsPageGuardService.canActivatePage prima dell'attivazione (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L99): guards uniti in un Set per tutta la pagina (core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261), i18nKeys per precaricare chunk di traduzione (core-libs/storefront/cms-structure/services/cms-components.service.ts:L266-L276), childRoutes (core-libs/storefront/cms-structure/

services/cms-components.service.ts:L205-L248). Esempio con `providers` + `guards` + `childRoutes`: feature-libs/organization/administration/components/budget/budget.config.ts:L29-L80.

- Componente senza backend: `provideCmsStructure({ componentId, pageSlotPosition })` crea `cmsStructure.components[componentId] = { typeCode, flexType }` e lo slot globale (core-libs/storefront/cms-structure/utils/cms-structure.util.ts:L31-L66); uso reale in `defaultCmsContentProviders` (core-libs/storefront/recipes/config/static-cms-structure.ts:L10-L19).

Limiti e trappole

- **Dati parziali**: il primo valore di `data$` è il solo dato statico (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L44-L51); nell'esempio `headline` è inizialmente `undefined`.
- **Componente senza uid**: `data$` è `of(static)` o `EMPTY` (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L55-L57); con `EMPTY` un `*ngIf="data$ | async"` non disegna nulla. Il ramo vale solo per componenti senza `uid`: i componenti di `provideCmsStructure`/`cmsStructure.slots` ricevono `uid = componentId` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L180-L180), sono aggiunti allo slot e a `pageStructure.components` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L153-L161) e i loro dati (`typeCode`, `flexType`, più eventuali attributi in `cmsStructure.components`) arrivano da `getComponentData(uid)` (core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L43-L54).
- `disableSSR: true` toglie il componente anche da `childRoutes` e `i18nKeys` lato server (core-libs/storefront/cms-structure/services/cms-components.service.ts:L187-L190, core-libs/storefront/cms-structure/services/cms-components.service.ts:L208-L210, core-libs/storefront/cms-structure/services/cms-components.service.ts:L269-L270).
- `childRoutes` funzionano solo su content page con label che inizia con `/` e causano `resetConfig` + nuova navigazione (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L82-L88, core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L98-L123).
- **Componente lazy** (`component: () => import(...).then(m => m.X)`): riconosciuto da `LazyComponentHandler` solo se `toString()` della funzione inizia con `function()` o `()=>` (core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L45-L48); altrimenti cade sul `DefaultComponentHandler` che tratta ogni funzione come classe (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L28-L30). DA VERIFICARE con l'output del builder esbuild di Angular 21 (dossier Q3); nel codice non-spec del repo non c'è un esempio reale di questa forma (dossier Q5).
- **Web component** (`component: 'url#tag'`): `WebComponentHandler` esiste ma non è registrato né esportato (core-libs/storefront/cms-structure/page/component/page-component.module.ts:L23-L40, core-libs/storefront/cms-structure/page/component/handlers/index.ts:L7-L9).
- `DefaultComponentHandler` usa ancora `ComponentFactoryResolver` (core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L56-L70), deprecato in Angular (angular:packages/core/src/linker/component_factory_resolver.ts:L38-L39): funziona anche con componenti standalone, ma è un punto da tenere d'occhio negli upgrade.

3.4 R4 — Inserire un componente in un outlet con `provideOutlet`

Quando usarlo

Quando vuoi aggiungere (o sostituire) UI in un punto nominato di un template di libreria che **non** è un componente CMS: prezzo e sommario del PDP, dettagli di una riga carrello, riepilogo ordine. Il punto deve esporre

un `[cxOutlet]`; i nomi sono negli enum `*Outlets` (es. `ProductDetailOutlets`, `CartOutlets`) e nell'Appendice A08.

Esempio

> Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// price-note.module.ts — esempio illustrativo, non presente nel repo
import { AsyncPipe, NgIf } from '@angular/common';
import { Component, NgModule, OnInit, Optional } from '@angular/core';
import { Product } from '@spartacus/core';
import {
  OutletContextData,
  OutletPosition,
  ProductDetailOutlets,
  provideOutlet,
} from '@spartacus/storefront';
import { Observable, of } from 'rxjs';
import { map } from 'rxjs/operators';

@Component({
  selector: 'app-price-note',
  template: `
    <p *ngIf="product$ | async as product" class="price-note">
      Prezzo IVA inclusa per {{ product.name }}
    </p>
  `,
  imports: [NgIf, AsyncPipe],
})
export class PriceNoteComponent implements OnInit {
  product$: Observable<Product | undefined> = of(undefined);

  // @Optional: OutletContextData esiste solo se il componente è creato da un outlet
  constructor(
    @Optional() protected outlet?: OutletContextData<{ product: Product }>
  ) {}

  ngOnInit(): void {
    if (this.outlet?.context$) {
      // context$ (non context, deprecato) segue i cambi di [cxOutletContext]
      this.product$ = this.outlet.context$.pipe(
        map((context) => context?.product)
      );
    }
  }
}

@NgModule({
  imports: [PriceNoteComponent],
  providers: [
```

```
    provideOutlet({
      id: ProductDetailOutlets.PRICE, // 'PDP.PRICE'
      position: OutletPosition.AFTER, // default di provideOutlet
      component: PriceNoteComponent,
    }),
  ],
})
export class PriceNoteModule {}
// Modulo eager: la registrazione la esegue OutletModule.forRoot(), già in BaseStorefrontModule.
// In un modulo caricato lazy aggiungere OutletModule.forChild() agli imports.
```

Catene di export:

SIMBOLO	CATENA
Product	core-libs/core/public_api.ts:L49-L49 (sezione augmentable) e core-libs/core/public_api.ts:L26-L26 → core-libs/core/src/model/index.ts:L19-L19 → core-libs/core/src/model/product.model.ts:L167-L167
OutletContextData, OutletPosition	core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L9-L9 → core-libs/storefront/cms-structure/outlet/index.ts:L11-L11 → core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L13, core-libs/storefront/cms-structure/outlet/outlet.model.ts:L21-L44
provideOutlet	core-libs/storefront/cms-structure/outlet/index.ts:L13-L13 (export nominato) → core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L47
OutletModule	core-libs/storefront/cms-structure/outlet/index.ts:L12-L12 → core-libs/storefront/cms-structure/outlet/outlet.module.ts:L53-L89
ProductDetailOutlets	core-libs/storefront/public_api.ts:L21-L21 → core-libs/storefront/cms-components/index.ts:L12-L12 → core-libs/storefront/cms-components/product/index.ts:L16-L16 → core-libs/storefront/cms-components/product/product-outlets.model.ts:L7-L12

Perché funziona (file sorgente)

- `provideOutlet` è un multi-provider su `PROVIDE_OUTLET_OPTIONS` (core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L47); `registerOutletsFactory` risolve una factory per ogni opzione e la aggiunge all'`OutletService` con default `OutletPosition.AFTER` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L47).
- `OutletModule.forRoot()` esegue la registrazione in `APP_INITIALIZER`, `forChild()` in `MODULE_INITIALIZER` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88); `forRoot()` è già importato da `BaseStorefrontModule` (core-libs/storefront/base-storefront.module.ts:L43-L43); `MODULE_INITIALIZER` si comporta come `APP_INITIALIZER` se il modulo è eager (core-libs/core/src/lazy-loading/tokens.ts:L9-L20).
- `ProductSummaryComponent` apre l'outlet `outlets.PRICE` con contesto `{ product: product }` (core-libs/storefront/cms-components/product/product-summary/product-summary.component.html:L6-L16).
- `[cxOutlet]` costruisce BEFORE → REPLACE → AFTER (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L121-L125); per una `ComponentFactory` crea un injector con `OutletContextData` (`reference`, `position`, `context`, `context$`) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L196-L213).
- Modelli reali: `EstimatedDeliveryDateComponent` AFTER su `CartOutlets.ITEM_DETAILS` (feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L28-L35); `OrderSummaryComponent` con `@Optional()` e `context$` (feature-libs/cart/base/components/cart-

shared/order-summary/order-summary.component.ts:L26-L38); `provideOutlet` in un root module eager (feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L42-L59).

Limiti e trappole

- **Default di posizione diversi**: `provideOutlet` → AFTER, `cxOutletRef` senza `cxOutletPos` → REPLACE (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L35).
- **Modulo lazy senza** `OutletModule.forChild()`: la registrazione non avviene, perché `forRoot` gira una sola volta all'avvio (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88); vedi come fa il cart (feature-libs/cart/base/components/cart-base-components.module.ts:L30-L30).
- **Registro non reattivo**: un outlet già renderizzato non vede registrazioni successive finché non si ri-renderizza (cambio di `cxOutlet` o `OutletRendererService.render`) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L92-L96, core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L38). DA VERIFICARE a runtime per registrazioni da moduli lazy (dossier Q2).
- **REPLACE multiplo**: tutte le registrazioni REPLACE vengono rese, non solo la prima (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L130-L139).
- `context` **vs** `context$`: `context` è `@deprecated since 3.0` e statico (core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L43); `ngOnChanges` inoltra i cambi di `cxOutletContext` solo su `outletContext$` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L97-L99).
- `PROVIDE_OUTLET_OPTIONS` **è** `@private` ("We plan to drive the outlets by standard configuration", core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L10-L12): usare sempre `provideOutlet`, mai il token.
- Con `forChild` le deps non sono `Self`: DA VERIFICARE la possibile doppia registrazione quando l'injector lazy non ha propri `PROVIDE_OUTLET_OPTIONS` (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L79-L83; cap. 17).

3.5 R5 — Template in un outlet con `cxOutletRef` (replace / before / after)

Quando usarlo

Quando il contenuto da inserire è un pezzo di HTML legato a una pagina specifica (non un componente riutilizzabile), oppure vuoi avvolgere o sostituire un'intera **pagina** (outlet = nome del template), uno **slot** (outlet = `position`) o **ogni istanza di un tipo CMS** (outlet = `flexType`). Le registrazioni vivono quanto il componente che contiene i `ng-template`.

Esempio

> Esempio illustrativo, non presente nel repo. Il componente è pensato come `component` di una route (vedi 3.6).

```typescript
// faq-page.component.ts — esempio illustrativo, non presente nel repo
import { AsyncPipe, NgFor } from '@angular/common';
import { Component } from '@angular/core';
import {
  OutletPosition,
  OutletRefDirective,
  PageLayoutComponent,
} from '@spartacus/storefront';
```

```
@Component({
  selector: 'app-faq-page',
  template: `
    <cx-page-layout />

    <!-- prima dello slot Section2A (contesto: { components$ }) -->
    <ng-template
      [cxOutletRef]="'Section2A'"
      [cxOutletPos]="outletPosition.BEFORE"
      let-model
    >
      <nav class="faq-index">
        <a *ngFor="let c of model.components$ | async" [href]="'#' + c.uid">
          {{ c.uid }}
        </a>
      </nav>
    </ng-template>

    <!-- dopo ogni CMSParagraphComponent (contesto: { component }) -->
    <ng-template
      [cxOutletRef]="'CMSParagraphComponent'"
      [cxOutletPos]="outletPosition.AFTER"
      let-model
    >
      <a class="faq-top" [attr.id]="model.component.uid" href="#">Torna su</a>
    </ng-template>
  `,
  imports: [PageLayoutComponent, OutletRefDirective, NgFor, AsyncPipe],
})
export class FaqPageComponent {
  // l'enum serve al template: con strictTemplates cxOutletPos="before" non compila
  protected readonly outletPosition = OutletPosition;
}
```

Catene di export:

SIMBOLO	CATENA
OutletRefDirective	core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L9-L9 → core-libs/storefront/cms-structure/outlet/index.ts:L7-L7 → core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L17-L36
OutletPosition	vedi 3.4
PageLayoutComponent	core-libs/storefront/cms-structure/index.ts:L10-L10 → core-libs/storefront/cms-structure/page/index.ts:L9-L9 → core-libs/storefront/cms-structure/page/page-layout/index.ts:L8-L8 → core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L30-L58

Perché funziona (file sorgente)

- `OutletRefDirective` registra il proprio `TemplateRef` in `ngOnInit` e lo rimuove in `ngOnDestroy` (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L17-L36), quindi le registrazioni durano quanto *FaqPageComponent*.

- Un `TemplateRef` è reso con `{ $implicit: cxOutletContext }`, da cui `let-model` (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L175-L189).
- Gli outlet di sistema e i loro contesti: template/section con `templateName$`, `slots$`, `section$` (core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L1-L18); slot con `components$`, componente con `component` (core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L1-L17).
- Modello reale: la demo `TestOutletComponentComponent` mette `<cx-page-layout />` e tre `cxOutletRef` (before/after/replace) su `CMSParagraphComponent` nello stesso template (projects/storefrontapp/src/test-outlets/test-outlet-component/test-outlet-component.component.html:L1-L13), standalone con `PageLayoutComponent` + `OutletRefDirective` (projects/storefrontapp/src/test-outlets/test-outlet-component/test-outlet-component.component.ts:L10-L17); varianti su slot e template in projects/storefrontapp/src/test-outlets/test-outlet-slot/test-outlet-slot.component.html:L1-L27 e projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-template.component.html:L1-L25.

Limiti e trappole

- `cxOutletPos` **testuale e** `strictTemplates`: l'input è di tipo `OutletPosition`, un enum di stringhe (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L21-L22, core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L13). Con `strictTemplates` il compilatore controlla anche gli attributi testuali (angular:packages/compiler-cli/src/ngtsc/core/src/compiler.ts:L1097-L1097, angular:packages/compiler-cli/src/ngtsc/typecheck/src/ops/inputs.ts:L199-L202), e `cxOutletPos="before"` dà TS2322 (letterale stringa non assegnabile all'enum). Per questo l'esempio usa `[cxOutletPos]="outletPosition.BEFORE"`; la forma testuale della demo (5.10) compila solo perché l'app ha `strictTemplates: false` (projects/storefrontapp/tsconfig.app.json:L19-L21).
- **Chiavi stringa libere**: un refuso nel nome di slot/tipo/template non dà errori; la demo stessa legge `model.sections$` mentre il contesto espone `section$` (projects/storefrontapp/src/test-outlets/test-outlet-template/test-outlet-template.component.html:L7-L7, core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L5-L9).
- **REPLACE sul template di pagina** sostituisce tutto: `ng-content` e tutti gli slot (core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L1-L18).
- **Ordine di istanziazione**: la direttiva registra nel suo `ngOnInit`; il contenitore deve esistere prima o insieme all'outlet bersaglio (nella demo è nello stesso template di `<cx-page-layout />`) (core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L29-L31).
- **Contesto fissato**: per i template `$implicit` è impostato alla creazione; se il contesto cambia, la vista non si aggiorna (usare gli observable del contesto, come `components$`) (core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L97-L99, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L176-L181).
- **Registro globale**: mentre *FaqPageComponent* è vivo, il template "after `CMSParagraphComponent`" vale per **ogni** paragrafo renderizzato, anche fuori dalla pagina FAQ (es. header/footer), perché `OutletService` è indicizzato solo per nome (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L42).

3.6 R6 — Route custom per una content page

Quando usarlo

Quando una content page CMS deve avere un URL diverso dalla sua label (o più alias), un nome semantico per costruire link con `cxUrl`, o un componente Angular proprio attorno al layout (come *FaqPageComponent* di R5). Senza route custom, una content page è già raggiungibile tramite la route `'**'` con l'URL uguale alla

label (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25, core-libs/core/src/routing/store/reducers/router.reducer.ts:L198-L205).

Esempio

> Esempio illustrativo, non presente nel repo. `/faq` è una label ipotetica di content page nel backend.

TYPESCRIPT

```typescript
// faq-routing.module.ts — esempio illustrativo, non presente nel repo
import { NgModule } from '@angular/core';
import { RouterModule } from '@angular/router';
import { CmsRoute, RoutingConfig, provideConfig } from '@spartacus/core';
import { CmsPageGuard, PageLayoutComponent } from '@spartacus/storefront';

const routes: CmsRoute[] = [
  {
    // path sostituito da ConfigurableRoutesService in base a routing.routes.faq
    path: 'faq',
    canActivate: [CmsPageGuard],
    component: PageLayoutComponent, // oppure FaqPageComponent di R5
    data: {
      cxRoute: 'faq', // nome semantico: chiave in routing.routes
      pageLabel: '/faq', // PageContext { id: '/faq', type: ContentPage }
    },
  },
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  providers: [
    provideConfig(<RoutingConfig>{
      routing: {
        routes: {
          // più alias: la route ottiene un matcher costruito dai paths
          faq: { paths: ['faq', 'domande-frequenti'] },
        },
      },
    }),
  ],
})
export class FaqRoutingModule {}
// Link semantico in un template: [routerLink]="{ cxRoute: 'faq' } | cxUrl"
```

Catene di export:

SIMBOLO	CATENA
CmsRoute	core-libs/core/public_api.ts:L30-L30 → core-libs/core/src/routing/index.ts:L12-L12 → core-libs/core/src/routing/models/cms-route.ts:L15-L17
RoutingConfig	core-libs/core/public_api.ts:L30-L30 → core-libs/core/src/routing/index.ts:L7-L7 → core-libs/core/src/routing/configurable-routes/index.ts:L7-L7 → core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L22-L45

SIMBOLO	CATENA
provideConfig	come in 3.1
CmsPageGuard	core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L8-L8 → core-libs/storefront/cms-structure/guards/index.ts:L7-L7 → core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L26-L84
PageLayoutComponent	come in 3.5

Perché funziona (file sorgente)

- RoutesConfig è un dizionario aperto "allows User's custom pages" (core-libs/core/src/routing/configurable-routes/routes-config.ts:L11-L13); RouteConfig ha paths, matchers, paramsMapping, disabled, protected, authFlow (core-libs/core/src/routing/configurable-routes/routes-config.ts:L15-L57).
- ConfigurableRoutesService.init() gira in provideAppInitializer (core-libs/core/src/routing/routing.module.ts:L93-L95) e fa router.resetConfig(configureRoutes(router.config)) (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L37-L68); configureRoute legge data.cxRoute e con un solo path imposta path, con più path crea un matcher (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L109).
- CustomSerializer marca cmsRequired perché CmsPageGuard ha guardName === 'CmsPageGuard' e calcola il PageContext da data.pageLabel come CONTENT_PAGE (core-libs/core/src/routing/store/reducers/router.reducer.ts:L137-L149, core-libs/core/src/routing/store/reducers/router.reducer.ts:L186-L195).
- L'adapter OCC chiede la content page con pageLabelOrId = context.id (core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L68-L100); CmsPageGuard carica la pagina e delega a canActivatePage o al not-found (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L26-L84).
- Modelli reali: route search con path: null, CmsPageGuard, PageLayoutComponent, data: { pageLabel, cxRoute } (core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L13-L53); orderGuest in una feature con defaultOrderRoutingConfig (feature-libs/order/root/order-root.module.ts:L82-L91, feature-libs/order/root/config/default-order-routing-config.ts:L9-L22); override applicativo dei paths di product (projects/storefrontapp/src/app/private/private.providers.ts:L33-L41). UrlPipe (cxUrl) trasforma { cxRoute, params } in link (core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts:L11-L17).
- Il meta resolver di default per le content page è ContentPageMetaResolver (core-libs/core/src/cms/page/content-page-meta.resolver.ts:L29-L65); per titoli/description custom vedi 52 e il cap. 40.

Limiti e trappole

- path **statico sovrascritto**: con paths di lunghezza 1 il path diventa paths[0], con più paths il path viene cancellato e sostituito da un matcher (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L106). La libreria usa path: null con // @ts-ignore (core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L17-L18); senza config la route ottiene un matcher da [] (mai matchato) e un warning dev "Could not configure the named route" (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L153-L179).
- **Niente** productCode**/**categoryCode**/**brandCode **nei parametri** se vuoi una content page: hanno precedenza su pageLabel (core-libs/core/src/routing/store/reducers/router.reducer.ts:L186-L195).
- pageLabel **va sulla route foglia**: è letto solo dall'ultimo snapshot figlio, mentre cxRoute da qualsiasi antenato (core-libs/core/src/routing/store/reducers/router.reducer.ts:L120-L133, core-libs/core/src/routing/store/reducers/router.reducer.ts:L193-L194).

- **Non avvolgere** `CmsPageGuard` **in un guard custom**: `cmsRequired` dipende dal `guardName` statico (core-libs/core/src/routing/store/reducers/router.reducer.ts:L137-L149, core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L27-L27).
- **Route in** `loadChildren`: `init()` gira una volta e ricorre solo sui `children` (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L37-L68): route con `cxRoute` dentro moduli caricati dopo non vengono configurate.
- `configureRoute` **muta l'oggetto route** (`delete route.path`, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L84-L99): non riusare lo stesso array `routes` altrove.
- **Wildcard in coda**: la route `'**'` è aggiunta in coda in APP_INITIALIZER (core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L18-L23): route aggiunte dopo a runtime in coda sarebbero oscurate.
- **Pagina inesistente**: il guard carica la pagina `notFound` come content page; se `notFound` non è configurata ritorna `false` (core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L107-L143).
- `loadStrategy: ONCE` evita il reload della pagina CMS alla rinavigazione (core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L89-L91, core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L11-L20).
- `DA VERIFICARE` il valore esatto da usare come `pageLabel` per la propria installazione: deve coincidere con label o id della content page nel backend (l'adapter invia `pageLabelOrId`, ma la risoluzione label/id avviene in SAP Commerce, fuori da questo repo).

3.7 R7 — Page layout per un template nuovo (e pagina statica)

Quando usarlo

Quando il backend introduce un **nuovo page template** (o nuovi slot) e Spartacus deve sapere quali slot disegnare e in che ordine, eventualmente diverso per breakpoint. In aggiunta, con `CmsStructureConfig` puoi definire la pagina interamente lato storefront (`ignoreBackend: true`) per pagine "di servizio" o prototipi.

Esempio

> Esempio illustrativo, non presente nel repo. *FaqPageTemplate*, *FaqIntro*, *FaqBody*, *FaqAside* sono nomi ipotetici.

TYPESCRIPT

```typescript
// faq-layout.module.ts — esempio illustrativo, non presente nel repo
import { NgModule } from '@angular/core';
import { CmsStructureConfig, provideConfig } from '@spartacus/core';
import { LayoutConfig } from '@spartacus/storefront';

@NgModule({
  providers: [
    // 1) Layout del nuovo template: base (mobile first) senza FaqAside,
    //    da md in su (md, lg, xl) anche FaqAside
    provideConfig(<LayoutConfig>{
      layoutSlots: {
        FaqPageTemplate: {
          slots: ['FaqIntro', 'FaqBody'],
          pageFold: 'FaqBody',
          md: { slots: ['FaqIntro', 'FaqBody', 'FaqAside'] },
        },
```

```
      },
    }),

    // 2) (Opzionale) pagina statica: il backend non viene chiamato
    provideConfig(<CmsStructureConfig>{
      cmsStructure: {
        components: {
          FaqIntroParagraph: {
            uid: 'FaqIntroParagraph',
            typeCode: 'CMSParagraphComponent',
            flexType: 'CMSParagraphComponent',
            content: '<p>Le risposte alle domande più comuni.</p>',
          },
        },
        pages: [
          {
            ignoreBackend: true,
            pageId: '/faq', // = id del PageContext (pageLabel della route R6)
            template: 'FaqPageTemplate',
            slots: {
              FaqIntro: { componentIds: ['FaqIntroParagraph'] },
              FaqBody: {},
            },
          },
        ],
      },
    }),
  ],
})
export class FaqLayoutModule {}
```

Catene di export:

SIMBOLO	CATENA
LayoutConfig	core-libs/storefront/public_api.ts:L26-L26 → core-libs/storefront/layout/index.ts:L9-L9 → core-libs/storefront/layout/config/index.ts:L8-L8 → core-libs/storefront/layout/config/layout-config.ts:L90-L125
CmsStructureConfig	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L8-L8 → core-libs/core/src/cms/config/index.ts:L9-L9 → core-libs/core/src/cms/config/cms-structure.config.ts:L75-L85
provideConfig	come in 3.1

Perché funziona (file sorgente)

- LayoutConfig (astratta, useExisting: Config) espone layoutSlots (core-libs/storefront/layout/config/layout-config.ts:L90-L125); SlotConfig = { slots?, pageFold? }, SlotGroup per lg/md/sm/xs (core-libs/storefront/layout/config/layout-config.ts:L36-L61).
- PageLayoutComponent usa la section se presente, altrimenti il template della pagina (core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L30-L58); PageLayoutService.

`resolveSlots` usa `layoutSlots[page.template]` e filtra gli slot mantenendo solo quelli presenti nella pagina (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L106-L127).

- Config per breakpoint: specifica se presente, altrimenti la più vicina tra i breakpoint più piccoli, altrimenti quella di base (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L217-L243); breakpoint di default in core-libs/storefront/layout/config/default-layout.config.ts:L9-L19. La config dei template passa da qui (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L166-L171).
- `CmsPageConnector.get` salta l'adapter se `shouldIgnoreBackend(pageContext.id)` e fonde sempre la struttura statica (core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L30-L70); la pagina è cercata per `pageId === PageContext.id` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L84-L92); `CmsPageSlotConfig = { componentIds?, properties? }` (core-libs/core/src/cms/config/cms-structure.config.ts:L63-L66).
- `PageTemplateDirective` aggiunge il nome del template come classe CSS (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L114-L119) e `PageSlotComponent` la `position` dello slot (core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L44-L46): i selettori CSS del nuovo layout sono `.FaqPageTemplate` e `.FaqIntro` (stili: cap. 38).
- Modelli reali: `layoutConfig` di riferimento (es. `ContentPage1Template: { slots: ['Section2A', 'Section2B'] }`) (core-libs/storefront/recipes/config/layout-config.ts:L67-L69), fornito nell'app con `provideConfigFactory(layoutConfigFactory)` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L44-L49); pagine statiche con `ignoreBackend: true` della demo (projects/storefrontapp/src/test-outlets/test-outlet-cms-page.config.ts:L33-L76).

Limiti e trappole

- **Mobile first**: una config per breakpoint vale anche per tutti i breakpoint più grandi che non ne hanno una propria, perché i breakpoint sono ordinati dal più piccolo al più grande e si tiene l'ultimo più piccolo con `slots` (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L229-L242, core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L134-L136). Un override su `xs` vale quindi anche per sm/md/lg/xl e rende inutile la config di base; per avere slot diversi solo su xs si mette la variante ridotta come base e quella completa su un breakpoint più grande, come fa `layoutConfig` con `lg`.
- **Lo slot deve esistere nella pagina**: uno slot in `layoutSlots` che la pagina non contiene viene filtrato (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L117-L119). Nell'esempio *FaqAside* non è nella pagina statica e quindi non viene reso.
- **Template senza config**: tutti gli slot della pagina nell'ordine delle chiavi + warning dev; una section senza config non rende nulla (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L120-L126, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L251-L273).
- **Array sostituiti**: un `slots: [...]` rimpiazza l'intera lista del template (core-libs/core/src/config/utils/deep-merge.ts:L36-L45); `provideCmsStructure({ pageTemplate, pageSlotPosition })` genera `slots: [pageSlotPosition]` e può azzerare gli altri slot se fuso dopo (core-libs/storefront/cms-structure/utils/cms-structure.util.ts:L78-L90).
- **Ordine rispetto a** `layoutConfigFactory`: nell'app demo è un `ConfigChunk` (projects/storefrontapp/src/app/spartacus/spartacus-configuration.module.ts:L46-L46); un chunk `layoutSlots` fornito prima di esso verrebbe sovrascritto per le chiavi in comune (core-libs/core/src/config/config-tokens.ts:L37-L39). Per un template nuovo il conflitto non c'è; per modificare un template esistente fornisci la tua config **dopo**.
- `cmsStructure.pages` **è un array**: più chunk con `pages` non si sommano, l'ultimo vince (core-libs/core/src/cms/config/cms-structure.config.ts:L82-L82).

- `pageId` **= id del** `PageContext`: con `pageLabel` è il suo valore, per content page da URL include la `/` iniziale (core-libs/core/src/cms/services/cms-structure-config.service.ts:L84-L92, core-libs/core/src/routing/store/reducers/router.reducer.ts:L193-L205).
- **Il CMS reale prevale sugli slot statici**: `mergeSlots` aggiunge uno slot configurato solo se la pagina non lo ha già (core-libs/core/src/cms/services/cms-structure-config.service.ts:L124-L167).
- `layoutConfig` (costante) è deprecato a favore di `layoutConfigFactory` (core-libs/storefront/recipes/config/layout-config.ts:L25-L25); `layoutConfigFactory` rimuove i `pageFold` (core-libs/storefront/recipes/config/layout-config.ts:L158-L198).

---

4. Diagrammi

4.1 Dal mapping al componente (R1, R2, R3)

■ **TAVOLA 51.1** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L65-L99, core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L6-L16, core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L67-L116, core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50, core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L49-L65, core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L35-L59.

4.2 Registrazione e rendering di un outlet (R4, R5)

■ **TAVOLA 51.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L47, core-libs/storefront/cms-structure/outlet/outlet.module.ts:L29-L88, core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L29-L31, core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L42, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L121-L213.

4.3 Route custom, pagina e layout (R6, R7)

■ **TAVOLA 51.3** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: core-libs/core/src/routing/routing.module.ts:L93-L95, core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L37-L109, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25, core-libs/core/src/routing/store/reducers/router.reducer.ts:L112-L220, core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L26-L84, core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L30-L70, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L106-L127.

5. Estratti di codice reale

5.1 Sostituzione di mapping in una feature (image-zoom)

feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom.module.ts:L51-L60

```typescript
  providers: [
    provideDefaultConfig(defaultProductImageZoomLayoutConfig),
    provideDefaultConfig(<CmsConfig>{
      cmsComponents: {
        ProductImagesComponent: {
          component: ProductImageZoomProductImagesComponent,
        },
      },
    }),
  ],
```

Una libreria usa `provideDefaultConfig`; nell'app, per R1, usa `provideConfig`.

5.2 Estensione con riuso di membri protected (image-zoom)

feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-product-images/product-image-zoom-product-images.component.ts:L21-L49

```typescript
@Component({
  selector: 'cx-product-images',
  templateUrl: './product-image-zoom-product-images.component.html',
  changeDetection: ChangeDetectionStrategy.OnPush,
  imports: [
    NgIf,
    LcpContextDirective,
    MediaComponent,
    ProductImageZoomTriggerComponent,
    CarouselScrollingComponent,
    FocusableCarouselItemDirective,
    AsyncPipe,
    TranslatePipe,
  ],
})
export class ProductImageZoomProductImagesComponent extends ProductImagesComponent {
  expandImage = new BehaviorSubject(false);
  selectedIndex: number | undefined;

  product$: Observable<Product> = this.product$;

  constructor(protected currentProductService: CurrentProductService) {
    super(currentProductService);
  }
```

```
  openImage(item: any): void {
    this.mainMediaContainer.next(item);
    this.selectedIndex = this.mainMediaContainer.value?.zoom?.galleryIndex;
  }
```

5.3 Estensione con costruttore (BreadcrumbComponent)

core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.component.ts:L32-L55

```typescript
// ...
  constructor(
    public component: CmsComponentData<CmsBreadcrumbsComponent>,
    protected pageMetaService: PageMetaService,
    private translation: TranslationService
  ) {
    super(component, pageMetaService);
  }

  ngOnInit(): void {
    super.ngOnInit();
    this.setCrumbs();
  }
// ...
```

(estratto abbreviato; il range copre la dichiarazione della classe, il costruttore e `ngOnInit`.)

5.4 Componente CMS canonico e suo mapping (Paragraph)

core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L20-L27

```typescript
@Component({
  selector: 'cx-paragraph',
  templateUrl: './paragraph.component.html',
  changeDetection: ChangeDetectionStrategy.OnPush,
  imports: [NgIf, AsyncPipe, SupplementHashAnchorsPipe],
})
export class ParagraphComponent {
  protected sanitizer = inject(DomSanitizer);
```

core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L44-L51

```typescript
  constructor(
    public component: CmsComponentData<CmsParagraphComponent>,
    protected router: Router
  ) {}
```

```
public bypassSecurityTrustHtml(html: string = ''): SafeHtml {
  return this.sanitizer.bypassSecurityTrustHtml(html);
}
```

core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L21-L32

TYPESCRIPT

```
providers: [
  provideDefaultConfig(<CmsConfig>{
    cmsComponents: {
      CMSParagraphComponent: {
        component: ParagraphComponent,
      },
      CMSTabParagraphComponent: {
        component: ParagraphComponent,
      },
    },
  }),
],
```

5.5 Injector del componente CMS

core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L28-L50

TYPESCRIPT

```
public getInjector(
  type: string,
  uid: string,
  parentInjector?: Injector
): Injector {
  const configProviders =
    this.cmsComponentsService.getMapping(type)?.providers ?? [];
  return Injector.create({
    providers: [
      {
        provide: CmsComponentData,
        useFactory: (dataProvider: ComponentDataProvider) => ({
          uid,
          data$: dataProvider.get(uid, type),
        }),
        deps: [ComponentDataProvider],
      },
      provideLcpPresenceForCmsComponent(),
      ...configProviders,
    ],
    parent: parentInjector ?? this.injector,
  });
}
```

5.6 Mapping di feature come default sovrascrivibile

core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L133

```typescript
        tap((featureComponentMapping) => {
          // We treat cms mapping configuration from a feature as a default,
          // that can be overridden by app/static configuration
          this.mappings[componentType] = deepMerge(
            {},
            featureComponentMapping,
            staticConfig
          );
```

5.7 `provideOutlet` con posizione esplicita

feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L28-L35

```typescript
providers: [
  provideDefaultConfig(defaultOccCartWithEddConfig),
  provideOutlet({
    id: CartOutlets.ITEM_DETAILS,
    position: OutletPosition.AFTER,
    component: EstimatedDeliveryDateComponent,
  }),
],
```

5.8 Componente che legge `OutletContextData.context$`

feature-libs/cart/base/components/cart-shared/order-summary/order-summary.component.ts:L26-L34

```typescript
constructor(@Optional() protected outlet?: OutletContextData<any>) {}

ngOnInit(): void {
  if (this.outlet?.context$) {
    this.subscription.add(
      this.outlet.context$.subscribe((context) => (this.cart = context))
    );
  }
}
```

5.9 Registrazione degli outlet (default AFTER)

core-libs/storefront/cms-structure/outlet/outlet.module.ts:L34-L45

```typescript
const result = () => {
  (providedOutletOptions ?? []).forEach((options) => {
    const factory = componentFactoryResolver.resolveComponentFactory(
```

```
        options.component
      );
      outletService.add(
        options.id,
        factory,
        options.position ?? OutletPosition.AFTER
      );
    });
  };
```

5.10 `cxOutletRef` nella demo

La forma testuale `cxOutletPos="before"` compila solo perché la demo ha `strictTemplates: false` (projects/storefrontapp/tsconfig.app.json:L19-L21); in un'app strict va legata all'enum, come in 3.5.

projects/storefrontapp/src/test-outlets/test-outlet-component/test-outlet-component.component.html:L1-L13

HTML

```
<cx-page-layout />

<ng-template [cxOutletRef]="testComponent" cxOutletPos="before" let-model>
  <p>Before component, uid: {{ model.component.uid }}</p>
</ng-template>

<ng-template [cxOutletRef]="testComponent" cxOutletPos="after" let-model>
  <p>After component, uid: {{ model.component.uid }}</p>
</ng-template>

<ng-template [cxOutletRef]="testComponent" let-model>
  <p>Replace component, uid: {{ model.component.uid }}</p>
</ng-template>
```

projects/storefrontapp/src/test-outlets/test-outlet-component/test-outlet-component.module.ts:L17-L28

TYPESCRIPT

```
@NgModule({
  imports: [
    CommonModule,
    PageLayoutModule,
    OutletRefModule,
    RouterModule.forChild([
      {
        path: 'test/outlet/component',
        component: TestOutletComponentComponent,
        canActivate: [CmsPageGuard],
      },
    ]),
```

5.11 Route semantica con `pageLabel`

core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L13-L22

TYPESCRIPT

```typescript
@NgModule({
  imports: [
    RouterModule.forChild([
      {
        // @ts-ignore
        path: null,
        canActivate: [CmsPageGuard],
        component: PageLayoutComponent,
        data: { pageLabel: 'search', cxRoute: 'search' },
      },
```

5.12 Override applicativo dei paths di una route

projects/storefrontapp/src/app/private/private.providers.ts:L33-L41

TYPESCRIPT

```typescript
    // Custom routing configuration for e2e testing:
    routing: {
      routes: {
        product: {
          paths: ['product/:productCode/:name', 'product/:productCode'],
          paramsMapping: { name: 'slug' },
        },
      },
    },
```

5.13 Come `configureRoute` applica la config

core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L109

TYPESCRIPT

```typescript
  protected configureRoute(route: Route): Route {
    const routeName = this.getRouteName(route);
    if (routeName) {
      const routeConfig = this.routingConfigService.getRouteConfig(routeName);
      this.validateRouteConfig(routeConfig, routeName, route);

      if (routeConfig?.disabled) {
        delete route.path;
        return {
          ...route,
          matcher: this.urlMatcherService.getFalsy(),
        };
      } else if (routeConfig?.matchers) {
        delete route.path;
        return {
```

```
          ...route,
          matcher: this.resolveUrlMatchers(route, routeConfig?.matchers),
        };
      } else if (routeConfig?.paths?.length === 1) {
        delete route.matcher;
        return { ...route, path: routeConfig?.paths[0] };
      } else {
        delete route.path;
        return {
          ...route,
          matcher: this.urlMatcherService.getFromPaths(
            routeConfig?.paths || []
          ),
        };
      }
    }
    return route; // if route doesn't have a name, just pass the original route
  }
```

5.14 PageContext da pageLabel

core-libs/core/src/routing/store/reducers/router.reducer.ts:L186-L196

TYPESCRIPT

```
      const { params } = state;
      if (params['productCode']) {
        context = { id: params['productCode'], type: PageType.PRODUCT_PAGE };
      } else if (params['categoryCode']) {
        context = { id: params['categoryCode'], type: PageType.CATEGORY_PAGE };
      } else if (params['brandCode']) {
        context = { id: params['brandCode'], type: PageType.CATEGORY_PAGE };
      } else if (state.data.pageLabel !== undefined) {
        context = { id: state.data.pageLabel, type: PageType.CONTENT_PAGE };
      }
    }
```

5.15 Pagine statiche nella demo

projects/storefrontapp/src/test-outlets/test-outlet-cms-page.config.ts:L33-L51

TYPESCRIPT

```
export function testOutletPagesCmsContentConfig(): CmsStructureConfig {
  return {
    cmsStructure: {
      components: {
        ...paragraphComponents,
        ...linkComponents,
      },
      pages: [
```

```
        {
          slots: {
            Section2A: {
              componentIds: ['Paragraph1'],
            },
            Section2B: {},
          },
          ignoreBackend: true,
          pageId: '/test/outlet/slot',
          template: 'ContentPage1Template',
        },
```

projects/storefrontapp/src/test-outlets/test-outlet-cms-page.config.ts:L9-L18

TYPESCRIPT

```typescript
export const paragraphComponents: {
  [key: string]: ContentSlotComponentData | any;
} = {
  Paragraph1: {
    uid: 'Paragraph1Component',
    typeCode: 'CMSParagraphComponent',
    flexType: 'CMSParagraphComponent',
    content: 'Paragraph1',
  },
};
```

5.16 Filtro degli slot dal layout

core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L106-L127

TYPESCRIPT

```typescript
  private resolveSlots(
    page: Page,
    section: string | undefined,
    breakpoint: BREAKPOINT
  ): string[] {
    const config = this.getSlotConfig(
      page.template ?? '',
      'slots',
      section,
      breakpoint
    );
    if (config && config.slots) {
      const pageSlots = page.slots ? Object.keys(page.slots) : [];
      return config.slots.filter((slot) => pageSlots.includes(slot));
    } else if (!section) {
      this.logMissingLayoutConfig(page);
      return page.slots ? Object.keys(page.slots) : [];
    } else {
      this.logMissingLayoutConfig(page, section);
```

```
      return [];
    }
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cmsComponents[type].component	classe / `() => import()`	—	core-libs/core/src/cms/config/cms-config.ts:L61-L61	core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L27-L34, core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L38-L52
cmsComponents[type].providers	StaticProvider[]	[]	core-libs/core/src/cms/config/cms-config.ts:L62-L73	core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L33-L46
cmsComponents[type].data	T extends CmsComponent	—	core-libs/core/src/cms/config/cms-config.ts:L79-L85	core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L39-L51
cmsComponents[type].guards	any[]	—	core-libs/core/src/cms/config/cms-config.ts:L77-L77	core-libs/storefront/cms-structure/services/cms-components.service.ts:L253-L261
cmsComponents[type].childRoutes	Route[] o { parent, children }	—	core-libs/core/src/cms/config/cms-config.ts:L45-L58	core-libs/storefront/cms-structure/services/cms-components.service.ts:L205-L248
cmsComponents[type].disableSSR	boolean	—	core-libs/core/src/cms/config/cms-config.ts:L75-L75	core-libs/storefront/cms-structure/services/cms-components.service.ts:L187-L190
cmsComponents[type].i18nKeys	string[]	—	core-libs/core/src/cms/config/cms-config.ts:L76-L76	core-libs/storefront/cms-structure/services/cms-components.service.ts:L266-L276
cmsComponents[type].deferLoading	DeferLoadingStrategy	—	core-libs/core/src/cms/config/cms-config.ts:L87-L101	core-libs/storefront/cms-structure/services/cms-components.service.ts:L195-L200
featureModules[name]	FeatureModuleConfig | string	—	core-libs/core/src/cms/config/cms-config.ts:L109-L129	core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L76

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
cmsStructure.{components,pages,slots}	oggetti / array	—	core-libs/core/src/cms/config/cms-structure.config.ts:L79-L85	core-libs/core/src/cms/services/cms-structure-config.service.ts:L84-L92, core-libs/core/src/cms/services/cms-structure-config.service.ts:L124-L167
cmsStructure.pages[].ignoreBackend	boolean	—	core-libs/core/src/cms/config/cms-structure.config.ts:L19-L25	core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L30-L70
routing.routes[name]	RouteConfig	core-libs/storefront/cms-structure/routing/default-routing-config.ts:L17-L73	core-libs/core/src/routing/configurable-routes/routes-config.ts:L15-L57	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L109
routing.loadStrategy	RouteLoadStrategy	ALWAYS	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L11-L45	core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L45-L47, core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L89-L91
route data.cxRoute	string	—	convenzione	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L149-L151, core-libs/core/src/routing/store/reducers/router.reducer.ts:L126-L128
route data.pageLabel	string	—	core-libs/core/src/routing/models/cms-route.ts:L10-L13	core-libs/core/src/routing/store/reducers/router.reducer.ts:L193-L194
layoutSlots[template|section]	LayoutSlotConfig	layoutConfigFactory() (app)	core-libs/storefront/layout/config/layout-config.ts:L90-L125	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L106-L127, core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L175-L243
breakpoints	LayoutBreakPoints	core-libs/storefront/layout/config/default-layout.config.ts:L9-L19	core-libs/storefront/layout/config/layout-config.ts:L90-L125	BreakpointService (cap. 18)
provideOutlet({ position })	OutletPosition	AFTER	core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L17-L32	core-libs/storefront/cms-structure/outlet/outlet.module.ts:L39-L43
cxOutletPos	OutletPosition	REPLACE	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L17-L36	core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L42

7. Commenti e note notevoli dal sorgente

- "To provide default configuration in libraries provideDefaultConfig should be used instead." — core-libs/core/src/config/config-providers.ts:L14-L14. È la ragione per cui le ricette usano `provideConfig` nell'app.
- "General rule is, that all config provided in libraries should be provided as default config." — core-libs/core/src/config/config-tokens.ts:L58-L58.
- Provider configurabili dei componenti: "these services cannot be extended with the native Angular DI system. The configurable component providers mitigate this downside" — core-libs/core/src/cms/config/cms-config.ts:L62-L72.
- "We treat cms mapping configuration from a feature as a default, that can be overridden by app/static configuration" — core-libs/storefront/cms-structure/services/cms-components.service.ts:L127-L128.
- Snapshot della config CMS: "we want to grab cms configuration available at config initialization phase as lazy-loaded modules can affect global configuration resulting in non-deterministic state" — core-libs/storefront/cms-structure/services/cms-components.service.ts:L60-L69.
- `PROVIDE_OUTLET_OPTIONS`: "@private We plan to drive the outlets by standard configuration" — core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L10-L12.
- `OutletContextData.context`: "@deprecated since 3.0 - use `context$` observable instead" — core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L38.
- `layoutConfig`: "@deprecated Use `layoutConfigFactory` and `provideConfigFactory(layoutConfigFactory)` instead." e "if you introduce custom page templates and/or slots, you most likely need to further adjust or replace this configuration." — core-libs/storefront/recipes/config/layout-config.ts:L21-L25.
- `CmsRoutesImplService`: "This service should be exposed in public API only after the refactor planned in https://github.com/SAP/spartacus/issues/7070" — core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L31.
- `LazyComponentHandler`: `PendingTasks` "prevents SSR from serializing and destroying the app while the import() is still in flight (which would cause NG0205 — destroyed injector)" — core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L61-L65.
- Commento sull'ordine delle route PLP: "So not to shadow the specific 'brand' route, the 'category' is the last route in the sequence." — core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L31-L32.
- Demo: "CAUTION: IT IS NOT MEANT FOR CUSTOMERS' APPLICATIONS!" — projects/storefrontapp/src/app/private/private.providers.ts:L14-L18. I `test-outlets` sono un modello di forma, non codice da copiare in produzione.

8. Extension point

Questo capitolo **è** la ricetta per gli extension point del CMS storefront; la tabella riassume dove ciascuno è spiegato a fondo e quale ricetta lo applica.

EXTENSION POINT	SIMBOLO PUBBLICO	SPIEGAZIONE	RICETTA
Mapping componente	`CmsConfig.cmsComponents`	15, 16	3.1, 3.3
Dati del componente	`CmsComponentData`	16	3.1–3.3

EXTENSION POINT	SIMBOLO PUBBLICO	SPIEGAZIONE	RICETTA
Componenti di libreria estendibili	`ParagraphComponent`, `ProductImagesComponent`, `NavigationComponent`, …	45	3.2
Outlet programmatici	`provideOutlet`, `OutletPosition`, `OutletContextData`, `OutletModule`	17	3.4
Outlet da template	`OutletRefDirective`, `OutletService`, `OutletRendererService`	17	3.5
Route semantiche e CMS	`RoutingConfig`, `CmsRoute`, `CmsPageGuard`, `PageLayoutComponent`	14	3.6
Layout	`LayoutConfig`, `PAGE_LAYOUT_HANDLER`	18	3.7
Pagine/componenti statici	`CmsStructureConfig`, `provideCmsStructure`	15	3.3, 3.7
Meta delle pagine	`PageMetaResolver`, `ContentPageMetaResolver`	40	52

Ricette collegate in Parte III: 50 — configurazione e DI (ordine dei chunk, `provideConfigFactory`), 52 — OCC e meta resolver, 53 — feature lazy, stili, traduzioni (mapping in feature lazy con `featureModules`, stili dei nuovi template), 54 — API pubblica vs interna (perché importare solo da entry point pubblici).

Extension point **non** pubblici da evitare: `PROVIDE_OUTLET_OPTIONS` e `registerOutletsFactory` (non esportati, `core-libs/storefront/cms-structure/outlet/index.ts:L12-L13`), `ComponentDataProvider`, `WebComponentHandler`, `CmsRoutesImplService` (non esportati: dossier E.2; `core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L33`). Per personalizzare le child routes si fornisce il token pubblico `CmsRoutesService` (`core-libs/storefront/cms-structure/services/cms-routes.service.ts:L11-L24`). Per trasformare gli slot a runtime c'è il multi-token `PAGE_LAYOUT_HANDLER` (`core-libs/storefront/cms-structure/page/page-layout/page-layout-handler.ts:L11-L18`), usato ad esempio da `CartPageLayoutHandler` (`feature-libs/cart/base/components/cart-base-components.module.ts:L41-L45`).

---

9. Trappole, casi limite, comportamenti controintuitivi

1. `provideDefaultConfig` **in app** per sovrascrivere un mapping: dipende dall'ordine dei provider; usare `provideConfig` (`core-libs/core/src/config/config-tokens.ts:L10-L28`).
2. **Merge parziale dei mapping**: i campi non ridefiniti si ereditano; gli array ridefiniti si sostituiscono (`core-libs/core/src/config/utils/deep-merge.ts:L36-L45`).
3. **Chiave di mapping =** `flexType`, non sempre `typeCode` (`core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L151-L158`).
4. **Mapping mancante**: nessun errore in produzione (`core-libs/storefront/cms-structure/services/cms-components.service.ts:L170-L178`).
5. **Snapshot statico della config CMS**: i mapping aggiunti tardi non sono visti (`core-libs/storefront/cms-structure/services/cms-components.service.ts:L60-L69`).
6. **Costruttori nelle sottoclassi**: vanno ripetuti tutti i parametri; meglio non dichiararne (`angular:packages/core/src/render3/di.ts:L918-L935`).
7. `data$` **emette prima i dati statici** (`core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L44-L51`) ed è `EMPTY` per componenti senza uid e senza `data` (`core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L55-L57`); i componenti di `provideCmsStructure` hanno invece `uid = componentId` e leggono da `getComponentData(uid)` (`core-libs/core/src/cms/services/cms-structure-config.service.ts:L180-L180`).
8. **Default di posizione**: AFTER per `provideOutlet`, REPLACE per `cxOutletRef` (`core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42`, `core-libs/storefront/cms-structure/outlet/outlet.`

service.ts:L34-L34).

9. `OutletModule.forChild()` **obbligatorio nei moduli lazy** (core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88).
10. **Registro outlet globale e non reattivo** (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L42, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L92-L96).
11. `path` **della route custom sovrascritto** dalla config semantica (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L95-L106).
12. **Parametri con nomi "riservati"** (`productCode`, `categoryCode`, `brandCode`) cambiano il tipo di pagina (core-libs/core/src/routing/store/reducers/router.reducer.ts:L186-L195).
13. `CmsPageGuard` **riconosciuto per** `guardName` (core-libs/core/src/routing/store/reducers/router.reducer.ts:L137-L149).
14. **Slot configurati ma assenti nella pagina** non vengono resi (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L117-L119).
15. `cmsStructure.pages` **e** `layoutSlots[...].slots` **sono array**: l'ultimo chunk vince (core-libs/core/src/cms/config/cms-structure.config.ts:L82-L82, core-libs/core/src/config/utils/deep-merge.ts:L36-L45).
16. `pageId` **statico deve coincidere con l'id del** `PageContext`, `/` iniziale inclusa per le content page da URL (core-libs/core/src/cms/services/cms-structure-config.service.ts:L84-L92, core-libs/core/src/routing/store/reducers/router.reducer.ts:L198-L205).

---

10. Come fare debug

- **Componente non disegnato**: in dev mode cercare in console "No component implementation found for the CMS component type" (core-libs/storefront/cms-structure/services/cms-components.service.ts:L173-L176) e "Can't resolve handler for component mapping" (core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L49-L65). Verificare il `flexType` reale nella risposta OCC della pagina (normalizer: core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L151-L158).
- **Quale mapping è attivo**: iniettare `CmsComponentsService` in un componente di debug e chiamare `getMapping('<tipo>')` (core-libs/storefront/cms-structure/services/cms-components.service.ts:L165-L181); per un tipo di feature lazy farlo dopo `determineMappings` (JSDoc in core-libs/storefront/cms-structure/services/cms-components.service.ts:L155-L164).
- **Componente creato con il componente sbagliato**: il wrapper emette `ComponentCreateEvent` dopo la creazione (core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L103-L116); ascoltarlo con `EventService` (cap. 20).
- **Outlet**: iniettare `OutletService` e chiamare `get('<id>', OutletPosition.AFTER, true)` per vedere le registrazioni (core-libs/storefront/cms-structure/outlet/outlet.service.ts:L53-L67); se l'outlet è già a schermo forzare il ri-render con `OutletRendererService.render('<id>')` (core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L24-L38).
- **Route**: in dev mode "Could not configure the named route" (core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L169-L176) e "No path was configured for the named route" (core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L26-L34). Ispezionare `router.config` dopo il bootstrap per vedere il `path`/`matcher` effettivo e la posizione della route `'**'`.
- **PageContext**: con lo store devtools guardare lo stato del router (`context`, `cmsRequired`, `semanticRoute`) prodotto da `CustomSerializer` (core-libs/core/src/routing/store/reducers/router.reducer.ts:L159-

L166).

- **Layout**: in dev mode `logMissingLayoutConfig` stampa gli slot disponibili e "No layout config found for ..., you can configure a 'LayoutConfig'" (core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L251-L273). Nel DOM, il nome del template è una classe dell'host del layout e la `position` è una classe di ogni `cx-page-slot` (core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L114-L119, core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L44-L46).
- **Pagina statica non usata**: verificare che `pageId` coincida con l'id del `PageContext` e che nessun chunk successivo ridefinisca `cmsStructure.pages` (core-libs/core/src/cms/services/cms-structure-config.service.ts:L84-L92).

---

11. Domande di autoverifica

1. **Perché nell'app si usa `provideConfig` e non `provideDefaultConfig` per rimappare `CMSParagraphComponent`?** Perché i chunk di `provideConfig` finiscono in `RootConfig`, fuso sopra `DefaultConfig`, dove stanno i mapping di libreria; con `provideDefaultConfig` vincerebbe l'ordine di iniezione. — core-libs/core/src/config/config-tokens.ts:L10-L28, core-libs/core/src/config/config-providers.ts:L18-L27.
2. **Se rimappo solo `component` di un tipo che nella libreria ha `guards: [AuthGuard]`, il guard resta?** Sì: `deepMerge` fonde gli oggetti, quindi `guards` è ereditato; ridefinendo `guards` l'array verrebbe invece sostituito per intero. — core-libs/core/src/config/utils/deep-merge.ts:L36-L45.
3. **Come ottiene `CmsComponentData` una sottoclasse senza costruttore?** Angular usa la factory ereditata risalendo i prototipi, e `CmsComponentData` è fornito dall'injector del wrapper. — angular:packages/core/src/render3/di.ts:L918-L935, core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L35-L49.
4. **Che cosa emette per primo `data$` se il mapping ha `data`?** I soli dati statici (`startWith`), poi `{...static, ...cms}`. — core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L44-L51.
5. **Ho messo `provideOutlet` in un modulo caricato lazy e non compare nulla. Perché?** La registrazione di `forRoot` gira solo in `APP_INITIALIZER`; nei moduli lazy serve `OutletModule.forChild()` (MODULE_INITIALIZER). — core-libs/storefront/cms-structure/outlet/outlet.module.ts:L54-L88.
6. **Un `ng-template [cxOutletRef]="'X'"` senza `cxOutletPos` aggiunge o sostituisce?** Sostituisce: `OutletService.add` usa `REPLACE` come default, a differenza di `provideOutlet` (AFTER). — core-libs/storefront/cms-structure/outlet/outlet.service.ts:L31-L35, core-libs/storefront/cms-structure/outlet/outlet.module.ts:L42-L42.
7. **Perché un componente in outlet deve usare `context$` e non `context`?** `context` è deprecato e fissato alla creazione; i cambi di `cxOutletContext` arrivano solo su `outletContext$`. — core-libs/storefront/cms-structure/outlet/outlet.model.ts:L33-L43, core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L97-L99.
8. **Una route con `data: { cxRoute: 'faq', pageLabel: '/faq' }` e config `paths: ['faq', 'domande-frequenti']`: che `path` ha a runtime?** Nessuno: con più paths `configureRoute` cancella `path` e imposta un matcher costruito dai paths. — core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L98-L106.
9. **Che `PageContext` produce quella route, e con quale parametro la chiede l'adapter OCC?** `{ id: '/faq', type: ContentPage }`; l'adapter invia `pageLabelOrId=/faq`. — core-libs/core/src/routing/store/

reducers/router.reducer.ts:L193-L194, core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L95-L96.

10. **Ho aggiunto *FaqAside* a `layoutSlots` (template *FaqPageTemplate*) ma non appare. Perché?** `resolveSlots` mantiene solo gli slot presenti nella pagina CMS (o statica); se la pagina non ha *FaqAside*, viene filtrato. — core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L117-L119.

TAVOLA 51.1 **4.1 Dal mapping al componente (R1, R2, R3)**

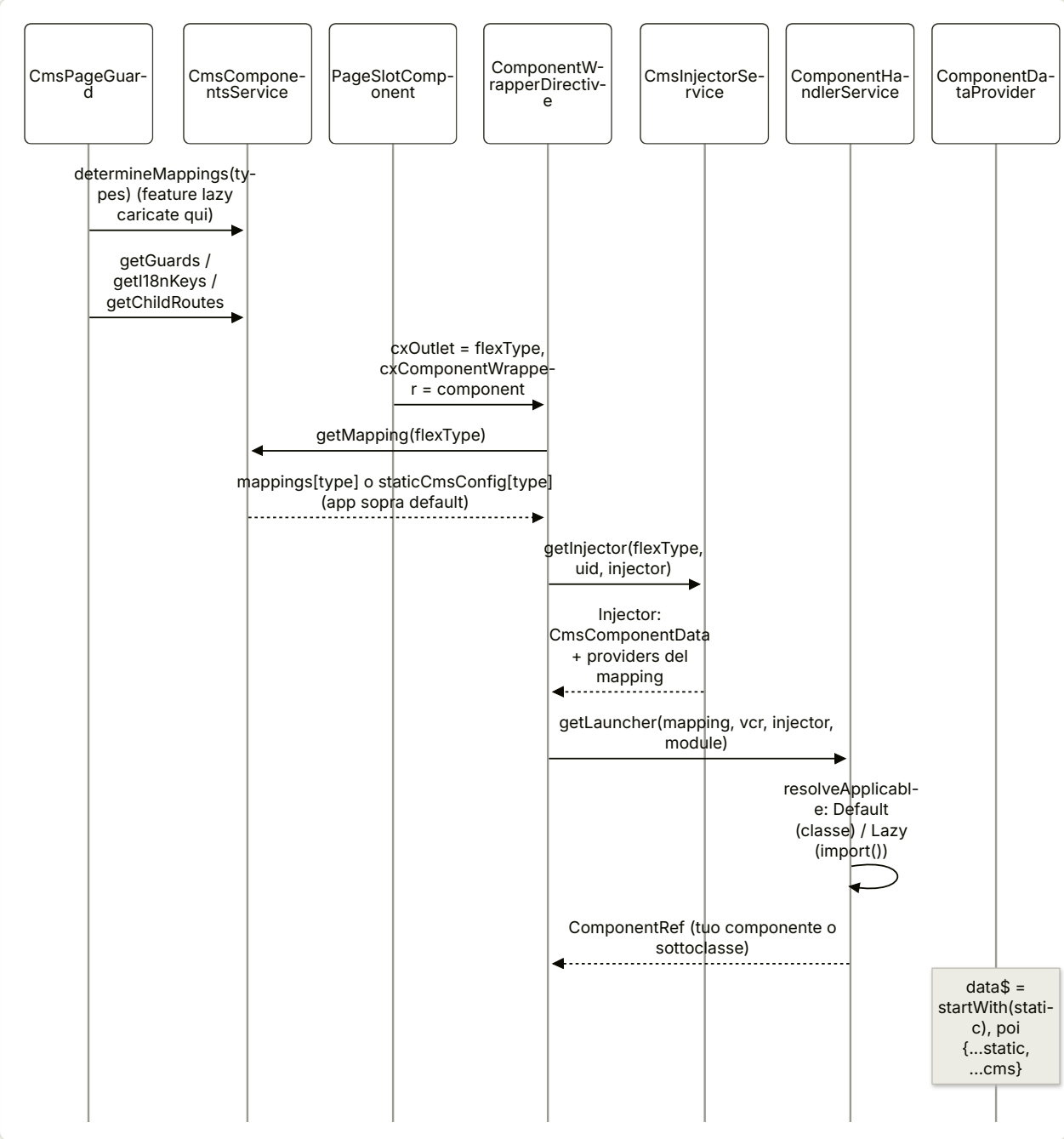


TAVOLA 51.2 **4.2 Registrazione e rendering di un outlet (R4, R5)**

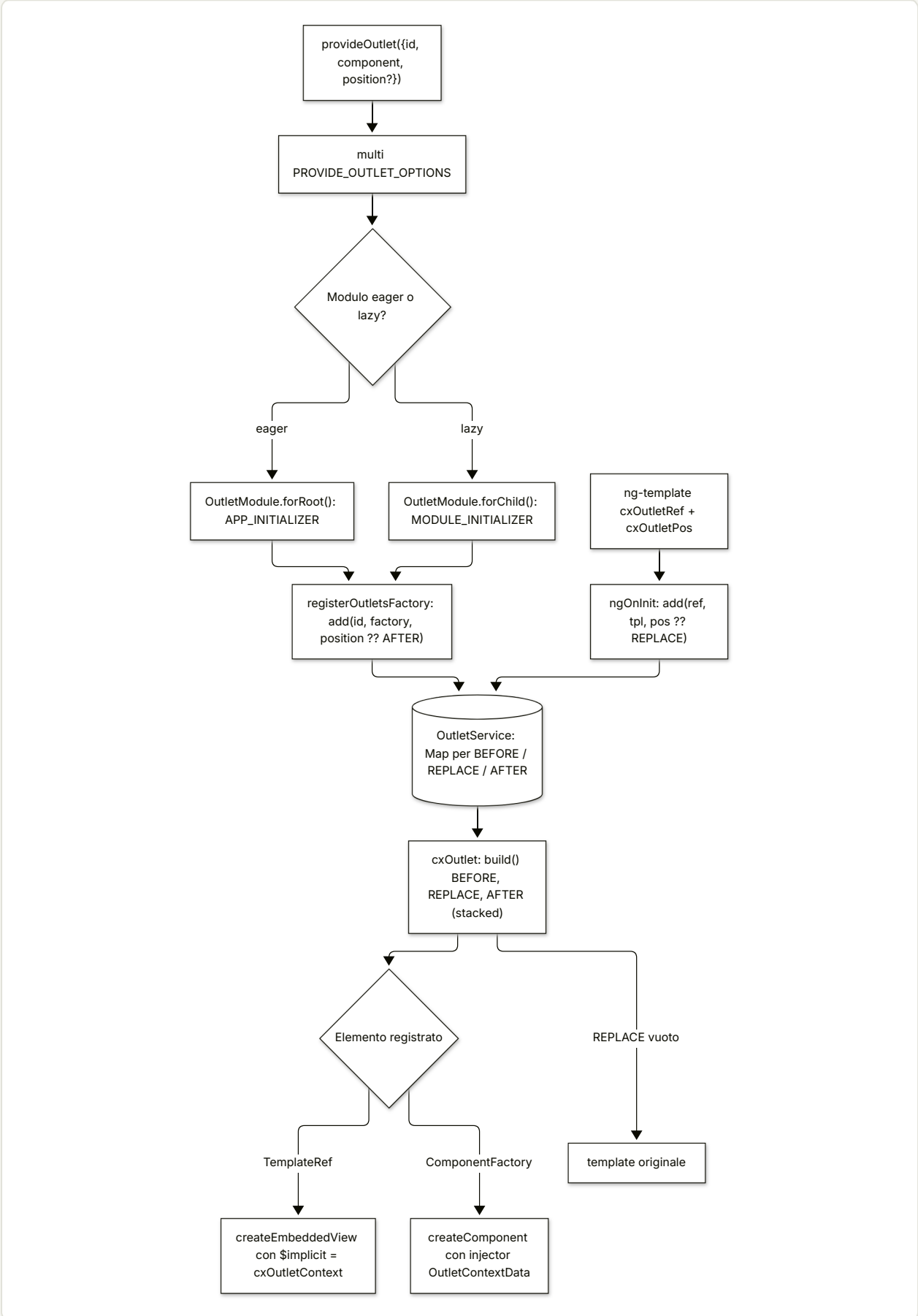
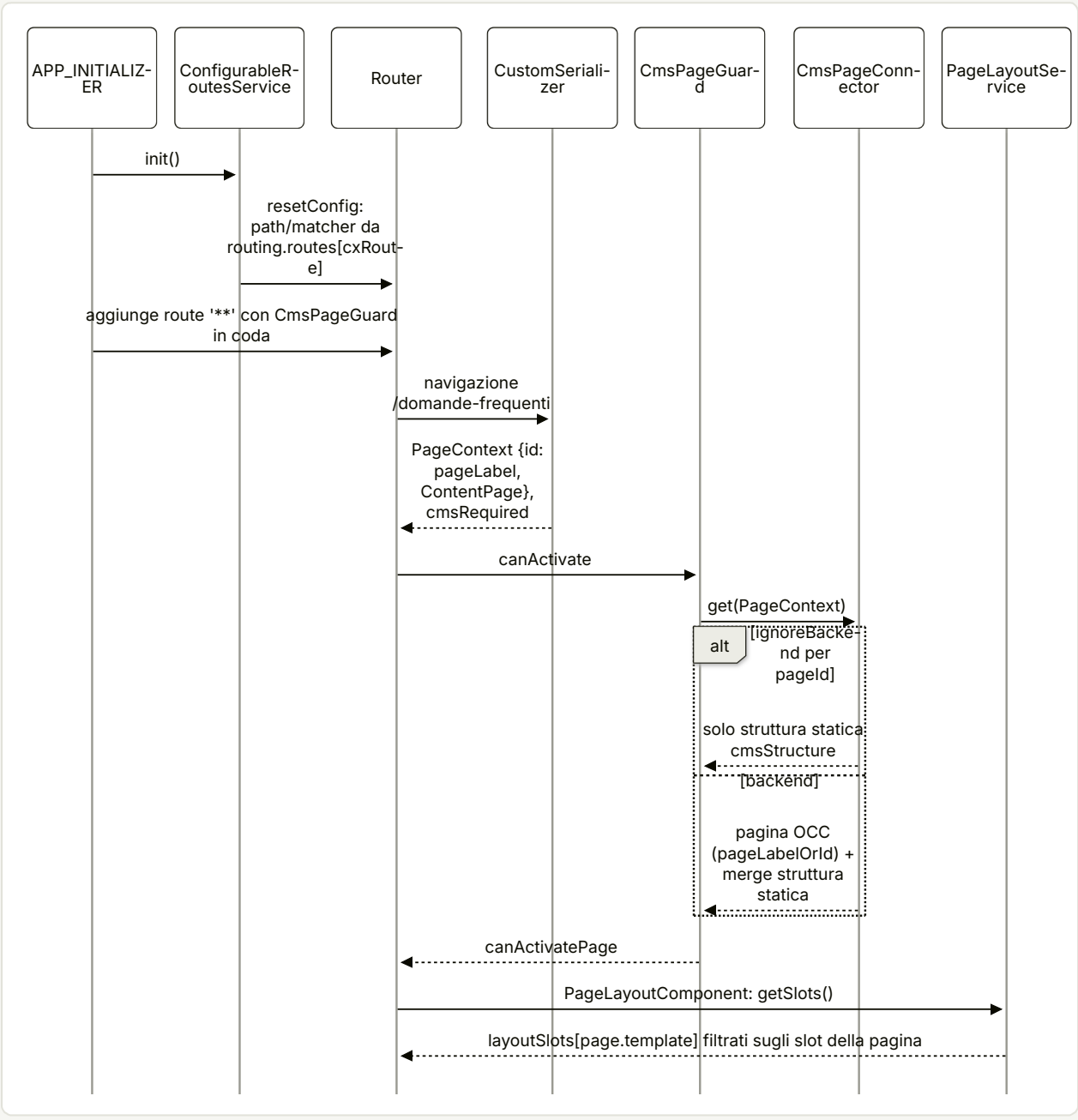


TAVOLA 51.3 **4.3 Route custom, pagina e layout (R6, R7)**



52

Ricette: OCC, adapter, converter, endpoint, meta resolver

Adapter e connector, normalizer e serializer, `fields` ed endpoint, `PageMetaResolver`.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Questo capitolo raccoglie le ricette per **cambiare i dati** che lo storefront scambia con il backend: da dove arrivano (adapter e connector), come vengono trasformati (normalizer e serializer), quali campi OCC si chiedono (`fields` ed endpoint per scope), quali nuovi endpoint si aggiungono (augmentation di `OccEndpoints` + `OccEndpointsService.buildUrl`) e come si producono i meta della pagina (`PageMetaResolver`).

Tutte le ricette poggiano su quattro meccanismi già descritti nella Parte II, che qui vengono dati per noti e richiamati solo dove servono:

1. **Adapter astratto come token DI.** Il dominio dipende da una classe astratta (`ProductAdapter`, `CartAdapter`, ...); il modulo OCC la lega all'implementazione con un semplice `useClass` (core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L51). Il JSDoc dell'adapter prodotto lo dice esplicitamente: i dati "can be loaded from alternative sources, as long as the structure converts to the `Product`" (core-libs/core/src/product/connectors/product/product.adapter.ts:L13-L15). Dettagli nel capitolo 21 — OCC.
2. **Converter impilabili su multi-provider.** `ConverterService` raccoglie tutti i converter registrati su un `InjectionToken` (root + moduli lazy) e li applica in catena con `reduce` (core-libs/core/src/util/converter.service.ts:L38-L65, core-libs/core/src/util/converter.service.ts:L130-L137).
3. **Endpoint come configurazione.** Gli URL sono template in `backend.occ.endpoints`, eventualmente per scope, risolti da `OccEndpointsService.buildUrl` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150). La mappa completa degli endpoint è nell'appendice A04 — Endpoint OCC.
4. **Selezione per punteggio dei meta resolver.** `PageMetaService` sceglie, tra tutte le istanze di `PageMetaResolver`, quella con lo score più alto per la pagina CMS corrente (core-libs/core/src/cms/facade/page-meta.service.ts:L53-L57, core-libs/core/src/util/applicable.ts:L47-L87). Dettagli nel capitolo 40 — SEO e PWA.

Il filo conduttore di tutto il capitolo è una domanda che ricorre in ogni ricetta: **in quale injector va registrata la personalizzazione, e in che ordine rispetto al default?** Il prodotto (`ProductOccModule`) vive nell'injector root (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L221-L221), il carrello e l'ordine in moduli lazy (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25, projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27): la stessa riga di codice funziona in un caso e viene ignorata nell'altro. I capitoli di dominio da tenere sotto mano sono 24 — Prodotto e ricerca, 25 — Carrello e 27 — Ordini.

> Convenzione: gli esempi di questo capitolo sono **illustrativi, non presenti nel repo**. Per ogni simbolo `@spartacus/*` importato è indicata la catena di export (`public_api.ts` → barrel → dichiarazione). I nomi inventati negli esempi (classi, token, campi) sono scritti in corsivo nel testo per distinguerli dai simboli reali.

2. Mappa dei file

PATH	RUOLO
core-libs/core/src/occ/config/occ-config.ts	`BackendConfig` (`occ.endpoints`, `loadingScopes`), `OccConfig` (`useExisting: Config`)

PATH	RUOLO
core-libs/core/src/occ/occ-models/occ-endpoints.model.ts	DEFAULT_SCOPE, OccEndpoint, OccEndpoints (augmentabile)
core-libs/core/src/occ/services/occ-endpoints.service.ts	OccEndpointsService.buildUrl, getEndpointForScope, isConfigured, getRawEndpointValue
core-libs/core/src/config/utils/string-template.ts	sostituzione dei placeholder ${...}
core-libs/core/src/occ/utils/occ-url-util.ts	urlPathJoin
core-libs/core/src/occ/services/occ-requests-optimizer.service.ts	fusione delle richieste per scope
core-libs/core/src/occ/services/occ-fields.service.ts, core-libs/core/src/occ/utils/occ-fields.ts	fusione/estrazione dei fields
core-libs/core/src/occ/config/loading-scopes-config.ts	LoadingScopeConfig, LoadingScopes
core-libs/core/src/util/converter.service.ts	Converter, ConverterService
core-libs/core/src/lazy-loading/unified-injector.ts	getMulti: istanze root + lazy
core-libs/core/src/lazy-loading/lazy-modules.service.ts	creazione dell'injector dei moduli lazy
core-libs/core/src/config/config-providers.ts, core-libs/core/src/config/config-tokens.ts, core-libs/core/src/config/utils/deep-merge.ts, core-libs/core/src/config/services/configuration.service.ts	provideConfig/provideDefaultConfig, merge e config dei moduli lazy
core-libs/core/src/product/connectors/product/product.adapter.ts, product.connector.ts, converters.ts	adapter/connector/token prodotto
core-libs/core/src/occ/adapters/product/product-occ.module.ts	binding ProductAdapter→OccProductAdapter + normalizer multi
core-libs/core/src/occ/adapters/product/occ-product.adapter.ts	OccProductAdapter (estendibile)
core-libs/core/src/occ/adapters/product/default-occ-product-config.ts	endpoint product per scope + loadingScopes
core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts	normalizer di riferimento (target ?? {...source})
core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts	uso di ADDRESS_SERIALIZER
core-libs/core/src/product/facade/product.service.ts, core-libs/core/src/product/services/product-loading.service.ts, core-libs/core/src/product/store/effects/product.effect.ts	scope e caricamento prodotto
core-libs/core/src/cms/page/page-meta.resolver.ts, page.resolvers.ts, base-page-meta.resolver.ts	meta resolver: base, interfacce, helper
core-libs/core/src/cms/facade/page-meta.service.ts	selezione e risoluzione dei meta
core-libs/core/src/cms/page/config/default-page-meta.config.ts, page-meta.config.ts	pageMeta.resolvers
core-libs/core/src/util/applicable.ts	resolveApplicable (score, parità)
core-libs/storefront/cms-structure/seo/seo-meta.service.ts	scrittura dei meta nel DOM
feature-libs/cart/base/core/connectors/cart/cart.adapter.ts, cart.connector.ts	adapter/connector cart
feature-libs/cart/base/core/cart-base-core.module.ts	ri-fornisce i connector nel modulo lazy

PATH	RUOLO
feature-libs/cart/base/occ/cart-base-occ.module.ts	binding OCC del cart + CART_NORMALIZER
feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts	OccCartAdapter
feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts	normalizer cart (applica PRODUCT_NORMALIZER)
feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts	CartOccEndpoints + augmentation
feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts	fields di default di carts/cart/createCart
integration-libs/omf/order/*	esempio reale: adapter OCC esteso e sostituito (lazy)
integration-libs/opf/gift-card/{root,core,occ}/*	esempio reale: normalizer, campi extra, endpoint, adapter/connector nuovi
feature-libs/product/bulk-pricing/*	esempio reale: scope prodotto custom
feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts, feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts	esempi reali di PageMetaResolver

---

3. Come funziona passo per passo

Le dieci ricette seguono lo stesso schema. Prima, però, tre regole trasversali che spiegano quasi tutte le trappole.

Regola A — precedenza della config. La config globale è `deepMerge({}, DefaultConfig, RootConfig)`: ciò che l'app passa a `provideConfig` vince sempre su `provideDefaultConfig` delle librerie (core-libs/core/src/config/config-tokens.ts:L10-L21, core-libs/core/src/config/config-providers.ts:L11-L27). Quando si carica un modulo lazy la config viene ricalcolata con precedenza `defaultConfig < ambientDefaultConfig < ambientConfig < rootConfig` e fusa *in place* nell'oggetto `Config` globale, salvo il flag `disableConfigUpdates` (core-libs/core/src/config/services/configuration.service.ts:L92-L105). `deepMerge` fonde solo gli oggetti: **stringhe e array sostituiscono** il valore precedente (core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/config/utils/deep-merge.ts:L31-L46).

Regola B — precedenza dei provider singoli. Dentro un injector, per un token non-multi vince l'ultimo provider registrato (angular:packages/core/src/di/r3_injector.ts:L497-L521); in un NgModule Angular processa prima gli `imports` (i `ModuleWithProviders` dopo tutti gli import) e poi i `providers` del modulo stesso (angular:packages/core/src/di/provider_collection.ts:L293-L322, angular:packages/core/src/di/provider_collection.ts:L346-L348). Un modulo lazy ha il suo injector (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L97), i cui provider oscurano quelli della root per chi viene istanziato lì dentro.

Regola C — ordine dei multi-provider. `UnifiedInjector.getMulti` legge il token prima dalla root e poi da ogni injector lazy (con `{ self: true }`) nell'ordine di caricamento, e concatena con `scan` (core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68, core-libs/core/src/lazy-loading/unified-injector.ts:L75-L93). È così che vengono raccolti sia i converter (core-libs/core/src/util/converter.service.ts:L52-L65) sia i `PageMetaResolver` (core-libs/core/src/cms/facade/page-meta.service.ts:L53-L57).

3.1 Ricetta: estendere e sostituire l'adapter OCC di una feature eager (`ProductAdapter`)

Quando usarlo

Quando una parte dei dati prodotto deve venire da una sorgente diversa da OCC (un PIM esterno, un mock, un microservizio) oppure quando la chiamata OCC va arricchita (header, logica di fallback) senza toccare facade, store ed effect. È la situazione descritta dal JSDoc di `ProductAdapter` (core-libs/core/src/product/connectors/product/product.adapter.ts:L13-L15).

Esempio

Esempio illustrativo, non presente nel repo. I prodotti con codice che inizia per `EXT-` vengono letti da un'API esterna; tutti gli altri passano per il percorso OCC standard (inclusa la fusione dei `fields`).

```ts
// app/product/external-aware-product.adapter.ts  (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';
import {
  OccProductAdapter,
  Product,
  ProductAdapter,
  ScopedProductData,
} from '@spartacus/core';

interface ExternalProduct {
  title: string;
  summary?: string;
}

const isExternal = (code: string) => code.startsWith('EXT-');

@Injectable()
export class ExternalAwareProductAdapter extends OccProductAdapter {
  // Percorso usato dagli effect: ProductConnector.getMany -> adapter.loadMany
  loadMany(products: ScopedProductData[]): ScopedProductData[] {
    const occProducts = products.filter((p) => !isExternal(p.code));
    const externalProducts = products.filter((p) => isExternal(p.code));
    return [
      ...super.loadMany(occProducts),
      ...externalProducts.map((p) => ({
        ...p,
        data$: this.load(p.code, p.scope),
      })),
    ];
  }

  load(productCode: string, scope?: string): Observable<Product> {
    if (!isExternal(productCode)) {
      return super.load(productCode, scope);
    }
```

```
    return this.http
      .get<ExternalProduct>(
        `https://pim.example.com/api/products/${encodeURIComponent(productCode)}`
      )
      .pipe(
        map((ext) => ({
          code: productCode,
          name: ext.title,
          summary: ext.summary,
        }))
      );
  }
}

@NgModule({
  providers: [{ provide: ProductAdapter, useClass: ExternalAwareProductAdapter }],
})
export class ExternalProductModule {}
// Importare ExternalProductModule DOPO il modulo che porta ProductOccModule,
// oppure mettere il provider direttamente nei `providers` di AppModule.
```

Catene di export dei simboli `@spartacus/core`:

SIMBOLO	CATENA
OccProductAdapter	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L7-L7 → core-libs/core/src/occ/adapters/index.ts:L9-L9 → core-libs/core/src/occ/adapters/product/index.ts:L11-L11 → core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L21-L21
ProductAdapter	core-libs/core/public_api.ts:L29-L29 → core-libs/core/src/product/index.ts:L7-L7 → core-libs/core/src/product/connectors/index.ts:L7-L7 → core-libs/core/src/product/connectors/product/index.ts:L8-L8 → core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L11
ScopedProductData	core-libs/core/public_api.ts:L29-L29 → core-libs/core/src/product/index.ts:L7-L7 → core-libs/core/src/product/connectors/index.ts:L7-L7 → core-libs/core/src/product/connectors/product/index.ts:L10-L10 → core-libs/core/src/product/connectors/product/scoped-product-data.ts:L13-L13
Product	core-libs/core/public_api.ts:L49-L49 (AUGMENTABLE_TYPES) → core-libs/core/src/model/product.model.ts:L167-L167

Perché funziona (file sorgente)

- `ProductOccModule` lega il token astratto con `useClass: OccProductAdapter` (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L37) ed è importato nell'injector root della demo (projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L15-L15, projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L221-L221): un provider successivo dello stesso token lo sostituisce (Regola B).
- `OccProductAdapter` è `@Injectable()` con dipendenze `protected` (`http`, `occEndpoints`, `converter`, `requestsOptimizer`) e `getEndpoint` `protected`: è pensato per essere esteso (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L20-L27, core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L54-L59).
- Gli effect raggruppano le `LoadProduct` simultanee e chiamano `productConnector.getMany` (core-libs/core/src/product/store/effects/product.effect.ts:L45-L53); il connector usa `loadMany` se l'adapter lo imple-

menta, altrimenti `load` per ciascun prodotto (core-libs/core/src/product/connectors/product/product.connector.ts:L23-L32). Per questo l'esempio sovrascrive **anche** `loadMany`.

- Per ogni elemento restituito l'effect crea `LoadProductSuccess({ code: productLoad.code, ...data }, productLoad.scope)`: l'ordine dell'array non conta, conta che ogni elemento conservi `code` e `scope` (core-libs/core/src/product/store/effects/product.effect.ts:L59-L90).
- Lo stesso schema "estendi l'adapter OCC e sostituisci il token" è usato nel repo da CDC per `UserConsentAdapter` (integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L16-L31, integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L20-L45) e da OPF per `UserPaymentAdapter` (integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L33-L40).

Limiti e trappole

- **Sovrascrivere solo `load` non basta**: `OccProductAdapter` implementa `loadMany`, quindi il percorso degli effect non passa mai da `load` (core-libs/core/src/product/connectors/product/product.connector.ts:L23-L32).
- **Ordine dei provider**: un modulo di override importato *prima* di quello che porta `ProductOccModule` perde (Regola B; angular:packages/core/src/di/provider_collection.ts:L293-L348).
- **Nessun normalizer sul ramo esterno**: l'esempio produce direttamente un `Product`; se si vuole riusare la catena di `PRODUCT_NORMALIZER` bisogna passare per `this.converter.pipeable(PRODUCT_NORMALIZER)` come fa `load` (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L29-L33), ricordando che i normalizer registrati si aspettano la forma `Occ.Product`.
- **Una richiesta per scope** sul ramo esterno: la fusione dei `fields` è fatta da `OccRequestsOptimizerService.scopedDataLoad`, che qui non viene usato (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L35-L52).
- `OccProductAdapter` non è `providedIn`: la sottoclasse va fornita con `useClass` (o dichiarata `providedIn: 'root'`, come fa `CdcUserConsentAdapter`) (integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L20-L20).

3.2 Ricetta: sostituire l'adapter di una feature lib lazy (`CartAdapter`)

Quando usarlo

Quando l'adapter da cambiare appartiene a una feature lib caricata in lazy (cart, order, checkout, ...). Il caso di riferimento nel repo è OMF, che sostituisce `OrderHistoryAdapter` con una sottoclasse di `OccOrderHistoryAdapter` (integration-libs/omf/order/omf-order.module.ts:L15-L26, integration-libs/omf/order/occ-omf-order-history.adapter.ts:L23-L30).

Esempio

Esempio illustrativo, non presente nel repo. `OccCartAdapter.load(userId, 'current')` non chiama l'endpoint `cart` ma scarica **tutti** i carrelli con `carts` e prende quello senza `saveTime` (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L50). L'adapter qui sotto, per `current`, chiama direttamente `cart` con `cartId = 'current'`.

```ts
// app/cart/single-call-cart.adapter.ts  (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { Observable } from 'rxjs';
import { CartBaseModule } from '@spartacus/cart/base';
import { CartAdapter } from '@spartacus/cart/base/core';
```

```
import { OccCartAdapter } from '@spartacus/cart/base/occ';
import { Cart, CART_NORMALIZER } from '@spartacus/cart/base/root';
import { Occ, OCC_CART_ID_CURRENT } from '@spartacus/core';

@Injectable()
export class SingleCallCartAdapter extends OccCartAdapter {
  public load(userId: string, cartId: string): Observable<Cart | undefined> {
    if (cartId !== OCC_CART_ID_CURRENT) {
      return super.load(userId, cartId);
    }
    return this.http
      .get<Occ.Cart>(
        this.occEndpointsService.buildUrl('cart', {
          urlParams: { userId, cartId },
        })
      )
      .pipe(this.converterService.pipeable(CART_NORMALIZER));
  }
}

// Wrapper caricato in lazy al posto di CartBaseModule:
// featureModules: { cartBase: { module: () => import('./cart-base-wrapper.module').then(m =>
m.CartBaseWrapperModule) } }
@NgModule({
  imports: [CartBaseModule],
  providers: [{ provide: CartAdapter, useClass: SingleCallCartAdapter }],
})
export class CartBaseWrapperModule {}
```

Catene di export:

SIMBOLO	CATENA
CartBaseModule	feature-libs/cart/base/public_api.ts:L7-L7 → feature-libs/cart/base/cart-base.module.ts:L15-L15
CartAdapter	feature-libs/cart/base/core/public_api.ts:L9-L9 → feature-libs/cart/base/core/connectors/index.ts:L8-L8 → feature-libs/cart/base/core/connectors/cart/index.ts:L7-L7 → feature-libs/cart/base/core/connectors/cart/cart.adapter.ts:L10-L10
OccCartAdapter	feature-libs/cart/base/occ/public_api.ts:L7-L7 → feature-libs/cart/base/occ/adapters/index.ts:L13-L13 → feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L28-L28
Cart	feature-libs/cart/base/root/public_api.ts:L19-L19 (AUGMENTABLE_TYPES) → feature-libs/cart/base/root/models/cart.model.ts:L62-L62
CART_NORMALIZER	feature-libs/cart/base/root/public_api.ts:L15-L15 → feature-libs/cart/base/root/tokens/index.ts:L8-L8 → feature-libs/cart/base/root/tokens/converters.ts:L17-L17
Occ	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L15-L15 → core-libs/core/src/occ/occ-models/index.ts:L8-L8 → core-libs/core/src/occ/occ-models/occ.models.ts:L7-L7
OCC_CART_ID_CURRENT	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L17-L17 → core-libs/core/src/occ/utils/index.ts:L14-L14 → core-libs/core/src/occ/utils/occ-constants.ts:L7-L7

Perché funziona (file sorgente)

- `CartBaseModule` = `CartBaseCoreModule` + `CartBaseOccModule` + `CartBaseComponentsModule` (feature-libs/cart/base/cart-base.module.ts:L12-L15); `CartBaseOccModule` lega `CartAdapter`→`OccCartAdapter` con `useClass` (feature-libs/cart/base/occ/cart-base-occ.module.ts:L32-L39).
- `CartBaseCoreModule` **ri-fornisce** `CartConnector` nei propri `providers` (feature-libs/cart/base/core/cart-base-core.module.ts:L22-L36): il connector viene istanziato nell'injector lazy e risolve il `CartAdapter` di quell'injector. Nel wrapper, i `providers` del modulo sono processati dopo gli `imports` (Regola B), quindi vince *SingleCallCartAdapter*.
- `OccCartAdapter` espone `http`, `occEndpointsService`, `converterService` come `protected` (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L27-L33).
- La demo carica il cart tramite un wrapper lazy (projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L26-L35, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25), esattamente come OMF si aggancia a `OrderWrapperModule` dopo `OrderModule` (projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27).

Limiti e trappole

- **Override nel posto sbagliato**: un `{ provide: CartAdapter, useClass: … }` nell'`AppModule` (root) viene oscurato dal provider dell'injector lazy, perché sia l'adapter sia il connector vivono lì (feature-libs/cart/base/occ/cart-base-occ.module.ts:L36-L39, feature-libs/cart/base/core/cart-base-core.module.ts:L29-L30, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L97).
- Il core del cart è bundlato con i componenti (`[CART_BASE_CORE_FEATURE]: CART_BASE_FEATURE`), quindi c'è un solo modulo lazy da wrappare (feature-libs/cart/base/root/cart-base-root.module.ts:L45-L53).
- DA VERIFICARE: la semantica lato backend di `users/{userId}/carts/current` (che carrello restituisce, e cosa succede se non esiste) non è verificabile nel repo; l'adapter di default filtra `saveTime === undefined` sulla lista (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L50), comportamento che un override deve riprodurre o accettare di cambiare.
- `OccCartAdapter` non è `providedIn` (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L27-L27): serve `useClass`.

3.3 Ricetta: nuovo dominio con adapter astratto, connector e implementazione OCC

Quando usarlo

Quando si introducono dati che Spartacus non conosce (punti fedeltà, garanzie, documenti) e si vuole lo stesso disaccoppiamento delle feature lib: il dominio dipende da un adapter astratto, il connector lo delega, il modulo OCC lo implementa. Modello reale: OPF gift card (integration-libs/opf/gift-card/core/connectors/opf-gift-card.adapter.ts:L14-L32, integration-libs/opf/gift-card/core/connectors/opf-gift-card.connector.ts:L16-L40, integration-libs/opf/gift-card/occ/opf-gift-card-occ.module.ts:L12-L21).

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/loyalty/loyalty.ts  (esempio illustrativo, non presente nel repo)
import { HttpClient } from '@angular/common/http';
import { inject, Injectable, InjectionToken, NgModule } from '@angular/core';
import { Observable } from 'rxjs';
```

```
import {
  Converter,
  ConverterService,
  OccConfig,
  OccEndpoint,
  OccEndpointsService,
  provideDefaultConfig,
} from '@spartacus/core';

// 1. Modello UI
export interface LoyaltyBalance {
  points: number;
  tier?: string;
}

// 1b. Risposta OCC grezza dell'endpoint custom (DTO dedicato)
export interface OccLoyaltyBalance {
  loyaltyPoints?: number;
  loyaltyTier?: string;
}

// 2. Endpoint: augmentation di OccEndpoints (tipizza la config)
export interface LoyaltyOccEndpoints {
  loyaltyBalance?: string | OccEndpoint;
}
declare module '@spartacus/core' {
  interface OccEndpoints extends LoyaltyOccEndpoints {}
}

// 3. Token del normalizer (locale, come CDC_PREFERENCE_SERIALIZER in CDC)
export const LOYALTY_BALANCE_NORMALIZER = new InjectionToken<
  Converter<any, LoyaltyBalance>
>('LoyaltyBalanceNormalizer');

// 4. Adapter astratto (token DI) e connector
export abstract class LoyaltyAdapter {
  abstract loadBalance(userId: string): Observable<LoyaltyBalance>;
}

@Injectable({ providedIn: 'root' })
export class LoyaltyConnector {
  protected adapter = inject(LoyaltyAdapter);
  getBalance(userId: string): Observable<LoyaltyBalance> {
    return this.adapter.loadBalance(userId);
  }
}

// 5. Implementazione OCC
@Injectable()
```

```
export class OccLoyaltyAdapter implements LoyaltyAdapter {
  protected http = inject(HttpClient);
  protected occEndpoints = inject(OccEndpointsService);
  protected converter = inject(ConverterService);

  loadBalance(userId: string): Observable<LoyaltyBalance> {
    return this.http
      .get<OccLoyaltyBalance>(this.getBalanceEndpoint(userId))
      .pipe(this.converter.pipeable(LOYALTY_BALANCE_NORMALIZER));
  }

  protected getBalanceEndpoint(userId: string): string {
    return this.occEndpoints.buildUrl('loyaltyBalance', {
      urlParams: { userId },
    });
  }
}

@Injectable({ providedIn: 'root' })
export class OccLoyaltyBalanceNormalizer
  implements Converter<any, LoyaltyBalance>
{
  convert(source: any, target?: LoyaltyBalance): LoyaltyBalance {
    target ??= { points: 0 };
    target.points = Number(source?.loyaltyPoints ?? 0);
    target.tier = source?.loyaltyTier;
    return target;
  }
}

// 6. Modulo che lega tutto
export const defaultLoyaltyOccConfig: OccConfig = {
  backend: {
    occ: {
      endpoints: {
        loyaltyBalance: 'users/${userId}/loyalty?fields=loyaltyPoints,loyaltyTier',
      },
    },
  },
};

@NgModule({
  providers: [
    provideDefaultConfig(defaultLoyaltyOccConfig),
    { provide: LoyaltyAdapter, useClass: OccLoyaltyAdapter },
    {
      provide: LOYALTY_BALANCE_NORMALIZER,
      useExisting: OccLoyaltyBalanceNormalizer,
      multi: true,
```

```
    },
  ],
})
export class LoyaltyOccModule {}
```

Catene di export dei simboli `@spartacus/core`:

SIMBOLO	CATENA
`Converter`, `ConverterService`	core-libs/core/public_api.ts:L35-L35 → core-libs/core/src/util/index.ts:L11-L11 → core-libs/core/src/util/converter.service.ts:L21-L21, core-libs/core/src/util/converter.service.ts:L35-L35
`Occ`	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L15-L15 → core-libs/core/src/occ/occ-models/index.ts:L8-L8 → core-libs/core/src/occ/occ-models/occ.models.ts:L7-L7
`OccConfig`	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L12-L12 → core-libs/core/src/occ/config/occ-config.ts:L46-L46
`OccEndpoint`	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L15-L15 → core-libs/core/src/occ/occ-models/index.ts:L7-L7 → core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L9-L9
`OccEndpoints` (augmentato)	core-libs/core/public_api.ts:L51-L51 (AUGMENTABLE_TYPES) → core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L21
`OccEndpointsService`	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L16-L16 → core-libs/core/src/occ/services/index.ts:L7-L7 → core-libs/core/src/occ/services/occ-endpoints.service.ts:L34-L34
`provideDefaultConfig`	core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L55-L55

Perché funziona (file sorgente)

- Il connector OPF è `providedIn: 'root'`, usa `inject(OpfGiftCardAdapter)` e delega 1:1 (integration-libs/opf/gift-card/core/connectors/opf-gift-card.connector.ts:L16-L40); il modulo OCC lega l'adapter e fornisce gli endpoint con `provideDefaultConfig` (integration-libs/opf/gift-card/occ/opf-gift-card-occ.module.ts:L12-L21).
- L'augmentation `declare module '@spartacus/core' { interface OccEndpoints extends X {} }` è lo schema di tutte le feature lib (feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L97-L99, integration-libs/opf/gift-card/occ/model/occ-opf-gift-card-endpoints.model.ts:L9-L22); `OccEndpoints` è nel blocco AUGMENTABLE_TYPES, che il builder sposta nel `.d.ts` radice proprio per renderlo augmentabile (core-libs/core/public_api.ts:L38-L58, tools/build-lib/augmented-types/index.js:L42-L52).
- Il token del normalizer può essere locale: CDC definisce `CDC_PREFERENCE_SERIALIZER` nel proprio pacchetto e lo registra multi (integration-libs/cdc/root/consent-management/converters/converter.ts:L11-L13, integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L24-L28).
- `ConverterService.pipeable` restituisce la sorgente invariata se il token non ha converter (core-libs/core/src/util/converter.service.ts:L80-L128).

Limiti e trappole

- **Se la feature è lazy, il connector va ri-fornito** nel modulo lazy, come fanno `CartBaseCoreModule` e `OpfGiftCardCoreModule` (feature-libs/cart/base/core/cart-base-core.module.ts:L22-L36, integration-

libs/opf/gift-card/core/opf-gift-card-core.module.ts:L11-L14); altrimenti il connector `providedIn: 'root'` viene istanziato nella root e non trova l'adapter registrato solo nell'injector lazy.

- **Il primo normalizer crea il target**: `reduce` parte da `undefined` (core-libs/core/src/util/converter.service.ts:L130-L137); per questo l'esempio fa `target ??= …`.
- **L'augmentation va importata**: un file che contiene solo `declare module` non è attivo finché non viene importato da qualche parte; il file B2B lo commenta esplicitamente e il core importa `./product-occ-config` per side effect (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L7-L17, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L26-L26).
- Gli errori HTTP non sono normalizzati automaticamente: l'adapter OPF li passa a `tryNormalizeHttpError` e applica `backOff({ shouldRetry: isJaloError })` (integration-libs/opf/gift-card/occ/adapters/occ-opf-gift-card.adapter.ts:L35-L66).

3.4 Ricetta: normalizer custom per un modello eager (`PRODUCT_NORMALIZER`)

Quando usarlo

Quando il backend restituisce un campo che il modello UI non ha (o lo restituisce in una forma diversa) e lo si vuole esporre a tutti i consumer di `Product` senza toccare l'adapter. Il normalizer viene applicato a ogni risposta prodotto, e anche ai prodotti dentro carrello, ordini, resi e quote.

Esempio

Esempio illustrativo, non presente nel repo. Il backend espone `energyClass` (campo custom) che diventa `energyLabel` in `Product`.

```ts
// app/product/energy-label.normalizer.ts  (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { Converter, Occ, Product, PRODUCT_NORMALIZER } from '@spartacus/core';

declare module '@spartacus/core' {
  interface Product {
    energyLabel?: string;
  }
}

@Injectable({ providedIn: 'root' })
export class EnergyLabelNormalizer implements Converter<Occ.Product, Product> {
  convert(source: Occ.Product, target?: Product): Product {
    target = target ?? { ...(source as unknown as Partial<Product>) };
    const energyClass = (source as any).energyClass;
    if (energyClass) {
      target.energyLabel = String(energyClass).toUpperCase();
    }
    return target as Product;
  }
}

@NgModule({
  providers: [
    {
```

```
      provide: PRODUCT_NORMALIZER,
      useExisting: EnergyLabelNormalizer,
      multi: true,
    },
  ],
})
export class EnergyLabelModule {}
// Il campo deve anche essere richiesto nei `fields` dell'endpoint: vedi 3.7 e 3.8.
```

SIMBOLO	CATENA
Converter	core-libs/core/public_api.ts:L35-L35 → core-libs/core/src/util/index.ts:L11-L11 → core-libs/core/src/util/converter.service.ts:L21-L21
Occ	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L15-L15 → core-libs/core/src/occ/occ-models/index.ts:L8-L8 → core-libs/core/src/occ/occ-models/occ.models.ts:L7-L7
Product (augmentato)	core-libs/core/public_api.ts:L49-L49 (AUGMENTABLE_TYPES) → core-libs/core/src/model/product.model.ts:L167-L167
PRODUCT_NORMALIZER	core-libs/core/public_api.ts:L29-L29 → core-libs/core/src/product/index.ts:L7-L7 → core-libs/core/src/product/connectors/index.ts:L7-L7 → core-libs/core/src/product/connectors/product/index.ts:L9-L9 → core-libs/core/src/product/connectors/product/converters.ts:L11-L11

Perché funziona (file sorgente)

- I normalizer di default sono `providedIn: 'root'` e registrati multi con `useExisting` in `ProductOccModule` (core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51, core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L14-L26).
- La convenzione `target ?? { ...source }` è quella di `ProductNameNormalizer` (core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts:L13-L26): se il normalizer è il primo della catena crea il target, altrimenti arricchisce quello ricevuto (core-libs/core/src/util/converter.service.ts:L130-L137).
- `ConverterService` legge la catena con `getLastValueSync(unifiedInjector.getMulti(token))` e la mette in cache, svuotandola quando compare un nuovo injector lazy (core-libs/core/src/util/converter.service.ts:L38-L65, core-libs/core/src/util/rxjs/get-last-value-sync.ts:L19-L23).
- `Product` è nel blocco AUGMENTABLE_TYPES del core (core-libs/core/public_api.ts:L49-L49); lo stesso schema di augmentation è usato da OPF per `Cart` (integration-libs/opf/gift-card/root/model/augmented-opf-gift-card.model.ts:L12-L23).
- `OccCartNormalizer` applica `PRODUCT_NORMALIZER` al `product` di ogni entry (feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts:L23-L43); lo fanno anche `OccOrderNormalizer`, `OccReplenishmentOrderNormalizer`, `OccReturnRequestNormalizer` e `OccQuoteEntryNormalizer` (feature-libs/order/occ/adapters/converters/occ-order-normalizer.ts:L74-L74, feature-libs/order/occ/adapters/converters/occ-replenishment-order-normalizer.ts:L34-L34, feature-libs/order/occ/adapters/converters/occ-return-request-normalizer.ts:L41-L41, feature-libs/quote/occ/converters/occ-quote-entry-normalizer.ts:L30-L30).

Limiti e trappole

- `multi: true` **dimenticato**: non sostituisce la catena "in silenzio". Nello stesso injector Angular lancia l'errore di provider misti in dev mode, sia per un multi dopo un singolo sia per un singolo dopo i multi (il caso tipico di questa ricetta); tra injector diversi `getMulti` lancia `Multi-providers mixed with single providers` (angular:packages/core/src/di/r3_injector.ts:L500-L505, angular:packages/core/src/di/r3_injector.

ts:L513-L519, core-libs/core/src/lazy-loading/unified-injector.ts:L84-L88). La skill di sviluppo nel repo afferma il contrario (core-libs/skills/skills/spartacus-developer/references/normalizers.md:L5-L7): fa fede il codice.

- **Default param sul target**: un `convert(source, target: Product = {})` in prima posizione perderebbe tutti i campi della sorgente (core-libs/core/src/util/converter.service.ts:L134-L136).
- **Prodotti parziali**: dentro carrello/ordini il prodotto arriva con i `fields` di quegli endpoint; il normalizer deve tollerare campi assenti (feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts:L31-L39).
- **Campi non tipizzati di** `Occ.*`: il namespace `Occ` non è nel blocco AUGMENTABLE_TYPES; gli esempi reali leggono i campi extra con `(source as any).campo` (integration-libs/opf/gift-card/root/normalizers/opf-gift-card-cart-occ-normalizer.ts:L26-L26, core-libs/core/public_api.ts:L38-L58).
- Per **sostituire** un normalizer di default invece di aggiungerne uno, si può ridefinire la classe concreta referenziata da `useExisting` (es. `{ provide: ProductNameNormalizer, useClass: … }`), perché `useExisting` diventa `ɵɵinject(classe)` (angular:packages/core/src/di/r3_injector.ts:L657-L664). DA VERIFICARE: nessun esempio nel repo, dedotto dal codice DI.

3.5 Ricetta: normalizer che lavora sull'output del default in una feature lazy (`CART_NORMALIZER`)

Quando usarlo

Quando si vuole aggiungere al `Cart` un campo calcolato a partire dal risultato di `OccCartNormalizer` (per esempio dalle entry già normalizzate), oppure in generale quando il proprio normalizer deve girare **dopo** quello della feature lib.

Esempio

Esempio illustrativo, non presente nel repo.

```ts
// app/cart/restricted-items.normalizer.ts  (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { CartBaseModule } from '@spartacus/cart/base';
import { Cart, CART_NORMALIZER } from '@spartacus/cart/base/root';
import { Converter, Occ } from '@spartacus/core';

declare module '@spartacus/cart/base/root' {
  interface Cart {
    hasRestrictedItems?: boolean;
  }
}

@Injectable({ providedIn: 'root' })
export class RestrictedItemsCartNormalizer implements Converter<Occ.Cart, Cart> {
  convert(source: Occ.Cart, target?: Cart): Cart {
    // Registrato dopo CartBaseModule: target è l'output di OccCartNormalizer
    target ??= { ...(source as any) } as Cart;
    // `restricted` va aggiunto al `product(...)` delle entry nei `fields`
    // di `carts` e `cart` (vedi 3.7), altrimenti il risultato è sempre false
    target.hasRestrictedItems = (target.entries ?? []).some(
```

```
      (entry) => (entry.product as any)?.restricted === true
    );
    return target;
  }
}

@NgModule({
  imports: [CartBaseModule],
  providers: [
    {
      provide: CART_NORMALIZER,
      useExisting: RestrictedItemsCartNormalizer,
      multi: true,
    },
  ],
})
export class CartBaseWrapperModule {}
```

SIMBOLO	CATENA
CartBaseModule	feature-libs/cart/base/public_api.ts:L7-L7 → feature-libs/cart/base/cart-base.module.ts:L15-L15
Cart (augmentato)	feature-libs/cart/base/root/public_api.ts:L19-L19 (AUGMENTABLE_TYPES) → feature-libs/cart/base/root/models/cart.model.ts:L62-L62
CART_NORMALIZER	feature-libs/cart/base/root/public_api.ts:L15-L15 → feature-libs/cart/base/root/tokens/index.ts:L8-L8 → feature-libs/cart/base/root/tokens/converters.ts:L17-L17
Converter	core-libs/core/public_api.ts:L35-L35 → core-libs/core/src/util/index.ts:L11-L11 → core-libs/core/src/util/converter.service.ts:L21-L21
Occ	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L15-L15 → core-libs/core/src/occ/occ-models/index.ts:L8-L8 → core-libs/core/src/occ/occ-models/occ.models.ts:L7-L7

Perché funziona (file sorgente)

- `OccCartNormalizer` è registrato multi su `CART_NORMALIZER` in `CartBaseOccModule` (feature-libs/cart/base/occ/cart-base-occ.module.ts:L40-L44), quindi nell'injector lazy del cart.
- Nello stesso injector i multi-provider sono accodati nell'ordine di processing (angular:packages/core/src/di/r3_injector.ts:L497-L512), e i `providers` del wrapper sono processati dopo gli `imports` (angular:packages/core/src/di/provider_collection.ts:L293-L348): il normalizer custom segue `OccCartNormalizer`.
- `getMulti` legge ogni injector lazy con `{ self: true }` e concatena root → lazy (core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67, core-libs/core/src/lazy-loading/unified-injector.ts:L82-L92); `ConverterService` svuota la cache al caricamento del modulo (core-libs/core/src/util/converter.service.ts:L38-L45).
- `Cart` è augmentabile da `@spartacus/cart/base/root`, come fa OPF (feature-libs/cart/base/root/public_api.ts:L18-L20, integration-libs/opf/gift-card/root/model/augmented-opf-gift-card.model.ts:L12-L23).

Limiti e trappole

- **Root vs lazy**: un normalizer di `CART_NORMALIZER` registrato nella root gira **prima** di `OccCartNormalizer`, riceve `target` `undefined` e non vede il suo output. È il caso reale di `OpfGiftCardCartOccNormalizer`, registrato in `OpfGiftCardRootModule` (eager), che infatti lavora solo sui campi raw `sapGiftCards`, `sapGiftCardSummary` e `_availableOperations` (integration-libs/opf/gift-card/root/normalizers/opf-gift-card-cart-occ-normalizer.ts:L39-L42, integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L46-L56, integration-libs/opf/gift-card/root/normalizers/opf-gift-card-cart-occ-normalizer.ts:L18-L54).
- **Più moduli lazy sullo stesso token**: l'ordine dipende dall'ordine di caricamento dei moduli (core-libs/core/src/lazy-loading/lazy-modules.service.ts:L47-L55). DA VERIFICARE caso per caso.
- Il normalizer custom vale per tutte le conversioni `CART_NORMALIZER`, sia `pipeable` sia `pipeableMany` della lista `carts` (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L35-L44).

3.6 Ricetta: serializer custom (UI → backend)

Quando usarlo

Quando il body inviato al backend deve avere una forma diversa dal modello UI: rinominare campi, rimuovere campi che il backend rifiuta, aggiungere default. I token `*_SERIALIZER` sono applicati dall'adapter **prima** dell'invio (`POST`/`PATCH` nel caso degli indirizzi: core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L87-L89).

Esempio

Esempio illustrativo, non presente nel repo. Normalizza il telefono e rimuove `formattedAddress` (campo calcolato) prima dell'invio.

```ts
// app/user/address.serializer.ts  (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { Address, ADDRESS_SERIALIZER, Converter } from '@spartacus/core';

@Injectable({ providedIn: 'root' })
export class CleanAddressSerializer implements Converter<Address, any> {
  convert(source: Address, target?: any): any {
    target = target ?? { ...source };
    if (target.phone) {
      target.phone = String(target.phone).replace(/[\s\-()]/g, '');
    }
    delete target.formattedAddress;
    return target;
  }
}

@NgModule({
  providers: [
    {
      provide: ADDRESS_SERIALIZER,
      useExisting: CleanAddressSerializer,
      multi: true,
    },
```

```
  ],
})
export class AddressSerializerModule {}
```

SIMBOLO	CATENA
Address	core-libs/core/public_api.ts:L41-L41 (AUGMENTABLE_TYPES) → core-libs/core/src/model/address.model.ts:L38-L38
ADDRESS_SERIALIZER	core-libs/core/public_api.ts:L34-L34 → core-libs/core/src/user/index.ts:L8-L8 → core-libs/core/src/user/connectors/index.ts:L7-L7 → core-libs/core/src/user/connectors/address/index.ts:L9-L9 → core-libs/core/src/user/connectors/address/converters.ts:L20-L20
Converter	core-libs/core/public_api.ts:L35-L35 → core-libs/core/src/util/index.ts:L11-L11 → core-libs/core/src/util/converter.service.ts:L21-L21

Perché funziona (file sorgente)

- `OccUserAddressAdapter.add` chiama `this.converter.convert(address, ADDRESS_SERIALIZER)` prima della `POST`; lo stesso in `update` e `verify` (core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L64-L78, core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L87-L87, core-libs/core/src/occ/adapters/user/occ-user-address.adapter.ts:L106-L106).
- Nel repo nessun modulo registra un provider per `ADDRESS_SERIALIZER`: finché l'app non ne fornisce uno, `convert` restituisce il modello così com'è (core-libs/core/src/util/converter.service.ts:L106-L112).
- Il serializer di riferimento del core è `OccCostCenterSerializer` (`active`→`activeFlag`), registrato multi su `COST_CENTER_SERIALIZER` (core-libs/core/src/occ/adapters/cost-center/converters/occ-cost-center-serializer.ts:L12-L26, core-libs/core/src/occ/adapters/cost-center/cost-center-occ.module.ts:L33-L37).

Limiti e trappole

- **Raggio d'azione più ampio del previsto**: `ADDRESS_SERIALIZER` è applicato anche da `OccCheckoutDeliveryAddressAdapter.createAddress`, da `OccOrgUnitAdapter` e dall'adapter OPF quick-buy (feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-address.adapter.ts:L37-L42, feature-libs/organization/administration/occ/adapters/occ-org-unit.adapter.ts:L157-L157, feature-libs/organization/administration/occ/adapters/occ-org-unit.adapter.ts:L169-L169, integration-libs/opf/quick-buy/occ/adapters/occ-opf-quick-buy-cart.adapter.ts:L41-L41).
- **Non sempre il body è il modello serializzato**: `OccProductReviewsAdapter.post` applica `PRODUCT_REVIEW_SERIALIZER` e poi costruisce un body form-urlencoded con soli `headline`, `comment`, `rating`, `alias` (core-libs/core/src/occ/adapters/product/occ-product-reviews.adapter.ts:L39-L55): un campo aggiunto dal serializer non parte.
- Registrato nella root, il serializer è visibile anche agli adapter lazy (checkout) perché la catena è raccolta con `getMulti` (core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68).

3.7 Ricetta: campi OCC aggiuntivi (override di `fields` su un endpoint esistente)

Quando usarlo

Quando il campo esiste già nella risposta OCC del backend (estensione del DTO lato SAP Commerce) ma Spartacus non lo chiede nel parametro `fields`. È il caso reale di OPF gift card, che aggiunge `sapGiftCards(...)`, `sapGiftCardSummary(...)` e `_availableOperations` agli endpoint `carts` e `cart` (integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-cart-config.ts:L7-L24).

Esempio

Esempio illustrativo, non presente nel repo. Aggiunge `energyClass` al dettaglio prodotto e un campo `costCenterNote` al carrello. Per il cart la stringa di default va ricopiata **per intero** da `default-occ-cart-config-factory.ts` (qui abbreviata con `/* ... */`).

```ts
// app/spartacus/occ-fields.providers.ts  (esempio illustrativo, non presente nel repo)
import { Provider } from '@angular/core';
import { CartOccEndpoints } from '@spartacus/cart/base/occ';
import { OccConfig, provideConfig } from '@spartacus/core';

// L'uso del tipo mantiene vivo l'import che attiva l'augmentation di OccEndpoints
const cartEndpoints: CartOccEndpoints = {
  carts:
    'users/${userId}/carts?fields=carts(DEFAULT,/* ...stringa di default completa...
*/,saveTime,name,description,costCenterNote)',
  cart: 'users/${userId}/carts/${cartId}?fields=DEFAULT,/* ...stringa di default completa...
*/,saveTime,name,description,costCenterNote',
};

const occFieldsConfig: OccConfig = {
  backend: {
    occ: {
      endpoints: {
        // oggetto: si fonde con gli altri scope di `product` (list, attributes, ...)
        product: {
          details:
            'products/${productCode}?
fields=averageRating,stock(DEFAULT),description,availableForPickup,code,url,price(DEFAULT),numberOfRevie
ws,manufacturer,categories(FULL),priceRange,multidimensional,tags,images(FULL),energyClass',
        },
        ...cartEndpoints,
      },
    },
  },
};

export const occFieldsProviders: Provider[] = [provideConfig(occFieldsConfig)];
```

SIMBOLO	CATENA
CartOccEndpoints	feature-libs/cart/base/occ/public_api.ts:L9-L9 → feature-libs/cart/base/occ/model/index.ts:L7-L7 → feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L9-L9
OccConfig	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L12-L12 → core-libs/core/src/occ/config/occ-config.ts:L46-L46
provideConfig	core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18

Perché funziona (file sorgente)

- `provideConfig` registra un `ConfigChunk` che vince su ogni `provideDefaultConfig`/`provideDefaultConfigFactory` di libreria, anche su quelli forniti da moduli lazy come `CartBaseOccModule` (che usa `provideDefaultConfigFactory(defaultOccCartConfigFactory)`, anch'esso un `DefaultConfigChunk`: core-libs/core/src/config/config-providers.ts:L73-L78; core-libs/core/src/config/config-providers.ts:L11-L27, core-libs/core/src/config/config-tokens.ts:L10-L21, core-libs/core/src/config/services/configuration.service.ts:L92-L105, feature-libs/cart/base/occ/cart-base-occ.module.ts:L35-L35).
- La demo applica la config OPF proprio con `provideConfig` dal modulo feature root (projects/storefrontapp/src/app/spartacus/features/opf/opf-feature.module.ts:L85-L91); B2B fa lo stesso con `defaultB2bOccConfig` (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L27-L49, projects/storefrontapp/src/app/spartacus/spartacus-b2b-configuration.providers.ts:L20-L20).
- `OccEndpointsService` legge `backend.occ.endpoints` dall'oggetto config a ogni `buildUrl` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L196-L204).
- Il campo arriva nella risposta e, se non c'è un normalizer che lo rimuove, passa nel modello: i normalizer di default partono da una copia della sorgente (core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts:L17-L18).

Limiti e trappole

- **Le stringhe si sostituiscono**: `deepMerge` non concatena, quindi per aggiungere un campo bisogna ricopiare l'intero `fields` di default e tenerlo allineato a ogni upgrade (core-libs/core/src/config/utils/deep-merge.ts:L43-L45, feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L15-L19).
- **Cart:** `cart` **E** `carts`: il carrello `current` arriva da `carts` via `loadAll` (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L50); anche `createCart` ha i propri `fields` (feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L18-L19), che OPF non ridefinisce.
- **Oggetto vs stringa per gli scope**: `product: { details: '…' }` si fonde e conserva gli altri scope; `product: '…'` (stringa) li cancella tutti e ogni `buildUrl('product', { scope })` logga un warning in dev mode e usa la stringa (core-libs/core/src/config/utils/deep-merge.ts:L38-L45, core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L224).
- **Il template può cambiare il protocollo**: se `addEntries` contiene `quantity=${quantity}`, `OccCartEntryAdapter.add` passa a body form-urlencoded (caso B2B) (feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L32-L49).
- **L'augmentation va importata** per tipizzare la config (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L7-L17).

3.8 Ricetta: scope di caricamento prodotto custom

Quando usarlo

Quando un componente ha bisogno di pochi campi prodotto specifici e non si vuole appesantire lo scope `details` per tutte le pagine. Lo scope custom ha un suo endpoint con un suo `fields`, viene richiesto solo da chi lo usa e, se coincide come URL base con altri scope richiesti insieme, viene fuso in un'unica richiesta. Modello reale: bulk pricing con lo scope `bulkPrices` (feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L9-L20).

Esempio

Esempio illustrativo, non presente nel repo.

TS

```ts
// app/product/energy-scope.ts  (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';
import {
  OccConfig,
  ProductScope,
  ProductService,
  provideDefaultConfig,
} from '@spartacus/core';

export const ENERGY_SCOPE = 'energy';

export const energyScopeOccConfig: OccConfig = {
  backend: {
    occ: {
      endpoints: {
        product: {
          [ENERGY_SCOPE]: 'products/${productCode}?fields=energyClass',
        },
      },
    },
  },
};

@Injectable({ providedIn: 'root' })
export class EnergyLabelService {
  constructor(protected productService: ProductService) {}

  // Uno scope custom è una semplice stringa; combinato con uno scope standard
  getEnergyLabel(productCode: string): Observable<string | undefined> {
    return this.productService
      .get(productCode, [ProductScope.LIST, ENERGY_SCOPE])
      .pipe(map((product) => (product as any)?.energyClass));
  }
}

@NgModule({
  providers: [provideDefaultConfig(energyScopeOccConfig)],
})
export class EnergyScopeModule {}
```

SIMBOLO	CATENA
OccConfig	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L12-L12 → core-libs/core/src/occ/config/occ-config.ts:L46-L46
ProductScope	core-libs/core/public_api.ts:L52-L52 (AUGMENTABLE_TYPES) + core-libs/core/src/product/index.ts:L10-L10 → core-libs/core/src/product/model/index.ts:L8-L8 → core-libs/core/src/product/model/product-scope.ts:L7-L7

SIMBOLO	CATENA
ProductService	core-libs/core/public_api.ts:L29-L29 → core-libs/core/src/product/index.ts:L9-L9 → core-libs/core/src/product/facade/index.ts:L12-L12 → core-libs/core/src/product/facade/product.service.ts:L20-L20
provideDefaultConfig	core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L55-L55

Perché funziona (file sorgente)

- `ProductService.get` accetta `(ProductScope | string)[] | ProductScope | string` (core-libs/core/src/product/facade/product.service.ts:L39-L46); `ProductLoadingService.get` espande gli scope e, con più scope, unisce le parti con `uniteLatest` + `deepMerge`, emettendo solo quando tutte le parti sono definite (core-libs/core/src/product/services/product-loading.service.ts:L49-L86).
- `OccProductAdapter.getEndpoint` chiama `buildUrl('product', { urlParams, scope })` (core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L54-L59); `getEndpointForScope` restituisce `endpointConfig[scope]` se configurato (core-libs/core/src/occ/services/occ-endpoints.service.ts:L195-L225).
- `OccFieldsService.getOptimalUrlGroups` raggruppa gli scope che condividono l'URL senza `fields` e ricostruisce un unico URL con i `fields` fusi; `OccRequestsOptimizerService` fa una sola richiesta e per ogni scope estrae i propri campi (core-libs/core/src/occ/services/occ-fields.service.ts:L50-L82, core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L34-L75).
- Il default `product` è un oggetto per scope (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L10-L28), quindi l'oggetto con la nuova chiave si fonde con esso (Regola A).
- `LoadingScopes` permette `include`, `maxAge` e `reloadOn` per scope (core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L31); il core lo usa per `details.include = [LIST, VARIANTS]` (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58).

Limiti e trappole

- **Enum augmentato ≠ membro runtime**: bulk pricing aumenta `ProductScope` con `declare module` **e** assegna il valore a runtime con un cast (feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L7-L15). L'esempio evita il problema usando una costante stringa.
- `extractFields`: quando gli scope sono fusi, ogni scope riceve solo le chiavi dei propri `fields` (salvo `DEFAULT`/`FULL`/`BASIC`), e il normalizer gira dopo l'estrazione (core-libs/core/src/occ/utils/occ-fields.ts:L106-L125, core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L61-L73). Per questo l'esempio legge `energyClass` grezzo; se il normalizer della 3.4 è attivo, il campo `energyLabel` viene calcolato sulla parte dello scope `energy`.
- `LoadingScopes` e `LoadingScopeConfig` non sono esportati dai barrel: si può scrivere la chiave `backend.loadingScopes.product.<scope>` in config, ma non importare i tipi da `@spartacus/core` (dossier: nessuna occorrenza in `index.ts`/`public_api.ts`).
- Un adapter custom senza `loadMany` perde la fusione delle richieste (core-libs/core/src/product/connectors/product/product.connector.ts:L23-L29).

3.9 Ricetta: endpoint custom con scope e parametri dinamici (`OccEndpoints` + `buildUrl`)

Quando usarlo

Quando si aggiunge una chiamata OCC nuova (servizio custom esposto da un'estensione SAP Commerce) e la si vuole rendere configurabile dall'app come gli endpoint standard: path, `fields` per scope, query parameter dinamici.

Esempio

Esempio illustrativo, non presente nel repo. Endpoint `loyaltyTransactions` con due scope; l'adapter passa `urlParams`, `queryParams` e `scope`; l'app ridefinisce un solo scope.

```ts
// app/loyalty/loyalty-transactions.ts  (esempio illustrativo, non presente nel repo)
import { HttpClient } from '@angular/common/http';
import { inject, Injectable, NgModule } from '@angular/core';
import { Observable } from 'rxjs';
import {
  DEFAULT_SCOPE,
  OccConfig,
  OccEndpoint,
  OccEndpointsService,
  provideConfig,
  provideDefaultConfig,
} from '@spartacus/core';

export interface LoyaltyTransactionsOccEndpoints {
  loyaltyTransactions?: string | OccEndpoint;
}
declare module '@spartacus/core' {
  interface OccEndpoints extends LoyaltyTransactionsOccEndpoints {}
}

export const LOYALTY_SUMMARY_SCOPE = 'summary';

// Default "di libreria"
export const defaultLoyaltyTransactionsConfig: OccConfig = {
  backend: {
    occ: {
      endpoints: {
        loyaltyTransactions: {
          [DEFAULT_SCOPE]: 'users/${userId}/loyalty/transactions?fields=FULL',
          [LOYALTY_SUMMARY_SCOPE]:
            'users/${userId}/loyalty/transactions?fields=transactions(code,points)',
        },
      },
    },
  },
};

@Injectable({ providedIn: 'root' })
export class OccLoyaltyTransactionsAdapter {
  protected http = inject(HttpClient);
  protected occEndpoints = inject(OccEndpointsService);

  load(userId: string, currentPage: number, summary = false): Observable<unknown> {
    const url = this.occEndpoints.buildUrl('loyaltyTransactions', {
      urlParams: { userId },
```

```
      queryParams: { currentPage, pageSize: 20 },
      scope: summary ? LOYALTY_SUMMARY_SCOPE : DEFAULT_SCOPE,
    });
    return this.http.get(url);
  }
}

@NgModule({
  providers: [
    provideDefaultConfig(defaultLoyaltyTransactionsConfig),
    // Override "di app": vince sul default, e si fonde perché è un oggetto
    provideConfig({
      backend: {
        occ: {
          endpoints: {
            loyaltyTransactions: {
              [LOYALTY_SUMMARY_SCOPE]:
                'users/${userId}/loyalty/transactions?fields=transactions(code,points,date)',
            },
          },
        },
      },
    }),
  ],
})
export class LoyaltyTransactionsModule {}
```

SIMBOLO	CATENA
DEFAULT_SCOPE, OccEndpoint	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L15-L15 → core-libs/core/src/occ/occ-models/index.ts:L7-L7 → core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L7-L9
OccEndpoints (augmentato)	core-libs/core/public_api.ts:L51-L51 (AUGMENTABLE_TYPES) → core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L21
OccConfig	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L12-L12 → core-libs/core/src/occ/config/occ-config.ts:L46-L46
OccEndpointsService	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L16-L16 → core-libs/core/src/occ/services/index.ts:L7-L7 → core-libs/core/src/occ/services/occ-endpoints.service.ts:L34-L34
provideConfig, provideDefaultConfig	core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18, core-libs/core/src/config/config-providers.ts:L55-L55

Perché funziona (file sorgente)

- `buildUrl(endpoint, attributes?, propertiesToOmit?)` risolve il template per scope, sostituisce i placeholder con `StringTemplate.resolve(url, urlParams, true)`, aggiunge i `queryParams` e antepone baseUrl+prefix+baseSite (core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150).
- `StringTemplate.resolve` sostituisce ogni `${chiave}` presente in `urlParams` e applica `encodeURIComponent` ai valori (core-libs/core/src/config/utils/string-template.ts:L15-L31).

- Con `queryParams`, la query già nel template viene separata, ri-parsata e integrata con `HttpParams.set`; `null` cancella un parametro, `undefined` è ignorato (core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L146, core-libs/core/src/occ/services/occ-endpoints.service.ts:L152-L168); l'encoder è `HttpParamsURIEncoder` (core-libs/core/src/util/http-params-uri.encoder.ts:L7-L23).
- `getBaseUrl` compone baseUrl, prefix (default `'/occ/v2/'`) e baseSite con `urlPathJoin`; ogni parte si omette con `propertiesToOmit` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102, core-libs/core/src/occ/config/default-occ-config.ts:L9-L15, core-libs/core/src/occ/utils/occ-url-util.ts:L12-L33). Esempio reale: ASM costruisce `asmSessionEvent` con `{ baseSite: false, prefix: false }` (feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L78-L90).
- `SiteContextInterceptor` aggiunge `lang` e `curr` a ogni richiesta il cui URL contiene `getBaseUrl()` (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68).

Limiti e trappole

- `buildUrl` **non è tipizzato sulla chiave** (`endpoint: string`): un refuso nella chiave non dà errore di compilazione (core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L115). Incapsulare la chiamata in un metodo `protected get…Endpoint`, come fa OPF, limita il rischio (integration-libs/opf/gift-card/occ/adapters/occ-opf-gift-card.adapter.ts:L68-L82).
- **Endpoint non configurato** → nessun errore: viene usato **il nome della chiave** come path (`<baseUrl>/occ/v2/<site>/loyaltyTransactions`); scope mancante → warning solo in dev mode e fallback sul `default` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L199-L224).
- **Placeholder senza valore** restano letterali nell'URL (core-libs/core/src/config/utils/string-template.ts:L20-L30).
- `queryParams` **ricodifica la query del template** con `encodeURIComponent` (per esempio `,`→`%2C`; le parentesi restano invariate) e `null` può rimuovere anche il `fields` del template (core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L146, core-libs/core/src/util/http-params-uri.encoder.ts:L16-L23).
- **URL fuori dal baseSite** (`baseSite: false` / `prefix: false`) non ricevono `lang`/`curr` (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L58-L58).
- In una libreria vera i due provider dell'esempio starebbero in posti diversi: `provideDefaultConfig` nella libreria, `provideConfig` nell'app (JSDoc di `provideDefaultConfig`, core-libs/core/src/config/config-providers.ts:L50-L61).

3.10 Ricetta: meta resolver custom (`PageMetaResolver`)

Quando usarlo

Quando una pagina (identificata da tipo, template o uid CMS) richiede title, heading, description, breadcrumb o robots diversi da quelli del resolver generico. Esempi reali: checkout (per template) e dettaglio ticket (per uid) (feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts:L22-L77, feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts:L44-L101).

Esempio

Esempio illustrativo, non presente nel repo. Resolver per la content page con uid `loyalty` (uid inventato, va creato nel CMS): heading tradotto, breadcrumb "Account" inserito dopo "Home"; title, description, robots e canonical URL delegati a `BasePageMetaResolver`, come fa `ContentPageMetaResolver` che questo resolver sostituisce sulla pagina (core-libs/core/src/cms/page/content-page-meta.resolver.ts:L50-L52, core-libs/core/src/cms/page/content-page-meta.resolver.ts:L62-L64).

TS

```ts
// app/loyalty/loyalty-page-meta.resolver.ts  (esempio illustrativo, non presente nel repo)
import { Injectable, NgModule } from '@angular/core';
import { combineLatest, Observable } from 'rxjs';
import { map } from 'rxjs/operators';
import {
  BasePageMetaResolver,
  BreadcrumbMeta,
  CanonicalPageResolver,
  PageBreadcrumbResolver,
  PageDescriptionResolver,
  PageHeadingResolver,
  PageMetaResolver,
  PageRobotsMeta,
  PageRobotsResolver,
  PageTitleResolver,
  PageType,
  TranslationService,
} from '@spartacus/core';

@Injectable({ providedIn: 'root' })
export class LoyaltyPageMetaResolver
  extends PageMetaResolver
  implements
    PageTitleResolver,
    PageHeadingResolver,
    PageDescriptionResolver,
    PageBreadcrumbResolver,
    PageRobotsResolver,
    CanonicalPageResolver
{
  constructor(
    protected translation: TranslationService,
    protected basePageMetaResolver: BasePageMetaResolver
  ) {
    super();
    this.pageType = PageType.CONTENT_PAGE; // +1
    this.pageUid = 'loyalty'; // +10: batte ContentPageMetaResolver (score 1)
  }

  resolveTitle(): Observable<string | undefined> {
    return this.basePageMetaResolver.resolveTitle();
  }

  resolveHeading(): Observable<string> {
    return this.translation.translate('loyalty.pageHeading');
  }

  resolveBreadcrumbs(): Observable<BreadcrumbMeta[]> {
    return combineLatest([
```

```
      this.translation.translate('loyalty.accountBreadcrumb'),
      this.basePageMetaResolver.resolveBreadcrumbs(),
    ]).pipe(
      map(([label, breadcrumbs = []]) => {
        const [home, ...rest] = breadcrumbs;
        return [home, { label, link: '/my-account' }, ...rest];
      })
    );
  }

  resolveDescription(): Observable<string | undefined> {
    return this.basePageMetaResolver.resolveDescription();
  }

  resolveRobots(): Observable<PageRobotsMeta[]> {
    return this.basePageMetaResolver.resolveRobots();
  }

  // Senza questi due metodi la pagina perderebbe description e canonical URL
  resolveCanonicalUrl(): Observable<string | undefined> {
    return this.basePageMetaResolver.resolveCanonicalUrl();
  }
}

@NgModule({
  providers: [
    {
      provide: PageMetaResolver,
      useExisting: LoyaltyPageMetaResolver,
      multi: true,
    },
  ],
})
export class LoyaltyPageMetaModule {}
```

SIMBOLO	CATENA
BasePageMetaResolver	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L16-L16 → core-libs/core/src/cms/page/index.ts:L7-L7 → core-libs/core/src/cms/page/base-page-meta.resolver.ts:L28-L28
BreadcrumbMeta, PageRobotsMeta	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L15-L15 → core-libs/core/src/cms/model/page.model.ts:L80-L85
PageBreadcrumbResolver, PageHeadingResolver, PageRobotsResolver, PageTitleResolver	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L16-L16 → core-libs/core/src/cms/page/index.ts:L12-L12 → core-libs/core/src/cms/page/page.resolvers.ts:L20-L101
PageMetaResolver	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L16-L16 → core-libs/core/src/cms/page/index.ts:L11-L11 → core-libs/core/src/cms/page/page-meta.resolver.ts:L17-L17

SIMBOLO	CATENA
PageType	core-libs/core/public_api.ts:L26-L26 → core-libs/core/src/model/index.ts:L8-L8 → core-libs/core/src/model/cms.model.ts:L38-L38
TranslationService	core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L18-L18 → core-libs/core/src/i18n/translation.service.ts:L9-L9

Perché funziona (file sorgente)

- `PageMetaResolver.getScore` somma ±1 per `pageType`, ±1 per `pageTemplate`, ±10 per `pageUid` (solo per le proprietà valorizzate); `hasMatch` = score > 0, `getPriority` = score (core-libs/core/src/cms/page/page-meta.resolver.ts:L17-L54). `page.type/template/pageId` vengono da `typeCode/template/uid` della pagina OCC (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L52-L63).
- `ContentPageMetaResolver` ha solo `pageType = CONTENT_PAGE` (score 1) (core-libs/core/src/cms/page/content-page-meta.resolver.ts:L21-L65): il resolver dell'esempio ha score 11 sulla pagina `loyalty`.
- `PageMetaService.resolvers$` raccoglie le istanze con `getMulti` (root + lazy) e `resolveApplicable` sceglie la priorità più alta; a parità vince il provider registrato per ultimo (core-libs/core/src/cms/facade/page-meta.service.ts:L53-L57, core-libs/core/src/util/applicable.ts:L47-L87).
- `resolve()` chiama solo i metodi elencati in `pageMeta.resolvers` che esistono sul resolver scelto e li unisce con `uniteLatest` (core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103, core-libs/core/src/cms/page/config/default-page-meta.config.ts:L9-L40).
- `BasePageMetaResolver` (`providedIn: 'root'`) implementa title, description, breadcrumbs, robots, canonical e si usa per delega (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L25-L42, core-libs/core/src/cms/page/base-page-meta.resolver.ts:L81-L107); la registrazione multi con `useExisting` è quella di `CheckoutCoreModule` (feature-libs/checkout/base/core/checkout-core.module.ts:L17-L33).

Limiti e trappole

- **Parità di score**: con gli stessi criteri di un resolver esistente si vince solo se registrati dopo; un criterio non corrispondente vale -1/-10 e può portare lo score a ≤ 0 (core-libs/core/src/util/applicable.ts:L76-L86, core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L49).
- **Metodo assente = proprietà non risolta**: il resolver scelto sostituisce in blocco il generico; se manca `resolveDescription` la description non viene prodotta (core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103). Gli esempi reali delegano esplicitamente a `BasePageMetaResolver` i metodi che vogliono conservare: `CheckoutPageMetaResolver`, per esempio, non risolve né breadcrumbs né canonical URL (feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts:L32-L39).
- `pageMeta.resolvers` **è un array**: fornirne uno nuovo cancella le 7 voci di default (Regola A). Per aggiungere una proprietà bisogna ripartire da `defaultPageMetaConfig` (esportato) (core-libs/core/src/cms/page/config/default-page-meta.config.ts:L9-L40, core-libs/core/src/cms/page/index.ts:L8-L8).
- **Proprietà nuove non arrivano al DOM**: `SeoMetaService` scrive solo title/description/image/robots/canonicalUrl (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L32-L45); `PageMeta` non è nel blocco AUGMENTABLE_TYPES (core-libs/core/src/cms/model/page.model.ts:L40-L78, core-libs/core/public_api.ts:L38-L58). DA VERIFICARE se un'augmentation di `PageMeta` da app funziona sul pacchetto pubblicato (non verificabile senza `node_modules`).
- `BasePageMetaResolver` **non è un resolver**: non estende `PageMetaResolver`; registrato sul token multi non avrebbe `hasMatch`/`getPriority` (core-libs/core/src/cms/page/base-page-meta.resolver.ts:L28-L35, core-libs/core/src/util/applicable.ts:L68-L80).
- Per cambiare un solo metodo di un resolver esistente (es. il title prodotto) si può estendere `ProductPageMetaResolver`, che è `providedIn: 'root'` con dipendenze `protected` (core-libs/core/src/product/services/product-page-meta.resolver.ts:L37-L102), e ridefinire la classe concreta referenziata

da `useExisting` in `ProductModule` (`core-libs/core/src/product/product.module.ts:L16-L21`).
DA VERIFICARE a runtime: nessun esempio nel repo.

---

4. Diagrammi

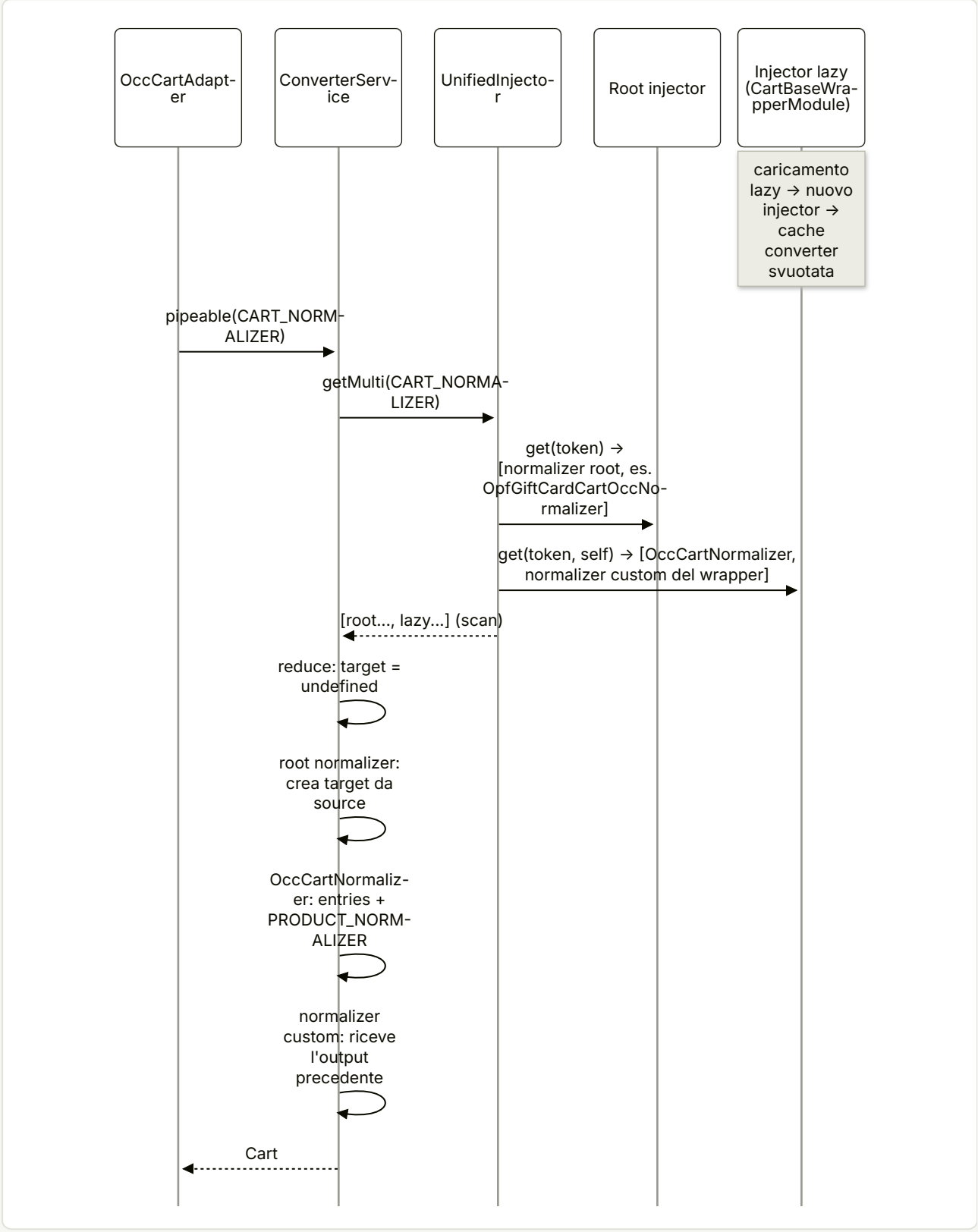
4.1 Dove registrare la personalizzazione

Decisione derivata dalle Regole A/B/C (§3) e dai binding reali: `ProductOccModule` nella root (`projects/storefrontapp/src/app/spartacus/spartacus-features.module.ts:L221-L221`), cart/order nei wrapper lazy (`projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L22-L25`, `projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L24-L27`).

■ TAVOLA 52.1 diagramma a tutta pagina, in fondo al capitolo

4.2 Catena di converter con un modulo lazy (cart)

Fatti: `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L107`, `core-libs/core/src/util/converter.service.ts:L38-L65`, `core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67`, `core-libs/core/src/util/converter.service.ts:L130-L137`.



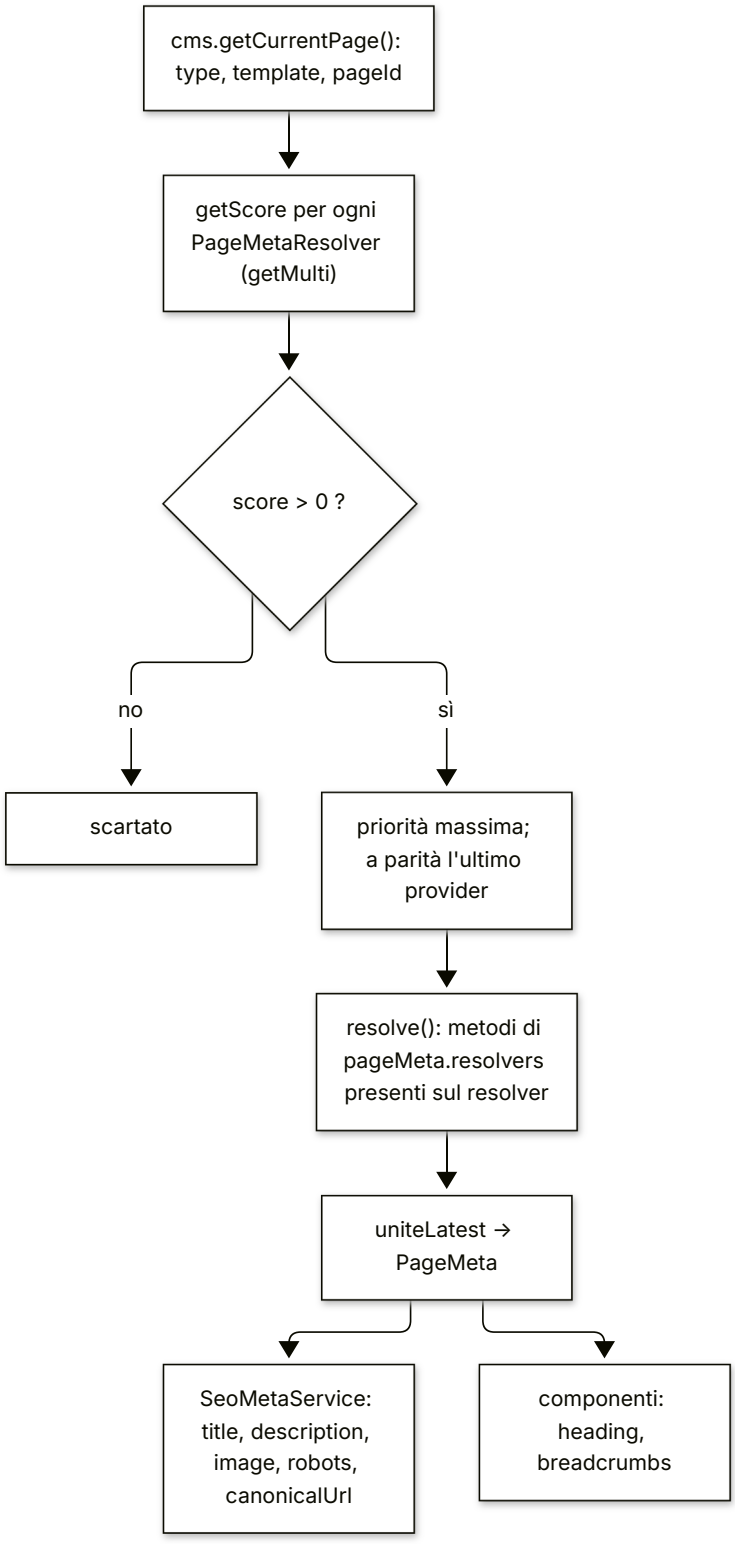
4.3 Da `buildUrl` alla richiesta HTTP

Fatti: `core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150`, `core-libs/core/src/occ/services/occ-endpoints.service.ts:L195-L225`, `core-libs/core/src/config/utils/string-template.ts:L15-L31`, `core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102`, `core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68`.

■ **TAVOLA 52.2** diagramma a tutta pagina, in fondo al capitolo

4.4 Selezione del meta resolver

Fatti: core-libs/core/src/cms/facade/page-meta.service.ts:L53-L63, core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L54, core-libs/core/src/util/applicable.ts:L63-L87, core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103, core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L32-L45.



5. Estratti di codice reale

Gli esempi delle ricette sono modellati su questi estratti.

Binding adapter + normalizer multi nella root (modello per 3.1 e 3.4) — core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L51

```ts
@NgModule({
  imports: [CommonModule],
  providers: [
    provideDefaultConfig(defaultOccProductConfig),
    {
      provide: ProductAdapter,
      useClass: OccProductAdapter,
    },
    // ...
    {
      provide: PRODUCT_NORMALIZER,
      useExisting: ProductImageNormalizer,
      multi: true,
    },
    {
      provide: PRODUCT_NORMALIZER,
      useExisting: ProductNameNormalizer,
      multi: true,
    },
```

Adapter OCC estendibile (modello per 3.1) — core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L20-L33

```ts
@Injectable()
export class OccProductAdapter implements ProductAdapter {
  constructor(
    protected http: HttpClient,
    protected occEndpoints: OccEndpointsService,
    protected converter: ConverterService,
    protected requestsOptimizer: OccRequestsOptimizerService
  ) {}

  load(productCode: string, scope?: string): Observable<Product> {
    return this.http
      .get(this.getEndpoint(productCode, scope))
      .pipe(this.converter.pipeable(PRODUCT_NORMALIZER));
  }
```

Sostituzione di un adapter in una feature lazy (modello per 3.2) — integration-libs/omf/order/omf-order.module.ts:L15-L26

```ts
@NgModule({
  providers: [
    OmfOrderHistoryService,
    provideDefaultConfig(defaultOmfConfig),
```

```
    { provide: OrderHistoryAdapter, useClass: OccOmfOrderHistoryAdapter },
    {
      provide: OrderHistoryFacade,
      useClass: OmfOrderHistoryService,
    },
  ],
})
export class OmfOrderModule {}
```

...registrata nel wrapper lazy dopo il modulo della feature — projects/storefrontapp/src/app/spartacus/features/order/order-wrapper.module.ts:L14-L27

```ts
const extensions: Type<any>[] = [];
if (environment.s4Service) {
  extensions.push(S4ServiceOrderModule);
}
if (environment.omf) {
  extensions.push(OmfOrderModule);
}
if (environment.opf) {
  extensions.push(OpfOrderModule);
}
@NgModule({
  imports: [OrderModule, ...extensions],
})
export class OrderWrapperModule {}
```

Override che riusa la logica ereditata — integration-libs/omf/order/occ-omf-order-history.adapter.ts:L69-L86

```ts
  public load(userId: string, orderCode: string): Observable<Order> {
    return this.getOrderGuid(orderCode).pipe(
      switchMap((guid) => {
        const url = this.getOrderDetailUrl(userId, orderCode);
        let headers = this.getRequestHeader(guid);
        if (userId === OCC_USER_ID_ANONYMOUS) {
          headers = InterceptorUtil.createHeader(
            USE_CLIENT_TOKEN,
            true,
            headers
          );
        }
        return this.http
          .get<Occ.Order>(url, { headers })
          .pipe(this.converter.pipeable(ORDER_NORMALIZER));
      })
    );
  }
```

Normalizer con campi raw non tipizzati (modello per 3.4 e 3.5) — integration-libs/opf/gift-card/root/normalizers/opf-gift-card-cart-occ-normalizer.ts:L18-L30

```ts
@Injectable({
  providedIn: 'root',
})
export class OpfGiftCardCartOccNormalizer implements Converter<Occ.Cart, Cart> {
  convert(source: Occ.Cart, target?: Cart): Cart {
    target ??= { ...(source as any) } as Cart;

    // Map sapGiftCards to opfGiftCards
    const sapGiftCards = (source as any).sapGiftCards;
    if (sapGiftCards && Array.isArray(sapGiftCards)) {
      target.opfGiftCards = sapGiftCards.map((card: any) =>
        this.convertGiftCard(card)
      );
    }
```

Registrazione nella root (eager) — gira prima dei normalizer lazy — integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L46-L56

```ts
  providers: [
    {
      provide: ORDER_NORMALIZER,
      useExisting: OpfGiftCardOrderOccNormalizer,
      multi: true,
    },
    {
      provide: CART_NORMALIZER,
      useExisting: OpfGiftCardCartOccNormalizer,
      multi: true,
    },
```

Serializer del core (modello per 3.6) — core-libs/core/src/occ/adapters/cost-center/converters/occ-cost-center-serializer.ts:L12-L26

```ts
@Injectable({
  providedIn: 'root',
})
export class OccCostCenterSerializer
  implements Converter<CostCenter, Occ.CostCenter>
{
  convert(source: CostCenter, target?: Occ.CostCenter): Occ.CostCenter {
    if (target === undefined) {
      target = { ...(source as any) } as Occ.CostCenter;
    }
    target.activeFlag = source.active;
```

```
    delete target.active;
    return target;
  }
```

Scope custom e augmentation dell'enum (modello per 3.8) — feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L7-L15 e feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L9-L20

```ts
import { ProductScope } from '@spartacus/core';

declare module '@spartacus/core' {
  enum ProductScope {
    BULK_PRICES = 'bulkPrices',
  }
}

(ProductScope as any)['BULK_PRICES'] = 'bulkPrices';
```

```ts
export const defaultOccBulkPricingConfig: OccConfig = {
  backend: {
    occ: {
      endpoints: {
        product: {
          bulkPrices:
            'orgProducts/${productCode}?fields=price(DEFAULT),volumePrices(FULL)',
        },
      },
    },
  },
};
```

Endpoint custom: augmentation + default + `buildUrl` **(modello per 3.3 e 3.9)** — integration-libs/opf/gift-card/occ/model/occ-opf-gift-card-endpoints.model.ts:L9-L22, integration-libs/opf/gift-card/occ/config/default-occ-opf-gift-card-config.ts:L9-L19, integration-libs/opf/gift-card/occ/adapters/occ-opf-gift-card.adapter.ts:L68-L72

```ts
export interface OccOpfGiftCardEndpoints {
  /**
   * Endpoint for applying gift card and getting balance
   */
  applyGiftCard?: string | OccEndpoint;
  /**
   * Endpoint for removing a gift card
   */
  removeGiftCard?: string | OccEndpoint;
}
```

```
declare module '@spartacus/core' {
  interface OccEndpoints extends OccOpfGiftCardEndpoints {}
}
```

TS

```
export const defaultGiftCardOccEndpointsConfig: OccConfig = {
  backend: {
    occ: {
      endpoints: {
        applyGiftCard: 'users/${userId}/carts/${cartId}/giftCards',
        removeGiftCard:
          'users/${userId}/carts/${cartId}/giftCards/${giftCardId}',
      },
    },
  },
};
```

TS

```
  protected getApplyGiftCardEndpoint(userId: string, cartId: string): string {
    return this.occEndpoints.buildUrl('applyGiftCard', {
      urlParams: { userId, cartId },
    });
  }
```

Meta resolver per template, con delega (modello per 3.10) — feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts:L32-L68

TS

```
export class CheckoutPageMetaResolver
  extends PageMetaResolver
  implements
    PageHeadingResolver,
    PageTitleResolver,
    PageDescriptionResolver,
    PageRobotsResolver
{
  constructor(
    protected translationService: TranslationService,
    protected activeCartFacade: ActiveCartFacade,
    protected basePageMetaResolver: BasePageMetaResolver
  ) {
    super();
    this.pageType = PageType.CONTENT_PAGE;
    this.pageTemplate = 'MultiStepCheckoutSummaryPageTemplate';
  }
  // ...
  resolveHeading(): Observable<string> {
    return this.translationService.translate('pageMetaResolver.checkout.title');
  }
```

Breadcrumb composti con `combineLatest` (modello per 3.10) — feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts:L81-L91

```ts
resolveBreadcrumbs(): Observable<BreadcrumbMeta[]> {
  return combineLatest([
    this.customerServiceBreadCrumb$,
    this.basePageMetaResolver.resolveBreadcrumbs(),
  ]).pipe(
    map(([customerServiceBreadCrumb, breadcrumbs = []]) => {
      const [home, ...restBreadcrumbs] = breadcrumbs;
      return [home, ...customerServiceBreadCrumb, ...restBreadcrumbs];
    })
  );
}
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
backend.occ.baseUrl	string	nessuno	core-libs/core/src/occ/config/occ-config.ts:L15-L15	core-libs/core/src/occ/services/occ-endpoints.service.ts:L93-L96
backend.occ.prefix	string	'/occ/v2/'	core-libs/core/src/occ/config/occ-config.ts:L16-L16, core-libs/core/src/occ/config/default-occ-config.ts:L12-L12	core-libs/core/src/occ/services/occ-endpoints.service.ts:L240-L248
backend.occ.endpoints.<key>	string | OccEndpoint	per libreria (vedi A04)	core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L299 + augmentation	core-libs/core/src/occ/services/occ-endpoints.service.ts:L170-L225
backend.occ.endpoints.<key>.<scope>	string	es. product.{default,list,details,...}	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28	core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L218
backend.loadingScopes.<model>.<scope>.{include,maxAge,reloadOn}	LoadingScopeConfig	product.details.include = [list, variants]	core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L31, core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L52-L58	core-libs/core/src/occ/services/loading-scopes.service.ts:L29-L77

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`pageMeta.resolvers`	`PageMetaResolverConfig[]`	7 voci (title, heading, breadcrumbs, description, image, robots, canonicalUrl)	`core-libs/core/src/cms/page/config/page-meta.config.ts:L22-L41`, `core-libs/core/src/cms/page/config/default-page-meta.config.ts:L11-L40`	`core-libs/core/src/cms/facade/page-meta.service.ts:L115-L125`
`features.disableConfigUpdates`	`boolean`	non impostato	`core-libs/core/src/features-config/config/features-config.ts:L116-L116`	`core-libs/core/src/config/services/configuration.service.ts:L103-L103`
`omf.guidHttpHeaderName` (esempio reale di config per adapter custom)	`string`	`'guid'`	`integration-libs/omf/order/config/default-omf-config.ts:L9-L13`	`integration-libs/omf/order/occ-omf-order-history.adapter.ts:L57-L65`

7. Commenti e note notevoli dal sorgente

- "Product's data can be loaded from alternative sources, as long as the structure converts to the `Product`." — il contratto che rende possibile la ricetta 3.1 (`core-libs/core/src/product/connectors/product/product.adapter.ts:L13-L15`).
- "Converters can be stacked together to to apply decoupled customizations" (con il refuso "to to" nell'originale) — `core-libs/core/src/util/converter.service.ts:L19-L19`.
- "// Clear cached converters when new injectors appear" — la ragione per cui un normalizer lazy entra in catena dopo il caricamento del modulo (`core-libs/core/src/util/converter.service.ts:L39-L39`).
- "// we are grouping all load actions that happens at the same time / to optimize loading and pass them all to productConnector.getMany" — spiega perché `loadMany` è il metodo che conta (`core-libs/core/src/product/store/effects/product.effect.ts:L45-L46`).
- "// TODO: Can we reuse getEndpointFromConfig in this method? Should we change behavior of this function?" — su `getEndpointForScope`, il cui fallback (nome della chiave come path) è un comportamento da conoscere (`core-libs/core/src/occ/services/occ-endpoints.service.ts:L195-L195`).
- "// TODO 4.0: default to encodeVariable = true" — `StringTemplate.resolve` codifica solo se richiesto; `buildUrl` passa `true` (`core-libs/core/src/config/utils/string-template.ts:L24-L24`, `core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150`).
- "// compatibility mechanism, can be disabled with feature toggle" — la fusione in place della config lazy (`core-libs/core/src/config/services/configuration.service.ts:L102-L102`).
- "applicables with the same priority are sorted by order of providers, the applicable that was provided later wins" — `core-libs/core/src/util/applicable.ts:L57-L57`.
- "// TODO: Revisit if typing is possible here with Template Literal Types when we update to TS >=4.1" — l'invocazione dei metodi dei resolver è non tipizzata (`(metaResolver as any)[...]`) (`core-libs/core/src/cms/facade/page-meta.service.ts:L87-L87`).
- "// We need this import for augmentation of OccEndpoints to pick up" — `core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L7-L7`.

- "// Handle b2b case where the x-www-form-urlencoded is still used" — feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L36-L36.
- "It's a workaround to make TS types augmentable" — il builder AUGMENTABLE_TYPES (tools/build-lib/augmented-types/index.js:L49-L51).
- "// Uncomment this when occ gets configured" — `productReferences` con `referenceType` lasciato commentato (core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L32-L34).

---

8. Extension point

EXTENSION POINT	MECCANISMO	RICETTA	CAPITOLO DI RIFERIMENTO
`ProductAdapter` e altri adapter del core	token singolo, `useClass`	§3.1	21 — OCC, 24 — Prodotto e ricerca
`CartAdapter`, `OrderHistoryAdapter`, adapter di feature lazy	token singolo nel wrapper lazy	§3.2	25 — Carrello, 27 — Ordini
Adapter/connector di un dominio nuovo	classe astratta + connector `providedIn: 'root'`	§3.3	21 — OCC
`PRODUCT_NORMALIZER`, `ORDER_NORMALIZER`, ...	multi-provider, catena `reduce`	§3.4	21 — OCC, 24
`CART_NORMALIZER` dopo il default	multi-provider nello stesso injector lazy	§3.5	25 — Carrello
`ADDRESS_SERIALIZER`, `PRODUCT_REVIEW_SERIALIZER`, `COST_CENTER_SERIALIZER`	multi-provider applicato prima dell'invio	§3.6	21 — OCC
`backend.occ.endpoints.<key>` (`fields`)	`provideConfig` + `deepMerge`	§3.7	A04 — Endpoint OCC, 25
`backend.occ.endpoints.product.<scope>`, `loadingScopes`	scope + fusione richieste	§3.8	24 — Prodotto e ricerca
`OccEndpoints` (augmentation), `OccEndpointsService.buildUrl`	module augmentation + template	§3.9	21 — OCC, A04
`PageMetaResolver`, `pageMeta.resolvers`	multi-provider + score	§3.10	40 — SEO e PWA

Ricette collegate in Parte III:

- 50 — Ricette: configurazione, DI, interceptor, eventi: `provideConfig`, ordine dei provider, override di servizi; un interceptor è spesso l'alternativa migliore a un adapter esteso solo per aggiungere un header.
- 51 — Ricette: componenti CMS, outlet, route e layout: i componenti che consumano i campi aggiunti con normalizer e scope.
- 53 — Ricette: feature lib lazy, carrello/checkout, stili, traduzioni: come costruire il wrapper lazy usato nelle ricette 3.2 e 3.5, e le chiavi di traduzione usate dai meta resolver.
- 54 — API pubblica vs interna, upgrade, anti-pattern, checklist: perché importare solo da entry point pubblici e come gestire le stringhe `fields` ricopiate negli upgrade.

9. Trappole, casi limite, comportamenti controintuitivi

1. **Override nell'injector sbagliato.** Adapter e connector del cart vivono nell'injector lazy: un override nella root viene oscurato (feature-libs/cart/base/occ/cart-base-occ.module.ts:L36-L39, feature-libs/cart/base/core/cart-base-core.module.ts:L29-L30, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L96-L97).
2. **Ordine nello stesso injector.** Per un adapter eager vince l'ultimo provider; un modulo di override importato prima di `ProductOccModule` perde (angular:packages/core/src/di/provider_collection.ts:L293-L348, angular:packages/core/src/di/r3_injector.ts:L513-L521).
3. `load` **vs** `loadMany`**.** Il percorso degli effect prodotto passa da `getMany`→`loadMany` (core-libs/core/src/product/store/effects/product.effect.ts:L45-L53, core-libs/core/src/product/connectors/product/product.connector.ts:L23-L32).
4. **Normalizer root prima dei lazy.** `getMulti` concatena root → lazy: un normalizer di `CART_NORMALIZER` nella root non vede l'output di `OccCartNormalizer` (core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67, integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L52-L56).
5. **Il primo normalizer crea il target.** `reduce` parte da `undefined` (core-libs/core/src/util/converter.service.ts:L130-L137).
6. `multi: true` **dimenticato = errore, non sostituzione.** (angular:packages/core/src/di/r3_injector.ts:L500-L505 multi dopo singolo, angular:packages/core/src/di/r3_injector.ts:L513-L519 singolo dopo multi, core-libs/core/src/lazy-loading/unified-injector.ts:L84-L88). In produzione (senza `ngDevMode`) un singolo registrato dopo i multi sostituisce il record (`this.records.set(token, record)`), l'istanza non è un array e `UnifiedInjector.getMulti`, usato da `ConverterService`, lancia `Multi-providers mixed with single providers`; un multi registrato dopo un singolo trova `multi` indefinito e `multiRecord.multi!.push(provider)` lancia un `TypeError` (angular:packages/core/src/di/r3_injector.ts:L497-L521, core-libs/core/src/util/converter.service.ts:L57-L57).
7. **Normalizer prodotto ovunque.** `PRODUCT_NORMALIZER` gira anche su entry di carrello, ordini, resi, quote, con prodotti parziali (feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts:L31-L39, feature-libs/order/occ/adapters/converters/occ-order-normalizer.ts:L74-L74).
8. **Serializer pass-through e raggio d'azione.** Senza provider `convert` restituisce il modello (core-libs/core/src/util/converter.service.ts:L106-L112); con un provider, `ADDRESS_SERIALIZER` tocca anche checkout e organization (feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-address.adapter.ts:L37-L42).
9. `fields` **= sostituzione dell'intera stringa.** (core-libs/core/src/config/utils/deep-merge.ts:L43-L45, integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-cart-config.ts:L11-L13).
10. `cart` **E** `carts` **(e** `createCart`**).** (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L50, feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L15-L19).
11. **Stringa al posto dell'oggetto per scope** cancella tutti gli scope (core-libs/core/src/config/utils/deep-merge.ts:L38-L45, core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L224).
12. **Endpoint non configurato** → la chiave diventa il path (core-libs/core/src/occ/services/occ-endpoints.service.ts:L199-L224).
13. **Placeholder non risolti** restano nell'URL (core-libs/core/src/config/utils/string-template.ts:L20-L30).
14. `queryParams` **ricodifica e può cancellare il** `fields` **del template** (core-libs/core/src/occ/services/occ-endpoints.service.ts:L125-L146, core-libs/core/src/occ/services/occ-endpoints.service.ts:L159-L164).

15. **Niente** `lang`/`curr` **fuori dal baseSite** (core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L58-L58).
16. `buildUrl` **accetta qualsiasi stringa** (core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L115).
17. **Augmentation non importata = non attiva** (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L7-L17, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L26-L26).
18. **Enum augmentato senza membro runtime** (feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L9-L15).
19. `extractFields` **per scope** (core-libs/core/src/occ/utils/occ-fields.ts:L117-L125).
20. `pageMeta.resolvers` **sostituito, non esteso** (core-libs/core/src/config/utils/deep-merge.ts:L7-L9, core-libs/core/src/cms/page/config/default-page-meta.config.ts:L11-L40).
21. **Parità di score dei resolver** (core-libs/core/src/util/applicable.ts:L76-L86).
22. **Proprietà meta nuove non renderizzate** (core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L39-L45).
23. `disableConfigUpdates`. Se attivo, la config dei moduli lazy non viene fusa nell'oggetto `Config` che `OccEndpointsService` legge: gli endpoint di default forniti solo da moduli lazy potrebbero non essere visibili (core-libs/core/src/config/services/configuration.service.ts:L102-L105, core-libs/core/src/occ/services/occ-endpoints.service.ts:L46-L48). `DA VERIFICARE` end-to-end.
24. **Doc della skill imprecisa** su `multi: true` e sui normalizer "price, and URL" del core, che non esistono in `ProductOccModule` (core-libs/skills/skills/spartacus-developer/references/normalizers.md:L5-L7, core-libs/core/src/occ/adapters/product/product-occ.module.ts:L42-L51).

---

10. Come fare debug

- **L'URL non è quello atteso.** Iniettare `OccEndpointsService` e confrontare `getRawEndpointValue(endpoint)` (template non risolto) con `buildUrl(...)`; `isConfigured(endpoint, scope)` dice se la chiave/scope esiste **senza** fallback (core-libs/core/src/occ/services/occ-endpoints.service.ts:L57-L79, core-libs/core/src/occ/services/occ-endpoints.service.ts:L170-L193). Un URL che finisce con il nome della chiave indica un endpoint non configurato (core-libs/core/src/occ/services/occ-endpoints.service.ts:L199-L224).
- **Scope ignorato.** In dev mode compare il warning `${endpoint} endpoint configuration missing for scope "${scope}"` (core-libs/core/src/occ/services/occ-endpoints.service.ts:L206-L218).
- **Override di config che "non prende".** Iniettare `OccConfig` (è lo stesso oggetto `Config`, core-libs/core/src/occ/config/occ-config.ts:L42-L52) e ispezionare `backend.occ.endpoints` dopo il caricamento del modulo lazy; ricordare che le stringhe sostituiscono (core-libs/core/src/config/utils/deep-merge.ts:L31-L46).
- **Normalizer che non gira o gira nell'ordine sbagliato.** Verificare in quale injector è registrato (root vs wrapper lazy) e se il modulo lazy è già stato caricato: prima del caricamento la catena contiene solo le istanze root (core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68, core-libs/core/src/util/converter.service.ts:L38-L45). Un breakpoint in `ConverterService.convertSource` mostra l'array effettivo (core-libs/core/src/util/converter.service.ts:L130-L137).
- **Errore** `Multi-providers mixed with single providers`: manca `multi: true` su una registrazione dello stesso token (core-libs/core/src/lazy-loading/unified-injector.ts:L84-L88).
- **Adapter custom ignorato.** Breakpoint nel costruttore: se non viene istanziato, il token è risolto da un altro injector (Regola B) o un provider successivo lo sovrascrive; per il prodotto controllare anche che `loadMany`

sia stato ridefinito (core-libs/core/src/product/connectors/product/product.connector.ts:L23-L32).

- **Campo assente nel modello.** Controllare la risposta HTTP: se il campo non c'è, manca nel `fields` dell'endpoint/scope effettivamente usato (per il cart `current`: `carts`) (feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L50); se c'è nella risposta ma non nel modello, con scope fusi può essere stato filtrato da `extractFields` (core-libs/core/src/occ/utils/occ-fields.ts:L106-L125).
- **Meta resolver non scelto.** Calcolare a mano lo score con `type`, `template`, `pageId` della pagina (core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L52-L63, core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L49); se la proprietà manca nel `PageMeta`, verificare che il metodo esista sul resolver e che la voce sia in `pageMeta.resolvers` (core-libs/core/src/cms/facade/page-meta.service.ts:L84-L103, core-libs/core/src/cms/facade/page-meta.service.ts:L115-L125).

---

11. Domande di autoverifica

1. **Perché un `{ provide: CartAdapter, useClass: X }` nell'`AppModule` non ha effetto nella demo?** Perché `CartAdapter` è legato in `CartBaseOccModule` e `CartConnector` è ri-fornito in `CartBaseCoreModule`, entrambi nell'injector lazy del wrapper: il connector risolve l'adapter di quell'injector (feature-libs/cart/base/occ/cart-base-occ.module.ts:L36-L39, feature-libs/cart/base/core/cart-base-core.module.ts:L22-L36, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L22-L25).
2. **Estendo `OccProductAdapter` e ridefinisco solo `load`: perché le pagine prodotto non cambiano?** Gli effect chiamano `productConnector.getMany`, che usa `loadMany` quando l'adapter lo implementa, e `OccProductAdapter` lo implementa (core-libs/core/src/product/store/effects/product.effect.ts:L45-L53, core-libs/core/src/product/connectors/product/product.connector.ts:L23-L32, core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L35-L52).
3. **Un normalizer di `CART_NORMALIZER` registrato nella root riceve un `target` popolato da `OccCartNormalizer`?** No: `getMulti` concatena prima le istanze root e poi quelle lazy, e `reduce` parte da `undefined`; il normalizer root gira per primo e deve creare il target (core-libs/core/src/lazy-loading/unified-injector.ts:L56-L67, core-libs/core/src/util/converter.service.ts:L130-L137).
4. **Cosa succede se dimentico `multi: true` registrando un normalizer?** Non c'è sostituzione silenziosa: Angular in dev mode lancia l'errore di provider misti nello stesso injector, e `UnifiedInjector.getMulti` lancia `Multi-providers mixed with single providers` se trova un'istanza non-array (angular:packages/core/src/di/r3_injector.ts:L500-L505, angular:packages/core/src/di/r3_injector.ts:L513-L519, core-libs/core/src/lazy-loading/unified-injector.ts:L84-L88).
5. **Voglio aggiungere un campo al `fields` del carrello: basta ridefinire `cart`?** No: la stringa sostituisce quella di default (va ricopiata per intero) e il carrello `current` è caricato con `carts` via `loadAll`; anche `createCart` ha i suoi `fields` (core-libs/core/src/config/utils/deep-merge.ts:L43-L45, feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L46-L50, feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L15-L19).
6. **Cosa restituisce `buildUrl('fooBar')` se `fooBar` non è configurato?** Un URL con `fooBar` come path dopo baseUrl+prefix+baseSite, senza errori (core-libs/core/src/occ/services/occ-endpoints.service.ts:L199-L224, core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102).
7. **Perché bulk pricing assegna `(ProductScope as any)['BULK_PRICES']` dopo il `declare module`?** Perché l'augmentation di un enum aggiunge solo il tipo, non il membro a runtime; in alternativa si può passare una stringa a `ProductService.get` (feature-libs/product/bulk-pricing/core/model/augmented-core.model.ts:L7-L15, core-libs/core/src/product/facade/product.service.ts:L39-L46).
8. **Un resolver con `pageType = CONTENT_PAGE` e `pageUid = 'x'` batte `ContentPageMetaResolver` sulla pagina `x`? E sulle altre content page?** Sulla pagina `x` sì (11 contro 1). Sulle altre content page ha score 1 − 10 =

−9, quindi `hasMatch` è falso e non viene considerato (`core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L54`, `core-libs/core/src/cms/page/content-page-meta.resolver.ts:L21-L65`).

9. **Aggiungo** `pageMeta: { resolvers: [{ property: 'foo', method: 'resolveFoo' }] }` **con** `provideConfig`**: cosa si rompe?** L'array sostituisce le 7 voci di default, quindi title, description, robots, ecc. non vengono più risolti; e `foo` non arriva comunque al DOM perché `SeoMetaService` scrive solo 5 proprietà (`core-libs/core/src/config/utils/deep-merge.ts:L7-L9`, `core-libs/core/src/cms/page/config/default-page-meta.config.ts:L11-L40`, `core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L32-L45`).
10. **Un endpoint costruito con** `{ baseSite: false, prefix: false }` **riceve** `lang` **e** `curr`**?** No: `SiteContextInterceptor` li aggiunge solo agli URL che contengono `getBaseUrl()` (baseUrl+prefix+baseSite) (`core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L54-L68`, `feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L78-L90`).

TAVOLA 52.1 **4.1 Dove registrare la personalizzazione**

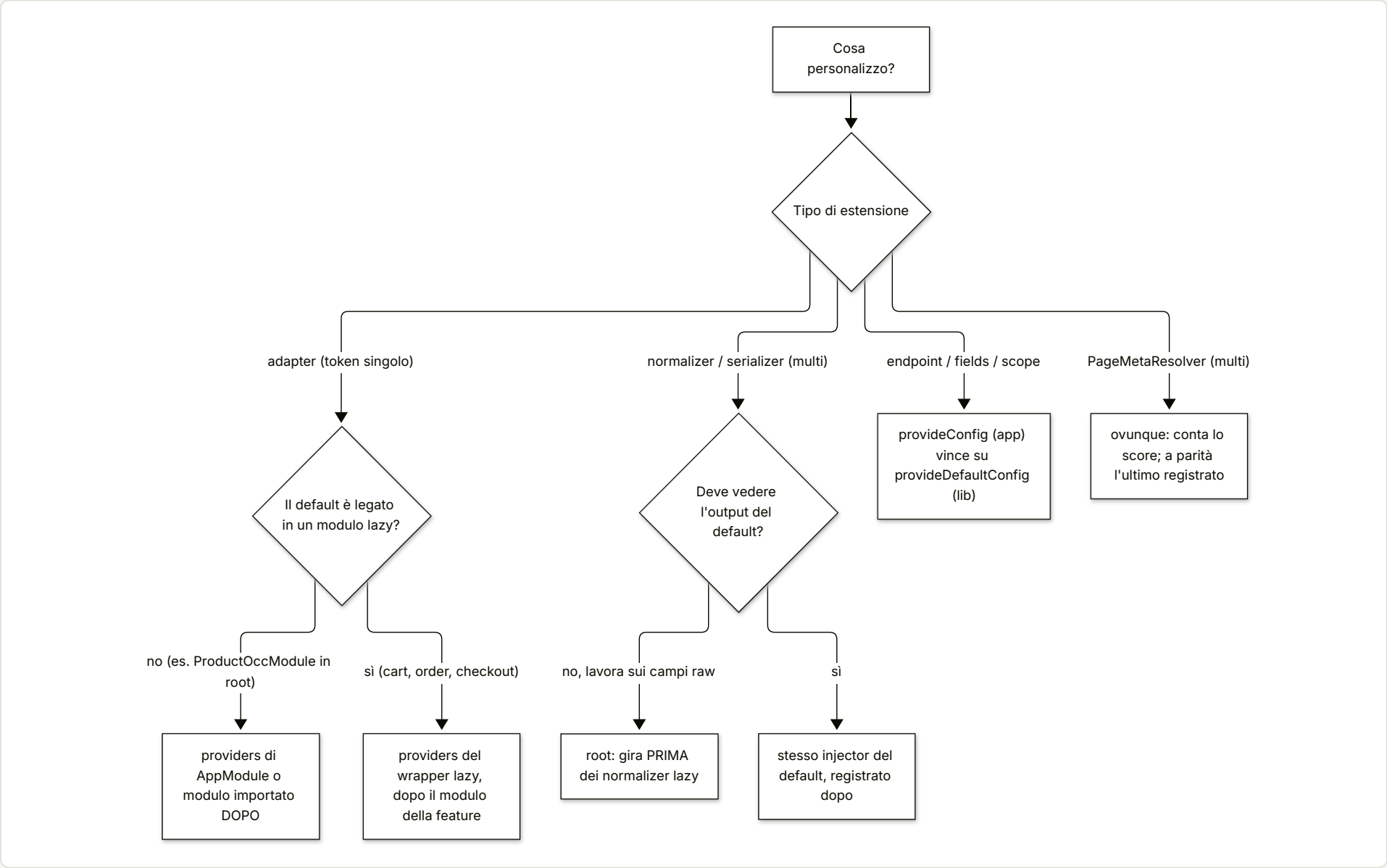
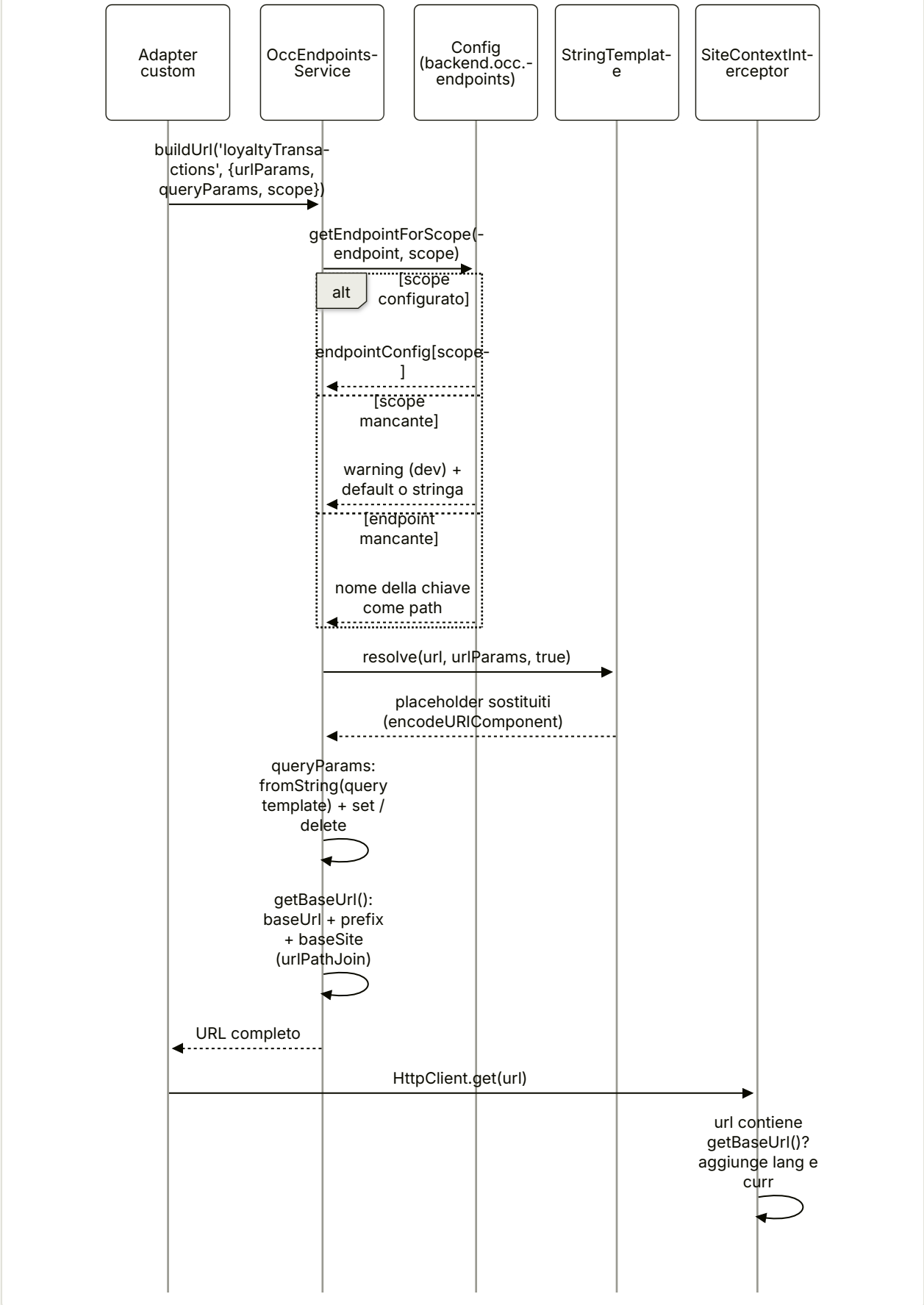


TAVOLA 52.2 **4.3 Da buildUrl alla richiesta HTTP**



53

Ricette: feature lib lazy, carrello/checkout, stili, traduzioni

Una feature lazy propria, estensioni di carrello e checkout, stili senza fork, traduzioni.

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Questo capitolo raccoglie le ricette per i quattro lavori più frequenti in un progetto Spartacus reale, quando l'app non si limita a configurare ma *aggiunge* funzionalità:

1. **creare una propria feature lazy-loaded** che si comporti come le feature lib ufficiali (root module eager, chunk lazy, componenti CMS che fanno scattare il caricamento, facade proxy utilizzabile dal codice eager);
2. **estendere il carrello e il checkout**: un campo nuovo sulla riga carrello, un add-to-cart diverso, la reazione agli eventi di carrello, uno step di checkout in più, un componente di step sostituito;
3. **personalizzare gli stili** senza forkare `@spartacus/styles`: escludere lo stile di un componente, riscriverlo, cambiare i colori del tema a build-time e a runtime;
4. **gestire le traduzioni**: sovrascrivere chiavi esistenti e aggiungere un chunk proprio.

Le ricette non introducono meccanismi nuovi: combinano quelli spiegati nella Parte II. Per questo ogni ricetta rimanda al capitolo che descrive il meccanismo in dettaglio:

- feature module, `featureModules`, dipendenze e injector dei chunk: capitolo 12;
- facade proxy (`facadeFactory`) ed eventi: capitolo 20;
- carrello: capitolo 25;
- checkout: capitolo 26;
- i18n: capitolo 37;
- stili e theming: capitolo 38.

Il filo conduttore di tutte le ricette è una regola di merge della configurazione: **la config dell'app vince sempre** su quella delle librerie e su quella dei chunk lazy (`core-libs/core/src/config/services/configuration.service.ts:L92-L106`), e **gli array vengono sostituiti, non concatenati** (`core-libs/core/src/config/utils/deep-merge.ts:L7-L9`, `:L36-L45`). Metà delle trappole di questo capitolo discende da queste due righe.

Un secondo filo conduttore è *dove* vive il codice: nel bundle iniziale (root module, config eager, traduzioni statiche) oppure nell'injector di un chunk lazy (servizi della feature, override di provider, outlet registrati con `MODULE_INITIALIZER`). Sbagliare injector è la causa tipica di un'estensione che "non fa nulla" senza dare errori.

2. Mappa dei file

File sorgente che rendono possibili le ricette (abilitanti) e file del repo da cui gli esempi sono modellati.

PATH	RUOLO
`core-libs/core/src/cms/config/cms-config.ts`	`FeatureModuleConfig` (`module`, `dependencies`, `cmsComponents`), `CmsConfig.featureModules`, `CmsComponentMapping` (`guards`, `i18nKeys`)
`core-libs/core/src/config/config-providers.ts`	`provideConfig` (`ConfigChunk`) e `provideDefaultConfig`/`provideDefaultConfigFactory` (`DefaultConfigChunk`)

PATH	RUOLO
core-libs/core/src/config/config-tokens.ts	RootConfig = merge dei ConfigChunk, Config = DefaultConfig + RootConfig
core-libs/core/src/config/services/configuration.service.ts	merge della config portata dai chunk lazy (ambientDefault, ambient) sotto la config root
core-libs/core/src/config/utils/deep-merge.ts	semantica di merge: oggetti ricorsivi, array e primitive sostituiti
core-libs/core/src/lazy-loading/feature-modules.service.ts	isConfigured, resolveFeature, alias e dipendenze
core-libs/core/src/lazy-loading/lazy-modules.service.ts	creazione del modulo lazy, injector padre, MODULE_INITIALIZER con self: true
core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts	facadeFactory()
core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts	FacadeDescriptor (methods, properties, async)
core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts	proxy: resolver, call() con connessione immediata
core-libs/storefront/cms-structure/services/cms-features.service.ts	mappa tipo CMS → feature, config CMS letta dai chunk del modulo lazy
core-libs/storefront/cms-structure/services/cms-components.service.ts	merge mapping feature + statico, getGuards, warning "No component implementation found"
core-libs/storefront/cms-structure/services/cms-guards.service.ts	istanze dei guard dei componenti CMS da UnifiedInjector
core-libs/storefront/cms-structure/outlet/outlet.providers.ts	provideOutlet
core-libs/storefront/cms-structure/outlet/outlet.module.ts	OutletModule.forRoot() / forChild() e registerOutletsFactory
feature-libs/cart/saved-cart/root/saved-cart-root.module.ts	modello di root module (featureModules senza module, rotte, alias core)
feature-libs/cart/saved-cart/root/facade/saved-cart.facade.ts	modello di facade proxy
feature-libs/cart/saved-cart/core/facade/facade-providers.ts	modello di ri-fornitura della facade nel chunk (useExisting)
projects/storefrontapp/src/app/spartacus/features/cart/cart-saved-cart-feature.module.ts	modello di feature module dell'app (fornisce module + i18n)
feature-libs/cart/base/root/facade/active-cart.facade.ts	ActiveCartFacade (addEntry, getEntries)
feature-libs/cart/base/root/models/cart-outlets.model.ts	enum CartOutlets
feature-libs/cart/base/root/models/cart-item-context.model.ts	CartItemContext
feature-libs/cart/base/root/public_api.ts	export del root carrello, OrderEntry aumentabile
feature-libs/cart/base/root/events/cart.events.ts	CartAddEntrySuccessEvent, CartUiEventAddToCart
feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts	AddToCartComponent (metodi pubblici/protected estendibili)

PATH	RUOLO
feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts	mapping di default di ProductAddToCartComponent
feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts	endpoint cart/carts/createCart con fields
feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts	spread delle entry: i campi extra arrivano nel modello
feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts	modello reale di estensione della riga carrello (outlet + endpoint)
feature-libs/estimated-delivery-date/root/model/augmented-core.model.ts	modello di augmentation di OrderEntry
projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts	wrapper lazy del carrello: punto in cui inserire le estensioni
feature-libs/checkout/base/root/config/checkout-config.ts	CheckoutConfig.checkout (steps, flows, ...)
feature-libs/checkout/base/root/config/default-checkout-config.ts	i 4 step di default
feature-libs/checkout/base/root/config/default-checkout-routing-config.ts	rotte semantiche degli step
feature-libs/checkout/base/root/model/checkout-step.model.ts	CheckoutStep, CheckoutStepType
feature-libs/checkout/base/root/checkout-root.module.ts	CHECKOUT_BASE_CMS_COMPONENTS, feature checkout
feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts	CheckoutStepsSetGuard (protected isStepSet)
feature-libs/checkout/base/components/services/checkout-step.service.ts	CheckoutStepService (next, back, getBackBntText)
feature-libs/checkout/b2b/root/model/augmented-types.ts	modello di nuovo CheckoutStepType
feature-libs/checkout/b2b/root/config/default-b2b-checkout-config.ts	modello di lista steps ridefinita
feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts	modello di override del guard con useExisting
integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts	modello di sostituzione di un componente di step
projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts	wrapper lazy del checkout
core-libs/styles/scss/_components.scss	$skipComponentStyles, allowlist, @extend %<selector>
feature-libs/cart/base/styles/components/_index.scss	allowlist della lib carrello
feature-libs/cart/base/styles/components/_add-to-cart.scss	placeholder %cx-add-to-cart

PATH	RUOLO
core-libs/styles/scss/theme/santorini/_variables.scss	variabili colore !default e $theme-colors
core-libs/styles/scss/root.scss	$theme-colors → custom property --cx-color-*
projects/storefrontapp/src/styles.scss, projects/storefrontapp/src/styles-config.scss, projects/storefrontapp/src/styles/lib-cart.scss, projects/storefrontapp/src/cms-themes.scss	integrazione degli stili nell'app demo
core-libs/core/src/site-theme/config/site-theme-config.ts	siteTheme.optionalThemes
core-libs/core/src/i18n/config/i18n-config.ts	i18n.resources, i18n.chunks, i18n.backend.loader
core-libs/core/src/i18n/translation-chunk.service.ts	mappa namespace → chunk (costruita una volta)
core-libs/core/src/i18n/i18next/i18next-initializer.ts	addResourceBundle delle risorse statiche all'init
core-libs/core/src/i18n/i18next/i18next-translation.service.ts	chiave chunk:key, fallback in dev/prod
feature-libs/cart/base/assets/translations/translations.ts	modello di TranslationChunksConfig
projects/storefrontapp/src/app/spartacus/spartacus.module.ts	ordine SpartacusFeaturesModule → SpartacusConfigurationModule

3. Come funziona passo per passo

3.0 Convenzioni e catene di export

Tutti gli esempi di questo capitolo sono **esempi illustrativi, non presenti nel repo**. I nomi inventati (feature "loyalty", "gift wrap", componente "pack add to cart", ecc.) sono scritti in corsivo nel testo e compaiono solo nei blocchi di codice. I path nei commenti `// --- src/app/...` indicano dove metterei il file in un'app generata dagli schematic; non esistono nel repo.

Gli import `@spartacus/*` usano solo entry point pubblici. Per ogni simbolo, la catena `public_api` → `index` → definizione è la seguente (verificata con `grep -n`; la ripeto in forma compatta in ogni ricetta).

SIMBOLO	ENTRY POINT	CATENA DI EXPORT
provideConfig	@spartacus/core	core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18
provideDefaultConfig	@spartacus/core	core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L55-L55
CmsConfig	@spartacus/core	core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L8-L8 → core-libs/core/src/cms/config/index.ts:L7-L7 → core-libs/core/src/cms/config/cms-config.ts:L128-L128

SIMBOLO	ENTRY POINT	CATENA DI EXPORT
facadeFactory	@spartacus/core	core-libs/core/public_api.ts:L24-L24 → core-libs/core/src/lazy-loading/index.ts:L12-L12 → core-libs/core/src/lazy-loading/facade-factory/index.ts:L8-L8 → core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L20
I18nConfig	@spartacus/core	core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L7-L7 → core-libs/core/src/i18n/config/i18n-config.ts:L18-L18
TranslatePipe (cxTranslate)	@spartacus/core	core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L15-L15 → core-libs/core/src/i18n/translate.pipe.ts:L29-L29
TranslationChunksConfig, TranslationResources	@spartacus/core	core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L17-L17 → core-libs/core/src/i18n/translation-resources.ts:L16-L16 / :L11-L11
EventService	@spartacus/core	core-libs/core/public_api.ts:L18-L18 → core-libs/core/src/event/index.ts:L8-L8 → core-libs/core/src/event/event.service.ts:L47-L47
RoutingConfig	@spartacus/core	core-libs/core/public_api.ts:L30-L30 → core-libs/core/src/routing/index.ts:L7-L7 → core-libs/core/src/routing/configurable-routes/index.ts:L7-L7 → core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L26-L26
OccConfig	@spartacus/core	core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L12-L12 → core-libs/core/src/occ/config/occ-config.ts:L46-L46
SiteThemeConfig	@spartacus/core	core-libs/core/public_api.ts:L32-L32 → core-libs/core/src/site-theme/index.ts:L7-L7 → core-libs/core/src/site-theme/config/site-theme-config.ts:L15-L15
provideOutlet	@spartacus/storefront	core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L9-L9 → core-libs/storefront/cms-structure/outlet/index.ts:L13-L13 → core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L41
OutletPosition	@spartacus/storefront	core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L9-L9 → core-libs/storefront/cms-structure/outlet/index.ts:L11-L11 → core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L9
CartOutlets	@spartacus/cart/base/root	feature-libs/cart/base/root/public_api.ts:L13-L13 → feature-libs/cart/base/root/models/index.ts:L10-L10 → feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L10
CartItemContext	@spartacus/cart/base/root	feature-libs/cart/base/root/public_api.ts:L13-L13 → feature-libs/cart/base/root/models/index.ts:L9-L9 → feature-libs/cart/base/root/models/cart-item-context.model.ts:L17-L17
OrderEntry	@spartacus/cart/base/root	feature-libs/cart/base/root/public_api.ts:L19-L19 (tipo aumentabile) → feature-libs/cart/base/root/models/cart.model.ts:L152-L152
CartAddEntrySuccessEvent, CartUiEventAddToCart	@spartacus/cart/base/root	feature-libs/cart/base/root/public_api.ts:L10-L10 → feature-libs/cart/base/root/events/index.ts:L9-L9 → feature-libs/cart/base/root/events/cart.events.ts:L66-L66 / :L122-L122
CartBaseModule	@spartacus/cart/base	feature-libs/cart/base/public_api.ts:L7-L7 → feature-libs/cart/base/cart-base.module.ts:L15-L15

SIMBOLO	ENTRY POINT	CATENA DI EXPORT
AddToCartComponent	@spartacus/cart/base/components/add-to-cart	feature-libs/cart/base/components/add-to-cart/public_api.ts:L7-L7 → feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L77-L77
CheckoutConfig	@spartacus/checkout/base/root	feature-libs/checkout/base/root/public_api.ts:L8-L8 → feature-libs/checkout/base/root/config/index.ts:L7-L7 → feature-libs/checkout/base/root/config/checkout-config.ts:L22-L22
CheckoutStep, CheckoutStepType	@spartacus/checkout/base/root	feature-libs/checkout/base/root/public_api.ts:L12-L12 e :L15-L15 (aumentabile) → feature-libs/checkout/base/root/model/index.ts:L9-L9 → feature-libs/checkout/base/root/model/checkout-step.model.ts:L20-L20 / :L7-L7
CHECKOUT_BASE_CMS_COMPONENTS	@spartacus/checkout/base/root	feature-libs/checkout/base/root/public_api.ts:L7-L7 → feature-libs/checkout/base/root/checkout-root.module.ts:L20-L20
CHECKOUT_FEATURE	@spartacus/checkout/base/root	feature-libs/checkout/base/root/public_api.ts:L11-L11 → feature-libs/checkout/base/root/feature-name.ts:L7-L7
CheckoutDeliveryModesFacade	@spartacus/checkout/base/root	feature-libs/checkout/base/root/public_api.ts:L10-L10 → feature-libs/checkout/base/root/facade/index.ts:L9-L9 → feature-libs/checkout/base/root/facade/checkout-delivery-modes.facade.ts:L30-L30
CheckoutModule	@spartacus/checkout/base	feature-libs/checkout/base/public_api.ts:L11-L11 → feature-libs/checkout/base/checkout.module.ts:L15-L15
CheckoutStepService	@spartacus/checkout/base/components	feature-libs/checkout/base/components/public_api.ts:L39-L39 → feature-libs/checkout/base/components/services/index.ts:L9-L9 → feature-libs/checkout/base/components/services/checkout-step.service.ts:L22-L22
CheckoutStepsSetGuard	@spartacus/checkout/base/components	feature-libs/checkout/base/components/public_api.ts:L38-L38 → feature-libs/checkout/base/components/guards/index.ts:L9-L9 → feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L36-L36
CheckoutAuthGuard	@spartacus/checkout/base/components	feature-libs/checkout/base/components/public_api.ts:L38-L38 → feature-libs/checkout/base/components/guards/index.ts:L8-L8 → feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L25-L25
CartNotEmptyGuard	@spartacus/checkout/base/components	feature-libs/checkout/base/components/public_api.ts:L38-L38 → feature-libs/checkout/base/components/guards/index.ts:L7-L7 → feature-libs/checkout/base/components/guards/cart-not-empty.guard.ts:L17-L17

Nota: PROVIDE_OUTLET_OPTIONS (il token dietro provideOutlet) **non** è esportato: core-libs/storefront/cms-structure/outlet/index.ts:L13-L13 esporta solo provideOutlet e ProvideOutletOptions, e il token è marcato @private (core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L10-L15).

3.1 Ricetta A — Feature lib custom lazy-loaded

Quando usarlo

Quando l'app aggiunge una funzionalità di una certa dimensione (qui: un programma fedeltà *loyalty*) che:

- ha componenti CMS propri, che compaiono solo su alcune pagine;
- ha servizi che altri pezzi dell'app (anche eager: header, checkout) vogliono chiamare;
- non deve pesare sul bundle iniziale.

È lo stesso schema delle feature lib ufficiali: root module eager con la sola "dichiarazione" della feature, chunk lazy con l'implementazione, facade proxy nel root che inoltra al chunk. Il modello più compatto nel repo è `@spartacus/cart/saved-cart` (feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L31-L50, projects/storefrontapp/src/app/spartacus/features/cart/cart-saved-cart-feature.module.ts:L21-L31).

Esempio

Esempio illustrativo, non presente nel repo. Più file in un blocco; l'ordine segue il grafo di dipendenze.

TYPESCRIPT

```typescript
// --- src/app/loyalty/root/feature-name.ts
export const LOYALTY_FEATURE = 'loyalty';
export const LOYALTY_CORE_FEATURE = 'loyaltyCore';

// --- src/app/loyalty/root/loyalty.facade.ts  (eager: parte del bundle iniziale)
import { Injectable } from '@angular/core';
import { facadeFactory } from '@spartacus/core';
import { Observable } from 'rxjs';
import { LOYALTY_CORE_FEATURE } from './feature-name';

export interface LoyaltyBalance {
  points: number;
  tier: string;
}

@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: LoyaltyFacade,
      feature: LOYALTY_CORE_FEATURE,
      // SOLO i metodi elencati esistono sul proxy; devono tornare Observable o void
      methods: ['getBalance', 'redeem'],
      // async: true serve solo se il chunk registra uno store NgRx lazy
    }),
})
export abstract class LoyaltyFacade {
  abstract getBalance(): Observable<LoyaltyBalance>;
  abstract redeem(points: number): Observable<unknown>;
}

// --- src/app/loyalty/root/loyalty-root.module.ts  (eager)
import { NgModule } from '@angular/core';
import { CmsConfig, provideDefaultConfig } from '@spartacus/core';
import { LOYALTY_CORE_FEATURE, LOYALTY_FEATURE } from './feature-name';

@NgModule({
  providers: [
    // il root module dichiara COSA copre la feature, non DA DOVE caricarla
    provideDefaultConfig(<CmsConfig>{
```

```
      featureModules: {
        [LOYALTY_FEATURE]: {
          cmsComponents: ['LoyaltyBannerComponent'],
        },
        // alias: la facade punta a "loyaltyCore", che per ora coincide con "loyalty"
        [LOYALTY_CORE_FEATURE]: LOYALTY_FEATURE,
      },
    }),
  ],
})
export class LoyaltyRootModule {}

// --- src/app/loyalty/loyalty.service.ts  (lazy)
import { HttpClient } from '@angular/common/http';
import { Injectable, inject } from '@angular/core';
import { Observable } from 'rxjs';
import { LoyaltyBalance, LoyaltyFacade } from './root/loyalty.facade';

@Injectable() // NON providedIn: 'root' -> vive nell'injector del chunk
export class LoyaltyService implements LoyaltyFacade {
  protected http = inject(HttpClient);

  getBalance(): Observable<LoyaltyBalance> {
    // URL illustrativa: per un endpoint OCC configurabile vedi il capitolo 52
    return this.http.get<LoyaltyBalance>('/api/loyalty/balance');
  }

  redeem(points: number): Observable<unknown> {
    return this.http.post('/api/loyalty/redeem', { points });
  }
}

// --- src/app/loyalty/loyalty-banner.component.ts  (lazy)
import { AsyncPipe } from '@angular/common';
import { Component, inject } from '@angular/core';
import { TranslatePipe } from '@spartacus/core';
import { LoyaltyFacade } from './root/loyalty.facade';

@Component({
  selector: 'app-loyalty-banner',
  imports: [AsyncPipe, TranslatePipe],
  template: `
    @if (balance$ | async; as balance) {
      <p>{{ 'loyalty.balance' | cxTranslate: { points: balance.points } }}</p>
    }
  `,
})
export class LoyaltyBannerComponent {
  protected readonly balance$ = inject(LoyaltyFacade).getBalance();
```

```
}

// --- src/app/loyalty/loyalty.module.ts  (modulo lazy: DEVE essere un NgModule)
import { NgModule } from '@angular/core';
import { CmsConfig, provideDefaultConfig } from '@spartacus/core';
import { LoyaltyBannerComponent } from './loyalty-banner.component';
import { LoyaltyService } from './loyalty.service';
import { LoyaltyFacade } from './root/loyalty.facade';

@NgModule({
  providers: [
    LoyaltyService,
    // la facade astratta, dentro il chunk, risolve al servizio concreto
    { provide: LoyaltyFacade, useExisting: LoyaltyService },
    // mapping CMS -> componente: vive nel chunk, come default config
    provideDefaultConfig(<CmsConfig>{
      cmsComponents: {
        LoyaltyBannerComponent: { component: LoyaltyBannerComponent },
      },
    }),
  ],
})
export class LoyaltyModule {}

// --- src/app/spartacus/features/loyalty-feature.module.ts
// (eager, da aggiungere agli imports di SpartacusFeaturesModule)
import { NgModule } from '@angular/core';
import { CmsConfig, provideConfig } from '@spartacus/core';
import { LOYALTY_FEATURE } from '../../loyalty/root/feature-name';
import { LoyaltyRootModule } from '../../loyalty/root/loyalty-root.module';

@NgModule({
  imports: [LoyaltyRootModule],
  providers: [
    // e' l'app a dire DA DOVE caricare il chunk
    provideConfig(<CmsConfig>{
      featureModules: {
        [LOYALTY_FEATURE]: {
          module: () =>
            import('../../loyalty/loyalty.module').then((m) => m.LoyaltyModule),
        },
      },
    }),
  ],
})
export class LoyaltyFeatureModule {}
```

Catene di export dei simboli `@spartacus/*` usati:

- facadeFactory: core-libs/core/public_api.ts:L24-L24 → core-libs/core/src/lazy-loading/index.ts:L12-L12 → core-libs/core/src/lazy-loading/facade-factory/index.ts:L8-L8 → core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L20.
- CmsConfig: core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L8-L8 → core-libs/core/src/cms/config/index.ts:L7-L7 → core-libs/core/src/cms/config/cms-config.ts:L128-L128.
- provideDefaultConfig, provideConfig: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L55-L55 / :L18-L18.
- TranslatePipe: core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L15-L15 → core-libs/core/src/i18n/translate.pipe.ts:L29-L29.

Perché funziona (file sorgente)

1. FeatureModuleConfig ha tre campi: module, dependencies, cmsComponents; featureModules accetta anche una stringa come alias — core-libs/core/src/cms/config/cms-config.ts:L109-L134.
2. Il root module della lib dichiara cmsComponents e alias ma non module; è l'app a fornire module con provideConfig — modello feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L31-L50 e projects/storefrontapp/src/app/spartacus/features/cart/cart-saved-cart-feature.module.ts:L21-L31.
3. A config stabile CmsFeaturesService associa ogni tipo CMS alla feature, **solo** se la feature ha sia module sia cmsComponents non vuoti — core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L75.
4. Quando la pagina contiene un tipo gestito da feature, CmsComponentsService passa dal resolver della feature — core-libs/storefront/cms-structure/services/cms-components.service.ts:L86-L105 — e FeatureModulesService.resolveFeature risolve alias, dipendenze e istanzia il modulo, con cache shareReplay — core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73.
5. Il valore restituito da module() passa per compileModuleAsync se non è una NgModuleFactory: deve essere un NgModule — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L236.
6. Il mapping CMS è letto dai soli chunk forniti con self: true nell'injector del modulo lazy (i ConfigChunk prevalgono sui DefaultConfigChunk) — core-libs/storefront/cms-structure/services/cms-features.service.ts:L188-L205 — ed è trattato come default sotto la config statica dell'app — core-libs/storefront/cms-structure/services/cms-components.service.ts:L117-L141.
7. facadeFactory delega a FacadeFactoryService.create via inject(), quindi va chiamata in useFactory — core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24. Il proxy è una sottoclasse anonima della facade con i soli methods/properties elencati — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L110-L130.
8. Il resolver del proxy fallisce subito se la feature non ha module; altrimenti prende injector.get(facadeClass) dall'injector del modulo lazy — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57. Lì trova il useExisting verso il servizio concreto — modello feature-libs/cart/saved-cart/core/facade/facade-providers.ts:L11-L17.
9. FacadeDescriptor.async esiste per lo store NgRx lazy ("Required to make lazy NgRx store feature ready") — core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L37; senza store non serve.

Limiti e trappole

- Se l'app non fornisce module, la feature non entra nella mappa dei componenti (warning "No component implementation found") e ogni chiamata alla facade dà l'errore "Feature loyaltyCore is not configured properly" — core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L68, core-

libs/storefront/cms-structure/services/cms-components.service.ts:L170-L176, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47.

- Un metodo non elencato in `methods` non esiste sul proxy (la classe è astratta): `TypeError` alla chiamata. Metodi con ritorno non Observable restituiscono `EMPTY` — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L84-L91, :L119-L126.
- Una chiamata di metodo scarica il chunk anche senza `subscribe` (`connect()` immediato) — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L92. Non usare la facade "per sondare".
- Se il chunk non ri-fornisce la facade con `useExisting`, `injector.get(LoyaltyFacade)` risale al root e trova il proxy stesso. `DA VERIFICARE` a runtime: comportamento dedotto dalla gerarchia degli injector (core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L49-L52), non eseguito.
- `module` deve risolvere a un NgModule: un componente standalone non è un valore valido — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L220-L236.
- Se la feature usa servizi di un'altra feature *non* tramite facade, va dichiarata `dependencies: [ALTRA_FEATURE]`: con una sola dipendenza l'injector padre del chunk è quello della dipendenza — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L97, core-libs/core/src/lazy-loading/feature-modules.service.ts:L108-L123.
- Il tipo CMS (qui *LoyaltyBannerComponent*) deve esistere lato backend come componente CMS con quel `typeCode`. `DA VERIFICARE`: i dati CMS non sono nel repo.
- Le traduzioni della feature vanno fornite eager (vedi Ricetta E2), non nel chunk.

3.2 Ricetta B1 — Nuovo campo sulla riga del carrello (augmentation + `fields` OCC + outlet)

Quando usarlo

Quando il backend restituisce (o può restituire) un attributo in più sulle entry del carrello e lo si vuole mostrare sotto ogni riga, senza sostituire `cx-cart-item`. Qui: un flag *giftWrap*. Il modello reale nel repo è `@spartacus/estimated-delivery-date`, che fa esattamente questo con `arrivalSlots` (feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L19-L37).

Esempio

Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// --- src/app/gift-wrap/augmented-cart.model.ts
import '@spartacus/cart/base/root';

declare module '@spartacus/cart/base/root' {
  interface OrderEntry {
    giftWrap?: boolean;
  }
}

// --- src/app/gift-wrap/gift-wrap-info.component.ts
import { AsyncPipe } from '@angular/common';
import { Component, inject } from '@angular/core';
import { CartItemContext, OrderEntry } from '@spartacus/cart/base/root';
import { TranslatePipe } from '@spartacus/core';
```

```
import { EMPTY, Observable } from 'rxjs';
import './augmented-cart.model';

@Component({
  selector: 'app-gift-wrap-info',
  imports: [AsyncPipe, TranslatePipe],
  template: `
    @if ((entry$ | async)?.giftWrap) {
      <span class="app-gift-wrap">{{ 'giftWrap.label' | cxTranslate }}</span>
    }
  `,
})
export class GiftWrapInfoComponent {
  // { optional: true }: fuori da cx-cart-item il contesto non esiste
  protected readonly entry$: Observable<OrderEntry> =
    inject(CartItemContext, { optional: true })?.item$ ?? EMPTY;
}

// --- src/app/gift-wrap/gift-wrap.module.ts
// (da importare nel wrapper lazy del carrello, DOPO CartBaseModule)
import { NgModule } from '@angular/core';
import { CartOutlets } from '@spartacus/cart/base/root';
import { OccConfig, provideConfig } from '@spartacus/core';
import { OutletPosition, provideOutlet } from '@spartacus/storefront';
import { GiftWrapInfoComponent } from './gift-wrap-info.component';

@NgModule({
  providers: [
    // l'endpoint e' una stringa intera: va ridefinito per intero (qui: + giftWrap)
    provideConfig(<OccConfig>{
      backend: {
        occ: {
          endpoints: {
            cart: 'users/${userId}/carts/${cartId}?
fields=DEFAULT,potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrder
Promotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),giftWrap,basePrice(forma
ttedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),tot
alDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,delivery
Cost(formattedValue),totalTax(formattedValue,
value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),user,saveTime,name,descr
iption',
            // stessa operazione, se servono, per 'carts' e 'createCart'
          },
        },
      },
    }),
    provideOutlet({
      id: CartOutlets.ITEM_DETAILS,
      position: OutletPosition.AFTER,
```

```
      component: GiftWrapInfoComponent,
    }),
  ],
})
export class GiftWrapModule {}

// --- src/app/spartacus/features/cart/cart-base-wrapper.module.ts (modificato)
import { NgModule } from '@angular/core';
import { CartBaseModule } from '@spartacus/cart/base';
import { GiftWrapModule } from '../../../gift-wrap/gift-wrap.module';

@NgModule({
  imports: [CartBaseModule, GiftWrapModule],
})
export class CartBaseWrapperModule {}
```

Catene di export:

- CartItemContext: feature-libs/cart/base/root/public_api.ts:L13-L13 → feature-libs/cart/base/root/models/index.ts:L9-L9 → feature-libs/cart/base/root/models/cart-item-context.model.ts:L17-L17.
- OrderEntry: feature-libs/cart/base/root/public_api.ts:L19-L19 → feature-libs/cart/base/root/models/cart.model.ts:L152-L152.
- CartOutlets: feature-libs/cart/base/root/public_api.ts:L13-L13 → feature-libs/cart/base/root/models/index.ts:L10-L10 → feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L10.
- TranslatePipe: core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L15-L15 → core-libs/core/src/i18n/translate.pipe.ts:L29-L29.
- OccConfig: core-libs/core/public_api.ts:L27-L27 → core-libs/core/src/occ/index.ts:L12-L12 → core-libs/core/src/occ/config/occ-config.ts:L46-L46.
- provideConfig: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18.
- provideOutlet, OutletPosition: core-libs/storefront/public_api.ts:L23-L23 → core-libs/storefront/cms-structure/index.ts:L9-L9 → core-libs/storefront/cms-structure/outlet/index.ts:L13-L13 / :L11-L11 → core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L41 / core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L9.
- CartBaseModule: feature-libs/cart/base/public_api.ts:L7-L7 → feature-libs/cart/base/cart-base.module.ts:L15-L15.

Perché funziona (file sorgente)

1. `OrderEntry` è esportato come tipo aumentabile dal root carrello — feature-libs/cart/base/root/public_api.ts:L18-L20; EDD lo aumenta con module augmentation — feature-libs/estimated-delivery-date/root/model/augmented-core.model.ts:L7-L15.
2. L'endpoint `cart` di default elenca esplicitamente i campi delle entries — feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L9-L19; EDD lo ridefinisce intero aggiungendo `arrivalSlots` — feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L10-L28.
3. `OccCartNormalizer` fa `...entry` su ogni entry: un campo OCC in più arriva nel modello senza converter — feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts:L23-L40.

4. `provideOutlet` è un multi-provider di `PROVIDE_OUTLET_OPTIONS` con posizione di default `AFTER` — core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L34-L47. Nel chunk carrello gli outlet sono registrati da `OutletModule.forChild()` tramite `MODULE_INITIALIZER` — core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88, importato da `CartBaseComponentsModule` — feature-libs/cart/base/components/cart-base-components.module.ts:L30-L30. I `MODULE_INITIALIZER` sono letti con `self: true` dall'injector del modulo lazy — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157; poiché *GiftWrapModule* è importato nello stesso wrapper, il suo `provideOutlet` è nello stesso injector.
5. L'outlet `ITEM_DETAILS` è renderizzato sia in `cx-cart-item` sia in `cx-cart-item-list-row` — feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.html:L37-L37, feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L54-L54 — ed entrambi forniscono `CartItemContext` a livello di componente — feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.ts:L33-L36.
6. Il wrapper lazy dell'app importa `CartBaseModule` seguito dalle estensioni — projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25; la feature `cartBase` punta al wrapper — projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L26-L35.
7. Uso `provideConfig` (e non `provideDefaultConfig` come EDD) perché, nella config portata dai chunk, `ambientConfig` vince su `ambientDefaultConfig` indipendentemente dall'ordine — core-libs/core/src/config/services/configuration.service.ts:L76-L106.

Limiti e trappole

- Gli endpoint con `fields` sono stringhe intere: due estensioni che ridefiniscono `cart` si sovrascrivono (vince l'ultima nel merge), e un aggiornamento dei `fields` di default nelle nuove versioni non viene ereditato — core-libs/core/src/config/utils/deep-merge.ts:L43-L45. Con EDD attivo bisogna unire a mano `arrivalSlots` e `giftWrap`.
- La config root dell'app vince su quella del chunk: se l'app definisce già `backend.occ.endpoints.cart` con `provideConfig` in un modulo eager (RootConfig), l'override nel chunk viene ignorato; una `provideDefaultConfig` eager invece perde contro il chunk — core-libs/core/src/config/services/configuration.service.ts:L92-L99.
- `provideOutlet` in un chunk che **non** importa `OutletModule.forChild()` non viene registrato (initializer letti `self: true`). `DA VERIFICARE` a runtime: dedotto staticamente da core-libs/core/src/lazy-loading/lazy-modules.service.ts:L153-L157 e core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88. Per questo il modulo va nel wrapper carrello, non in una feature lazy separata.
- Il componente in outlet compare solo dopo il caricamento del chunk carrello — core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88.
- Il modello EDD dichiara `@Optional() protected cartItemContext = inject(CartItemContext);` (feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L26-L26): `@Optional()` su un campo non rende opzionale `inject()`, che lancia se il token manca e non è richiesto `optional` (angular:packages/core/src/di/injector_compatibility.ts:L188-L197). Nell'esempio uso `{ optional: true }`.
- Per *inviare* un attributo custom all'add-to-cart non basta l'augmentation: `addEntry` accetta solo `productCode`, `quantity`, `pickupStore` (feature-libs/cart/base/root/facade/active-cart.facade.ts:L100-L104) e il body OCC è costruito in `OccCartEntryAdapter.add` (feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L51-L63). Va sostituito l'adapter nel chunk carrello (vedi ricette OCC). `DA VERIFICARE`: la catena store/effects/connector non è stata riletta per questo capitolo (vedi capitolo 25).

3.3 Ricetta B2 — Sostituire/estendere l'add-to-cart

Quando usarlo

Quando il pulsante "Aggiungi al carrello" deve comportarsi diversamente: arrotondare la quantità a confezioni (*pack*), aggiungere un controllo, cambiare markup. Si sostituisce il componente mappato sul tipo CMS `ProductAddToCartComponent`, riusando la logica di `AddToCartComponent` per ereditarietà.

Esempio

Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// --- src/app/cart/pack-add-to-cart.component.ts
import { ChangeDetectionStrategy, Component, NgModule } from '@angular/core';
import { ReactiveFormsModule } from '@angular/forms';
import { AddToCartComponent } from '@spartacus/cart/base/components/add-to-cart';
import { CmsConfig, TranslatePipe, provideConfig } from '@spartacus/core';

const PACK_SIZE = 6;

@Component({
  // stesso selector dell'originale: continua a ricevere gli stili globali %cx-add-to-cart
  selector: 'cx-add-to-cart',
  changeDetection: ChangeDetectionStrategy.OnPush,
  imports: [ReactiveFormsModule, TranslatePipe],
  template: `
    @if (productCode) {
      <form [formGroup]="addToCartForm" (submit)="addToCart()">
        <label>
          {{ 'addToCart.quantityFull' | cxTranslate }}
          <input type="number" min="1" formControlName="quantity" />
        </label>
        <button
          #addToCartDialogTriggerEl
          class="btn btn-primary btn-block"
          type="submit"
          [disabled]="!hasStock"
        >
          {{ 'addToCart.addToCart' | cxTranslate }}
        </button>
      </form>
    }
  `,
})
export class PackAddToCartComponent extends AddToCartComponent {
  // nessun costruttore: Angular eredita la factory DI di AddToCartComponent
  override addToCart(): void {
    const control = this.addToCartForm.get('quantity');
    const requested = Number(control?.value) || 0;
    if (requested > 0) {
      control?.setValue(Math.ceil(requested / PACK_SIZE) * PACK_SIZE);
```

```
    }
    super.addToCart();
  }
}

// --- src/app/cart/pack-add-to-cart.module.ts (eager, importato dall'app)
import { NgModule } from '@angular/core';
import { CmsConfig, provideConfig } from '@spartacus/core';
import { PackAddToCartComponent } from './pack-add-to-cart.component';

@NgModule({
  providers: [
    provideConfig(<CmsConfig>{
      cmsComponents: {
        // config statica dell'app: fusa SOPRA il mapping della feature addToCart
        ProductAddToCartComponent: { component: PackAddToCartComponent },
      },
    }),
  ],
})
export class PackAddToCartModule {}
```

Catene di export:

- AddToCartComponent: feature-libs/cart/base/components/add-to-cart/public_api.ts:L7-L7 → feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L77-L77.
- CmsConfig: core-libs/core/public_api.ts:L14-L14 → core-libs/core/src/cms/index.ts:L8-L8 → core-libs/core/src/cms/config/index.ts:L7-L7 → core-libs/core/src/cms/config/cms-config.ts:L128-L128.
- TranslatePipe: core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L15-L15 → core-libs/core/src/i18n/translate.pipe.ts:L29-L29.
- provideConfig: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18.

Perché funziona (file sorgente)

1. `ProductAddToCartComponent` è mappato su `AddToCartComponent` come default config dentro `AddToCartModule` (con `data.inventoryDisplay: false`) — feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts:L34-L44 — e appartiene alla feature `addToCart` con chunk separato — feature-libs/cart/base/root/cart-base-root.module.ts:L42-L44, projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L46-L55.
2. Per un tipo gestito da feature, il mapping effettivo è `deepMerge({}, mappingFeature, configStatica)`: la config dell'app sovrascrive `component` e lascia gli altri campi — core-libs/storefront/cms-structure/services/cms-components.service.ts:L117-L141. La config statica è uno snapshot preso a config stabile — core-libs/storefront/cms-structure/services/cms-components.service.ts:L61-L68.
3. `addToCartForm`, `hasStock`, `productCode` e `addToCart()` sono pubblici — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L78-L113, :L254-L303; `addToCart()` legge la quantità dal form, chiama `getEntries()` (garantisce il caricamento della lib), poi `addEntry` e dispatcha `CartUiEventAddToCart` — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L278-L302.

4. Il riferimento `#addToCartDialogTriggerEl` serve al refocus dopo il dialog: è letto da `@ViewChild` e copiato nell'evento da `createCartUiEventAddToCart` (protected) — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L90-L93, :L305-L318.
5. Gli stili Spartacus sono globali e legati al selector: `AddToCartComponent` non dichiara `styleUrls` — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L61-L76; la lib carrello estende `%cx-add-to-cart` sul selettore `cx-add-to-cart` — feature-libs/cart/base/styles/components/_index.scss:L19-L35.

Limiti e trappole

- Il chunk della feature `addToCart` viene caricato comunque (il tipo è gestito da feature) e `data.inventoryDisplay` del default resta nel mapping — core-libs/storefront/cms-structure/services/cms-components.service.ts:L92-L99, :L126-L133.
- Importare `@spartacus/cart/base/components/add-to-cart` da un modulo eager porta `AddToCartComponent` nel bundle iniziale. DA VERIFICARE: effetto sul bundle non misurato (build non eseguibile). In alternativa il mapping può vivere in un modulo lazy importato nel chunk della feature (stesso schema della Ricetta C2).
- Un selector diverso (es. `app-pack-add-to-cart`) **non** riceve gli stili di `cx-add-to-cart` — feature-libs/cart/base/styles/components/_index.scss:L28-L35.
- Il template originale ha in più outlet (`ADD_TO_CART_PICKUP_OPTION`), stock e icone — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.html:L1-L71: riscrivendo il template quelle parti vanno reintrodotte se servono.
- L'ereditarietà lega il componente ai dettagli interni di `AddToCartComponent` (campi pubblici non documentati come API stabile): a ogni upgrade va riletto il sorgente (vedi capitolo 54).
- `addEntry` è `void` e non accetta campi extra — feature-libs/cart/base/root/facade/active-cart.facade.ts:L100-L104.

3.4 Ricetta B3 — Reagire agli eventi del carrello

Quando usarlo

Per tracking, notifiche, ricarico di dati correlati (come fa la lib quote) quando un prodotto viene aggiunto al carrello, senza toccare i componenti.

Esempio

Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// --- src/app/cart/add-to-cart-analytics.ts
import { Injectable, NgModule, OnDestroy, inject } from '@angular/core';
import {
  CartAddEntrySuccessEvent,
  CartUiEventAddToCart,
} from '@spartacus/cart/base/root';
import { EventService } from '@spartacus/core';
import { Subscription } from 'rxjs';

@Injectable({ providedIn: 'root' })
export class AddToCartAnalyticsListener implements OnDestroy {
  protected eventService = inject(EventService);
  protected subscription = new Subscription();
```

```
  constructor() {
    // intenzione dell'utente (UI): parte dal componente add-to-cart
    this.subscription.add(
      this.eventService
        .get(CartUiEventAddToCart)
        .subscribe((event) => this.track('add_click', event.productCode, event.quantity))
    );
    // esito lato backend: mappato dall'action NgRx nel chunk carrello
    this.subscription.add(
      this.eventService
        .get(CartAddEntrySuccessEvent)
        .subscribe((event) =>
          this.track('add_success', event.productCode, event.quantityAdded ?? event.quantity)
        )
    );
  }

  protected track(name: string, productCode: string, quantity: number): void {
    // invio verso il proprio sistema di analytics
  }

  ngOnDestroy(): void {
    this.subscription.unsubscribe();
  }
}

// istanzia il listener all'avvio (stesso schema di CartBaseEventModule)
@NgModule({})
export class AddToCartAnalyticsModule {
  constructor(_listener: AddToCartAnalyticsListener) {
    // Intentional empty constructor
  }
}
```

Catene di export:

- CartAddEntrySuccessEvent, CartUiEventAddToCart: feature-libs/cart/base/root/public_api.ts:L10-L10 → feature-libs/cart/base/root/events/index.ts:L9-L9 → feature-libs/cart/base/root/events/cart.events.ts:L66-L66 / :L122-L122.
- EventService: core-libs/core/public_api.ts:L18-L18 → core-libs/core/src/event/index.ts:L8-L8 → core-libs/core/src/event/event.service.ts:L47-L47.

Perché funziona (file sorgente)

1. EventService espone register, get, dispatch — core-libs/core/src/event/event.service.ts:L45-L47, :L68-L68, :L90-L90, :L105-L105.
2. CartAddEntrySuccessEvent porta productCode, quantity, entry?, quantityAdded?, deliveryModeChanged? — feature-libs/cart/base/root/events/cart.events.ts:L66-L76 — ed è mappato da

`CART_ADD_ENTRY_SUCCESS` nel `CartEventBuilder` del core lazy — feature-libs/cart/base/core/event/cart-event.builder.ts:L47-L87.

3. `CartUiEventAddToCart` è dispatchato da `AddToCartComponent.addToCart()` — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L294-L301 — ed estende `CxEvent`, non `CartEvent` — feature-libs/cart/base/root/events/cart.events.ts:L122-L136.
4. Modello di listener root: `QuoteCartEventListener` — feature-libs/quote/core/event/quote-cart-event.listener.ts:L18-L45; modello di istanziazione con NgModule a costruttore vuoto — feature-libs/cart/base/root/events/cart-base-event.module.ts:L10-L15, feature-libs/quote/core/quote-core.module.ts:L21-L24.
5. Alternativa senza codice per i tag manager: `tagManager.gtm.events` — projects/storefrontapp/src/app/spartacus/features/tracking/personalization-feature.module.ts:L30-L39.

Limiti e trappole

- Gli eventi `CartAddEntry*` sono registrati dal `CartEventBuilder` del chunk carrello: prima del caricamento non esiste sorgente; `dispatch` usa un `Subject` senza replay — feature-libs/cart/base/core/cart-base-core.module.ts:L22-L28, core-libs/core/src/event/event.service.ts:L111-L112, :L123-L125.
- `CartUiEventAddToCart` non estende `CartEvent`: un listener su `CartEvent` non lo riceve — feature-libs/cart/base/root/events/cart.events.ts:L122-L122.
- `CartUiEventAddToCart` è emesso solo dal componente standard (e dalle sottoclassi che chiamano `super.addToCart()`): un add-to-cart fatto chiamando direttamente `ActiveCartFacade.addEntry` produce solo gli eventi `CartAddEntry*` — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L278-L302.
- Approfondimento sul modello eventi: capitolo 20; ricette generiche sugli eventi: capitolo 50.

3.5 Ricetta C1 — Nuovo step di checkout

Quando usarlo

Quando il checkout deve chiedere qualcosa in più in una pagina dedicata, con un posto nella progress bar e con il blocco "non puoi andare avanti se non hai completato lo step". Qui: uno step *loyalty points* tra modalità di consegna e pagamento. Il modello reale è lo step "payment type" del checkout B2B.

Esempio

Esempio illustrativo, non presente nel repo. Parte eager (tipo, step, rotta, componenti della feature) e parte lazy (componente, guard).

```typescript
// --- src/app/checkout-loyalty/augmented-checkout.model.ts
import { CheckoutStepType } from '@spartacus/checkout/base/root';

declare module '@spartacus/checkout/base/root' {
  enum CheckoutStepType {
    LOYALTY_POINTS = 'loyaltyPoints',
  }
}
// senza questa riga CheckoutStepType.LOYALTY_POINTS e' undefined a runtime
(CheckoutStepType as any)['LOYALTY_POINTS'] = 'loyaltyPoints';

// --- src/app/checkout-loyalty/loyalty-checkout-root.module.ts  (eager)
```

```
// Import augmented types first (come il root B2B)
import './augmented-checkout.model';
import { NgModule } from '@angular/core';
import {
  CHECKOUT_BASE_CMS_COMPONENTS,
  CHECKOUT_FEATURE,
  CheckoutConfig,
  CheckoutStepType,
} from '@spartacus/checkout/base/root';
import { CmsConfig, RoutingConfig, provideConfig } from '@spartacus/core';

@NgModule({
  providers: [
    // array: va ridefinita TUTTA la lista
    provideConfig(<CheckoutConfig>{
      checkout: {
        steps: [
          { id: 'deliveryAddress', name: 'checkoutProgress.deliveryAddress', routeName:
'checkoutDeliveryAddress', type: [CheckoutStepType.DELIVERY_ADDRESS] },
          { id: 'deliveryMode', name: 'checkoutProgress.deliveryMode', routeName:
'checkoutDeliveryMode', type: [CheckoutStepType.DELIVERY_MODE] },
          { id: 'loyaltyPoints', name: 'loyaltyCheckout.stepName', routeName: 'checkoutLoyaltyPoints',
type: [CheckoutStepType.LOYALTY_POINTS] },
          { id: 'paymentDetails', name: 'checkoutProgress.paymentDetails', routeName:
'checkoutPaymentDetails', type: [CheckoutStepType.PAYMENT_DETAILS] },
          { id: 'reviewOrder', name: 'checkoutProgress.reviewOrder', routeName: 'checkoutReviewOrder',
type: [CheckoutStepType.REVIEW_ORDER] },
        ],
      },
    }),
    provideConfig(<RoutingConfig>{
      routing: {
        routes: { checkoutLoyaltyPoints: { paths: ['checkout/loyalty-points'] } },
      },
    }),
    // il nuovo tipo CMS deve far parte della feature "checkout" (array: rispalmare la base)
    provideConfig(<CmsConfig>{
      featureModules: {
        [CHECKOUT_FEATURE]: {
          cmsComponents: [...CHECKOUT_BASE_CMS_COMPONENTS, 'CheckoutLoyaltyPoints'],
        },
      },
    }),
  ],
})
export class LoyaltyCheckoutRootModule {}

// --- src/app/checkout-loyalty/loyalty-choice.service.ts
import { Injectable } from '@angular/core';
```

```
import { BehaviorSubject } from 'rxjs';

@Injectable({ providedIn: 'root' })
export class LoyaltyChoiceService {
  // undefined = step non ancora completato
  readonly pointsToUse$ = new BehaviorSubject<number | undefined>(undefined);
}

// --- src/app/checkout-loyalty/checkout-loyalty-points.component.ts  (lazy)
import { Component, inject } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { CheckoutStepService } from '@spartacus/checkout/base/components';
import { TranslatePipe } from '@spartacus/core';
import { LoyaltyChoiceService } from './loyalty-choice.service';

@Component({
  selector: 'app-checkout-loyalty-points',
  imports: [TranslatePipe],
  template: `
    <h2>{{ 'loyaltyCheckout.stepName' | cxTranslate }}</h2>
    <button class="btn btn-secondary" (click)="back()">{{ backBtnText | cxTranslate }}</button>
    <button class="btn btn-primary" (click)="usePoints(0)">{{ 'loyaltyCheckout.skip' | cxTranslate }}
</button>
  `,
})
export class CheckoutLoyaltyPointsComponent {
  protected activatedRoute = inject(ActivatedRoute);
  protected checkoutStepService = inject(CheckoutStepService);
  protected loyaltyChoice = inject(LoyaltyChoiceService);

  backBtnText = this.checkoutStepService.getBackBntText(this.activatedRoute);

  usePoints(points: number): void {
    this.loyaltyChoice.pointsToUse$.next(points);
    this.checkoutStepService.next(this.activatedRoute);
  }

  back(): void {
    this.checkoutStepService.back(this.activatedRoute);
  }
}

// --- src/app/checkout-loyalty/loyalty-steps-set.guard.ts  (lazy)
import { Injectable, inject } from '@angular/core';
import { GuardResult } from '@angular/router';
import { CheckoutStepsSetGuard } from '@spartacus/checkout/base/components';
import { CheckoutStep, CheckoutStepType } from '@spartacus/checkout/base/root';
import { Observable } from 'rxjs';
import { map, take } from 'rxjs/operators';
```

```
import { LoyaltyChoiceService } from './loyalty-choice.service';

@Injectable({ providedIn: 'root' })
export class LoyaltyStepsSetGuard extends CheckoutStepsSetGuard {
  protected loyaltyChoice = inject(LoyaltyChoiceService);

  protected override isStepSet(step: CheckoutStep): Observable<GuardResult> {
    if (step && !step.disabled && step.type[0] === CheckoutStepType.LOYALTY_POINTS) {
      return this.loyaltyChoice.pointsToUse$.pipe(
        take(1),
        map((points) => (points !== undefined ? true : this.getUrl(step.routeName)))
      );
    }
    return super.isStepSet(step);
  }
}

// --- src/app/checkout-loyalty/loyalty-checkout.module.ts
// (lazy: da importare in CheckoutWrapperModule DOPO CheckoutModule)
import { NgModule } from '@angular/core';
import {
  CartNotEmptyGuard,
  CheckoutAuthGuard,
  CheckoutStepsSetGuard,
} from '@spartacus/checkout/base/components';
import { CmsConfig, provideDefaultConfig } from '@spartacus/core';
import { CheckoutLoyaltyPointsComponent } from './checkout-loyalty-points.component';
import { LoyaltyStepsSetGuard } from './loyalty-steps-set.guard';

@NgModule({
  providers: [
    provideDefaultConfig(<CmsConfig>{
      cmsComponents: {
        CheckoutLoyaltyPoints: {
          component: CheckoutLoyaltyPointsComponent,
          guards: [CheckoutAuthGuard, CartNotEmptyGuard],
        },
      },
    }),
    { provide: CheckoutStepsSetGuard, useExisting: LoyaltyStepsSetGuard },
  ],
})
export class LoyaltyCheckoutModule {}
```

Catene di export:

- CheckoutStepType, CheckoutStep: feature-libs/checkout/base/root/public_api.ts:L12-L12 e :L15-L15 → feature-libs/checkout/base/root/model/index.ts:L9-L9 → feature-libs/checkout/base/root/model/checkout-step.model.ts:L7-L7 / :L20-L20.

- CheckoutConfig: feature-libs/checkout/base/root/public_api.ts:L8-L8 → feature-libs/checkout/base/root/config/index.ts:L7-L7 → feature-libs/checkout/base/root/config/checkout-config.ts:L22-L22.
- CHECKOUT_BASE_CMS_COMPONENTS: feature-libs/checkout/base/root/public_api.ts:L7-L7 → feature-libs/checkout/base/root/checkout-root.module.ts:L20-L20.
- CHECKOUT_FEATURE: feature-libs/checkout/base/root/public_api.ts:L11-L11 → feature-libs/checkout/base/root/feature-name.ts:L7-L7.
- CmsConfig, RoutingConfig, provideConfig, provideDefaultConfig, TranslatePipe: vedi tabella §3.0 (core-libs/core/src/cms/config/cms-config.ts:L128-L128, core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L26-L26, core-libs/core/src/config/config-providers.ts:L18-L18, :L55-L55, core-libs/core/src/i18n/translate.pipe.ts:L29-L29).
- CheckoutStepService: feature-libs/checkout/base/components/public_api.ts:L39-L39 → feature-libs/checkout/base/components/services/index.ts:L9-L9 → feature-libs/checkout/base/components/services/checkout-step.service.ts:L22-L22.
- CheckoutStepsSetGuard, CheckoutAuthGuard, CartNotEmptyGuard: feature-libs/checkout/base/components/public_api.ts:L38-L38 → feature-libs/checkout/base/components/guards/index.ts:L9-L9 / :L8-L8 / :L7-L7 → feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L36-L36, feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L25-L25, feature-libs/checkout/base/components/guards/cart-not-empty.guard.ts:L17-L17.

Perché funziona (file sorgente)

1. CheckoutStep = id, name (chiave i18n), routeName, type, disabled?, nameMultiLine? — feature-libs/checkout/base/root/model/checkout-step.model.ts:L20-L27; CheckoutStepType ha 4 valori — feature-libs/checkout/base/root/model/checkout-step.model.ts:L7-L12 — ed è esportato come tipo aumentabile — feature-libs/checkout/base/root/public_api.ts:L14-L17.
2. Il B2B aumenta l'enum e assegna il valore runtime — feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23 — importando il file per primo nel root module — feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L8-L9.
3. Il B2B ridefinisce l'intera lista steps — feature-libs/checkout/b2b/root/config/default-b2b-checkout-config.ts:L13-L51 — e aggiunge la rotta del nuovo step — feature-libs/checkout/b2b/root/config/default-checkout-b2b-routing-config.ts:L9-L15.
4. CheckoutStepService costruisce gli step da getCheckoutFlow()?.steps, scartando i disabled — feature-libs/checkout/base/components/services/checkout-step.service.ts:L88-L100; next/back navigano allo step adiacente e getBackBntText restituisce la chiave del pulsante indietro — feature-libs/checkout/base/components/services/checkout-step.service.ts:L65-L86.
5. La progress bar traduce step.name con cxTranslate — feature-libs/checkout/base/components/checkout-progress/checkout-progress.component.html:L27-L29.
6. Il nuovo tipo CMS deve stare nella feature checkout (che dipende da cartBase) — feature-libs/checkout/base/root/checkout-root.module.ts:L20-L49; il B2B lo fa rispalmando la lista base — feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L23-L42.
7. Il componente del nuovo step B2B è mappato con guards — feature-libs/checkout/b2b/components/checkout-payment-type/checkout-payment-type.module.ts:L28-L35. I guard dei componenti CMS sono istanze ottenute dall'UnifiedInjector e filtrate con isCanActivate — core-libs/storefront/cms-structure/services/cms-guards.service.ts:L36-L48, core-libs/storefront/cms-structure/services/utils.ts:L9-L11.
8. CheckoutStepsSetGuard.canActivate individua lo step corrente via paths[0] e verifica solo lo step precedente con isStepSet — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L73-L104; isStepSet è protected e per un tipo sconosciuto ritorna of(true) — feature-libs/

checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132; `getUrl` è `protected` — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L168-L172.

9. Il B2B sostituisce il guard con una sottoclasse registrata con `useExisting` nel modulo lazy — feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37.
10. La URL `checkout/loyalty-points` è servita dalla rotta CMS `**` (`CmsPageGuard` + `PageLayoutComponent`) — core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25.

Limiti e trappole

- Senza estendere il guard, `isStepSet` ritorna `true` per il tipo sconosciuto: lo step successivo è raggiungibile anche se il nuovo non è completato — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132.
- **Con il B2B attivo** il guard effettivo è `CheckoutB2BStepsSetGuard`, che ridefinisce `canActivate` e chiama `isB2BStepSet`, non `isStepSet` — feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L57-L86, :L99-L122: in quel caso si estende il guard B2B (e lo si registra al posto di entrambi).
- `CheckoutStepsSetGuard` è agganciato ai componenti di progress (`CheckoutProgress`, mobile), non ai componenti di step — feature-libs/checkout/base/components/checkout-progress/checkout-progress.module.ts:L37-L37. `DA VERIFICARE`: la pagina CMS dello step (ContentPage con label `/checkout/loyalty-points`, slot con *CheckoutLoyaltyPoints* e `CheckoutProgress`) va creata nel backend; i dati CMS non sono nel repo.
- Se la URL della pagina non coincide con `paths[0]` della `routeName` (o lo step è `disabled`) il guard reindirizza a `checkout` con warning in dev — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L81-L101.
- `checkout.steps` e `featureModules.checkout.cmsComponents` sono array: ridefinendoli si sostituisce la lista intera — core-libs/core/src/config/utils/deep-merge.ts:L43-L45. Con il B2B la lista base degli step è diversa (`paymentType` in testa) e i suoi tipi CMS in più vanno inclusi.
- `CheckoutStepService` legge gli step nel costruttore — feature-libs/checkout/base/components/services/checkout-step.service.ts:L57-L63: la config degli step deve essere eager (root), come fa il B2B nel root module.
- Le guard funzionali (`CanActivateFn`) vengono scartate da `isCanActivate` senza errore — core-libs/storefront/cms-structure/services/utils.ts:L9-L11.
- Il *LoyaltyChoiceService* dell'esempio tiene lo stato in memoria: dopo un refresh lo step risulta non completato. Uno stato reale va letto dal backend (attributo del carrello). Dettagli sul checkout: capitolo 26.

3.6 Ricetta C2 — Sostituire il componente di uno step

Quando usarlo

Quando lo step esiste già ma la sua UI va rifatta (qui: la scelta della modalità di consegna mostrata a "tile"). Si rimappa il tipo CMS dello step (`CheckoutDeliveryMode`) su un componente proprio, dentro il chunk checkout, come fa la lib digital-payments con `CheckoutPaymentDetails`.

Esempio

Esempio illustrativo, non presente nel repo.

```typescript
// --- src/app/checkout/delivery-mode-tiles.ts
import { AsyncPipe } from '@angular/common';
import { Component, NgModule, inject } from '@angular/core';
```

```
import { ActivatedRoute } from '@angular/router';
import { CheckoutStepService } from '@spartacus/checkout/base/components';
import { CheckoutDeliveryModesFacade } from '@spartacus/checkout/base/root';
import { CmsConfig, TranslatePipe, provideConfig } from '@spartacus/core';

@Component({
  selector: 'app-delivery-mode-tiles',
  imports: [AsyncPipe, TranslatePipe],
  template: `
    @for (mode of modes$ | async; track mode.code) {
      <button class="btn btn-secondary" (click)="select(mode.code)">{{ mode.name }}</button>
    }
    <button class="btn btn-secondary" (click)="back()">{{ backBtnText | cxTranslate }}</button>
    <button class="btn btn-primary" (click)="next()">{{ 'common.continue' | cxTranslate }}</button>
  `,
})
export class DeliveryModeTilesComponent {
  protected activatedRoute = inject(ActivatedRoute);
  protected checkoutStepService = inject(CheckoutStepService);
  protected deliveryModesFacade = inject(CheckoutDeliveryModesFacade);

  readonly modes$ = this.deliveryModesFacade.getSupportedDeliveryModes();
  backBtnText = this.checkoutStepService.getBackBntText(this.activatedRoute);

  select(code: string | undefined): void {
    if (code) {
      this.deliveryModesFacade.setDeliveryMode(code).subscribe();
    }
  }

  next(): void {
    this.checkoutStepService.next(this.activatedRoute);
  }

  back(): void {
    this.checkoutStepService.back(this.activatedRoute);
  }
}

// lazy: da importare in CheckoutWrapperModule DOPO CheckoutModule
@NgModule({
  providers: [
    provideConfig(<CmsConfig>{
      cmsComponents: {
        // solo `component`: guards e data del default restano (deepMerge)
        CheckoutDeliveryMode: { component: DeliveryModeTilesComponent },
      },
    }),
  ],
```

```
})
export class DeliveryModeTilesModule {}

// --- src/app/spartacus/features/checkout/checkout-wrapper.module.ts (modificato)
// @NgModule({ imports: [CheckoutModule, DeliveryModeTilesModule] })
// export class CheckoutWrapperModule {}
```

Catene di export:

- CheckoutStepService: feature-libs/checkout/base/components/public_api.ts:L39-L39 → feature-libs/checkout/base/components/services/index.ts:L9-L9 → feature-libs/checkout/base/components/services/checkout-step.service.ts:L22-L22.
- CheckoutDeliveryModesFacade: feature-libs/checkout/base/root/public_api.ts:L10-L10 → feature-libs/checkout/base/root/facade/index.ts:L9-L9 → feature-libs/checkout/base/root/facade/checkout-delivery-modes.facade.ts:L30-L30.
- CmsConfig, TranslatePipe, provideConfig: core-libs/core/public_api.ts:L14-L14 / :L23-L23 / :L15-L15 → core-libs/core/src/cms/config/cms-config.ts:L128-L128, core-libs/core/src/i18n/translate.pipe.ts:L29-L29, core-libs/core/src/config/config-providers.ts:L18-L18 (catene complete in §3.0).
- CheckoutModule (nel wrapper): feature-libs/checkout/base/public_api.ts:L11-L11 → feature-libs/checkout/base/checkout.module.ts:L15-L15.

Perché funziona (file sorgente)

1. Il mapping di default di CheckoutDeliveryMode è default config nel chunk, con data.composition.inner e guards: [CheckoutAuthGuard, CartNotEmptyGuard] — feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.module.ts:L36-L50.
2. La config CMS della feature è deepMerge(defaultChunk..., configChunk...) dei chunk self: true del modulo lazy: il provideConfig del modulo importato nel wrapper prevale sul default — core-libs/storefront/cms-structure/services/cms-features.service.ts:L188-L205; deepMerge fonde l'oggetto mapping chiave per chiave — core-libs/core/src/config/utils/deep-merge.ts:L36-L45.
3. Modello reale: digital-payments rimappa CheckoutPaymentDetails con ConfigModule.withConfig e solo component — integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L32-L45 — mentre i guards restano quelli del default — feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-method.module.ts:L36-L38.
4. Il wrapper checkout dell'app importa CheckoutModule e poi le estensioni — projects/storefrontapp/src/app/spartacus/features/checkout/checkout-wrapper.module.ts:L16-L33.
5. CheckoutDeliveryModesFacade è una facade proxy sulla feature checkoutCore — feature-libs/checkout/base/root/facade/checkout-delivery-modes.facade.ts:L13-L55; il componente standard usa lo stesso schema next/back con ActivatedRoute — feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.ts:L102-L102, :L151-L157.

Limiti e trappole

- data.composition.inner: ['PickupInStoreDeliveryModeComponent'] del default resta nel mapping: se il componente custom non rende i figli (cx-page-component), il blocco pickup sparisce senza errori — feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.module.ts:L41-L45.
- Ridefinire guards sostituisce l'array: per aggiungere un guard vanno ripetuti anche quelli di default — core-libs/core/src/config/utils/deep-merge.ts:L43-L45.

- Il componente standard gestisce stato di caricamento, preselezione, focus da tastiera e filtro di `pickup` — feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.ts:L95-L160: in una riscrittura vanno rifatti (vedi accessibilità nel capitolo 38 e capitolo 26).
- Ordine nel wrapper: l'estensione va dopo `CheckoutModule`; con `provideDefaultConfig` sarebbe l'ordine dei chunk a decidere — core-libs/storefront/cms-structure/services/cms-features.service.ts:L200-L204, angular:packages/core/src/di/provider_collection.ts:L293-L353.
- Se lo stesso tipo è rimappato anche nella config statica dell'app, vince la config statica — core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L133.

3.7 Ricetta D1 — Escludere e riscrivere lo stile di un componente (SCSS)

Quando usarlo

Quando lo stile standard di un componente Spartacus è troppo distante da quello desiderato e "lottare" con la specificità costa più che ripartire da zero. Qui: `cx-add-to-cart`. Il meccanismo è la lista Sass `$skipComponentStyles`.

Esempio

Esempio illustrativo, non presente nel repo (l'"esempio" degli stili è SCSS). Due file dell'app, sullo schema dell'app demo.

```scss
// --- src/styles-config.scss  (importato in testa a OGNI bundle di stile)
// Always bump to the latest (breaking) styles during development.
$useLatestStyles: true;

// D1: niente stili standard (placeholder %cx-add-to-cart) per cx-add-to-cart, nel bundle core e in quelli delle lib;
// restano le regole annidate in contenitori/layout (vedi Limiti e trappole).
// Va assegnata PRIMA degli import di Spartacus: le variabili della lib sono `!default`.
$skipComponentStyles: (cx-add-to-cart);

// --- src/styles/spartacus/cart.scss  (bundle della lib cart: generato dallo schematic)
@import '../../styles-config';
@import '@spartacus/cart';

// Le regole proprie stanno nello stesso bundle della lib, dopo l'import:
// qui sono disponibili mixin e variabili di @spartacus/styles/scss/core.
cx-add-to-cart {
  .quantity {
    margin: 0 0 1rem;

    label {
      @include type('8');
    }
  }

  button[type='submit'] {
    border-radius: 0;
    background-color: var(--cx-color-primary);
```

```
    color: var(--cx-color-inverse);
  }
}

// Variante "additiva" invece dello skip: aggiungere regole al placeholder,
// nello STESSO bundle in cui avviene l'@extend (senza $skipComponentStyles).
// %cx-add-to-cart {
//   .info { font-weight: var(--cx-font-weight-bold); }
// }
```

Catene di export (Sass): `@spartacus/cart` → feature-libs/cart/_index.scss:L1-L17 (importa `@spartacus/styles/scss/core` e `./base/index`) → feature-libs/cart/base/styles/components/_index.scss:L19-L35 (allowlist, `$skipComponentStyles`, `@extend`) → placeholder feature-libs/cart/base/styles/components/_add-to-cart.scss:L1-L1. `@spartacus/styles/scss/core` → core-libs/styles/scss/_core.scss:L1-L5.

Perché funziona (file sorgente)

1. I selettori dei componenti sono generati con `@extend %<selector>` per ogni voce dell'allowlist, saltando quelli presenti in `$skipComponentStyles` — core-libs/styles/scss/_components.scss:L37-L53; `$skipComponentStyles: () !default` — core-libs/styles/scss/_components.scss:L4-L7.
2. La lib carrello replica lo schema con `$cart-components-allowlist` (contiene `cx-add-to-cart`) e lo stesso `$skipComponentStyles` — feature-libs/cart/base/styles/components/_index.scss:L19-L44.
3. Lo stile del componente è un placeholder Sass, quindi non emette CSS se nessuno lo estende — feature-libs/cart/base/styles/components/_add-to-cart.scss:L1-L1.
4. L'app demo importa `styles-config` in testa sia al bundle principale — projects/storefrontapp/src/styles.scss:L1-L4 — sia ai bundle delle lib — projects/storefrontapp/src/styles/lib-cart.scss:L1-L2; ogni lib è un bundle separato in `project.json` — projects/storefrontapp/project.json:L85-L87.
5. Lo schematic genera `src/styles/spartacus/<lib>.scss` con l'import relativo di `styles-config` seguito dalla lib — core-libs/schematics/src/shared/utils/lib-utils.ts:L641-L698.
6. `@spartacus/cart` importa `scss/core` (mixin come `type()`) prima dei componenti — feature-libs/cart/_index.scss:L1-L13; `scss/core` non emette CSS di componenti — core-libs/styles/scss/_core.scss:L1-L5.

Limiti e trappole

- `$skipComponentStyles` è `!default` in ogni lib: assegnato dopo l'import non ha effetto; e poiché ogni lib è un bundle Sass separato, va messo in `styles-config.scss` — core-libs/styles/scss/_components.scss:L1-L19, feature-libs/cart/base/styles/components/_index.scss:L26-L26.
- Lo skip agisce solo sui selettori in allowlist con placeholder: stili scritti come selettori diretti (es. EDD) non si disattivano così — feature-libs/estimated-delivery-date/styles/_estimated-delivery-date.scss:L1-L5.
- Il loop su `body` estende `%<selector>__body` con `!optional`; per `cx-add-to-cart` quel placeholder non esiste nel repo, quindi lo skip non ha effetti su `body` — feature-libs/cart/base/styles/components/_index.scss:L38-L43.
- Restano le regole su `cx-add-to-cart` scritte nello stile di componenti contenitori e dei layout (es. `cx-product-summary`, `ProductDetailsPageTemplate`): lo skip non le tocca — core-libs/styles/scss/components/product/details/_product-summary.scss:L39-L41, core-libs/styles/scss/layout/page-templates/_product-detail.scss:L38-L40.

- La variante "additiva" sul placeholder funziona solo nella stessa compilazione Sass dove avviene l'`@extend` (il bundle della lib), non in `styles.scss`. DA VERIFICARE con compilazione: semantica di `@extend` per bundle, build non eseguibile (`projects/storefrontapp/project.json:L43-L47`, `:L85-L87`).
- Un componente custom con selector diverso non riceve né gli stili standard né lo skip (vedi Ricetta B2).
- La lib carrello ridefinisce il mixin `visible-focus` nel proprio bundle — `feature-libs/cart/_index.scss:L5-L11`: riscrivendo lo stile non va perso il focus visibile (vedi capitolo 38).

3.8 Ricetta D2 — Colori del tema: variabili Sass e custom property (SCSS)

Quando usarlo

Per applicare i colori del brand. Due livelli: **build-time** (variabili Sass `$primary` & co., una palette per tutta l'app) e **runtime** (classi tema che ridefiniscono `--cx-color-*`, selezionabili per sito o dall'utente).

Esempio

Esempio illustrativo, non presente nel repo (SCSS, con un complemento TypeScript per registrare il tema opzionale).

SCSS

```scss
// --- src/styles-config.scss
$useLatestStyles: true;

// build-time: solo variabili `!default` del tema santorini, assegnate PRIMA di scss/core
$primary: #0a6e4f;
$primary-accent: #07523b;
$text: #1b1b1b;

// --- src/brand-themes.scss  (importato in styles.scss DOPO '@spartacus/styles')
// runtime: una classe tema ridefinisce le custom property generate su :root
.brand-dark {
  --cx-color-background: #121212;
  --cx-color-text: #f2f2f2;
  --cx-color-inverse: #1e1e1e;
  --cx-color-primary: #4fd1a5;
  --cx-color-primary-accent: #2fae84;
  --cx-color-visual-focus: #ffd166;
}

// --- src/styles.scss (estratto, ordine come nell'app demo)
// @import 'styles-config';
// @import '@spartacus/styles/scss/core';
// ... bootstrap ...
// @import '@spartacus/styles';
// @import 'brand-themes';
```

TYPESCRIPT

```typescript
// --- complemento: rendere selezionabile il tema "brand-dark"
import { NgModule } from '@angular/core';
import { SiteThemeConfig, provideConfig } from '@spartacus/core';
```

```
@NgModule({
  providers: [
    provideConfig(<SiteThemeConfig>{
      siteTheme: {
        // array: sostituisce i default, quindi ripeto i due temi high-contrast
        optionalThemes: [
          { i18nNameKey: 'siteThemeSwitcher.themes.highContrastDark', className: 'cx-theme-high-
contrast-dark' },
          { i18nNameKey: 'siteThemeSwitcher.themes.highContrastLight', className: 'cx-theme-high-
contrast-light' },
          { i18nNameKey: 'brandTheme.dark', className: 'brand-dark' },
        ],
      },
    }),
  ],
})
export class BrandThemeModule {}
```

Catene di export: `SiteThemeConfig`: core-libs/core/public_api.ts:L32-L32 → core-libs/core/src/site-theme/index.ts:L7-L7 → core-libs/core/src/site-theme/config/site-theme-config.ts:L15-L15; `provideConfig`: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18. Il tipo degli elementi è `SiteTheme` (core-libs/core/public_api.ts:L26-L26 → core-libs/core/src/model/index.ts:L12-L12 → core-libs/core/src/model/misc.model.ts:L140-L143). Sass: `@spartacus/styles/scss/core` → core-libs/styles/scss/_core.scss:L5-L5 → core-libs/styles/scss/_theme.scss:L10-L10 → core-libs/styles/scss/theme/santorini/_variables.scss:L30-L50.

Perché funziona (file sorgente)

1. Le variabili colore del tema santorini sono `!default` — core-libs/styles/scss/theme/santorini/_variables.scss:L30-L50; il commento d'intestazione dice che si possono ridefinire valori ma non aggiungere variabili — core-libs/styles/scss/theme/santorini/_variables.scss:L1-L9.
2. Il tema attivo è importato da `_theme.scss` — core-libs/styles/scss/_theme.scss:L9-L12 — tramite `scss/core` — core-libs/styles/scss/_core.scss:L1-L5.
3. `$theme-colors` è costruita dalle variabili — core-libs/styles/scss/theme/santorini/_variables.scss:L60-L85 — e `root.scss` la traduce in `--cx-color-<nome>` su `:root` — core-libs/styles/scss/root.scss:L1-L6.
4. Gli stili dei componenti usano le custom property (es. `var(--cx-color-secondary)`) — feature-libs/cart/base/styles/components/_add-to-cart.scss:L1-L30.
5. L'app demo definisce classi tema che ridefiniscono `--cx-color-*` — projects/storefrontapp/src/cms-themes.scss:L1-L26 — importate per ultime perché devono sovrascrivere `:root` — projects/storefrontapp/src/styles.scss:L27-L29.
6. `ThemeService.setTheme` aggiunge/rimuove la classe sull'elemento del root component — core-libs/storefront/layout/theme/theme.service.ts:L52-L61; il tema di default viene da `context.theme` (o dal base site con il toggle `applyBaseSiteThemeFromCms`), gli opzionali da `siteTheme.optionalThemes` — core-libs/core/src/site-theme/facade/site-theme.service.ts:L51-L63, :L86-L89; i default sono due temi high-contrast — core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22.

Limiti e trappole

- `$theme-colors`, `$background-focus` e `$border-focus` **non** sono `!default`: vengono riassegnate a ogni import del tema, quindi non si ridefiniscono da `styles-config` — core-libs/styles/scss/theme/santorini/_variables.scss:L51-L52, :L60-L85. Per quei colori si usano le custom property. DA VERIFICARE con compilazione (build Sass non eseguibile).
- Le variabili Sass vanno in `styles-config.scss`, perché ogni bundle di lib re-importa il tema: assegnarle solo in `styles.scss` lascerebbe i bundle delle lib con i colori standard — projects/storefrontapp/src/styles/lib-cart.scss:L1-L2.
- La classe tema è applicata al root component, non a `cx-storefront` né a `html` — core-libs/storefront/layout/theme/theme.service.ts:L52-L61.
- `siteTheme.optionalThemes` è un array: va ripetuto per intero — core-libs/core/src/config/utils/deep-merge.ts:L43-L45. La chiave `i18nNameKey` del nuovo tema va tradotta (Ricetta E2).
- Il `theme` del base site da CMS è usato solo con il toggle `applyBaseSiteThemeFromCms` e se `context.theme` statico non è impostato — core-libs/core/src/site-theme/facade/site-theme.service.ts:L51-L63. Approfondimento: capitolo 38.

3.9 Ricetta E1 — Sovrascrivere chiavi di traduzione esistenti

Quando usarlo

Per cambiare il testo di etichette standard (qui: "Proceed to Checkout" del carrello) senza toccare i file della lib.

Esempio

Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// --- src/app/spartacus/translation-overrides.module.ts
// Da importare DOPO SpartacusFeaturesModule (es. accanto a SpartacusConfigurationModule).
import { NgModule } from '@angular/core';
import { I18nConfig, provideConfig } from '@spartacus/core';

@NgModule({
  providers: [
    provideConfig(<I18nConfig>{
      i18n: {
        resources: {
          // lingua -> chunk -> namespace -> chiave (stessa struttura del chunk originale)
          en: {
            cart: {
              cartDetails: {
                proceedToCheckout: 'Go to checkout',
              },
            },
          },
          de: {
            cart: {
              cartDetails: {
                proceedToCheckout: 'Zur Kasse',
```

```
              },
            },
          },
        },
      },
    }),
  ],
})
export class TranslationOverridesModule {}
```

Catene di export: `I18nConfig`: core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L7-L7 → core-libs/core/src/i18n/config/i18n-config.ts:L18-L18; `provideConfig`: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18.

Perché funziona (file sorgente)

1. `i18n.resources` è di tipo `TranslationResources` (lingua → chunk → oggetto chiavi) — core-libs/core/src/i18n/config/i18n-config.ts:L78-L78, core-libs/core/src/i18n/translation-resources.ts:L7-L18.
2. Il chunk `cart` contiene il namespace `cartDetails` — feature-libs/cart/base/assets/translations/translations.ts:L9-L19 — e il JSON ha i namespace al primo livello — feature-libs/cart/base/assets/translations/en/cart.json:L1-L5.
3. `RootConfig` è il `deepMerge` dei `ConfigChunk` nell'ordine del multi-provider — core-libs/core/src/config/config-tokens.ts:L37-L39; `resources` è un oggetto e viene fuso chiave per chiave — core-libs/core/src/config/utils/deep-merge.ts:L36-L45.
4. Nei multi-provider Angular vengono prima i provider dei moduli importati, poi quelli del modulo — angular:packages/core/src/di/provider_collection.ts:L293-L321, :L346-L353; `SpartacusModule` importa `SpartacusFeaturesModule` prima di `SpartacusConfigurationModule` — projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L20 — e le traduzioni della lib cart sono fornite nel feature module — projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L56-L67.
5. Le `resources` statiche sono aggiunte a i18next all'init con `addResourceBundle(lang, chunk, data, deep=true, overwrite=true)` — core-libs/core/src/i18n/i18next/i18next-initializer.ts:L36-L44, :L79-L92.
6. La chiave effettiva è `<chunk>:<chiave>` e il chunk di `cartDetails.proceedToCheckout` è ricavato dal primo segmento — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L121-L123, core-libs/core/src/i18n/translation-chunk.service.ts:L41-L49.

Limiti e trappole

- L'ordine conta: un override fornito *prima* del `provideConfig` della lib viene sovrascritto da quest'ultimo — core-libs/core/src/config/config-tokens.ts:L37-L39, angular:packages/core/src/di/provider_collection.ts:L293-L353.
- La struttura deve rispettare il chunk originale: mettere `cartDetails` sotto un chunk diverso crea una chiave i18next diversa e l'override non si vede — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L121-L123.
- Le risorse sono lette una sola volta, all'`APP_INITIALIZER` a config `i18n` stabile: fornirle in un chunk lazy non funziona. DA VERIFICARE a runtime (dedotto da core-libs/core/src/i18n/i18next/i18next-providers.ts:L13-L25 e core-libs/core/src/i18n/i18next/i18next-initializer.ts:L79-L92).

- Chiave mancante: in dev compare `[chiave]` con warning, in produzione uno spazio non separabile — `core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L100-L119`. Approfondimento: capitolo 37.

3.10 Ricetta E2 — Nuovo chunk di traduzioni per una feature custom

Quando usarlo

Quando una feature dell'app (Ricette A, B1, C1, D2) introduce testi propri. Si crea un chunk dedicato con i propri namespace, fornito eager insieme al root module della feature.

Esempio

Esempio illustrativo, non presente nel repo.

TYPESCRIPT

```typescript
// --- src/app/my-store/assets/translations.ts
import { TranslationChunksConfig, TranslationResources } from '@spartacus/core';

// chunk "myStore" -> namespace (primo segmento delle chiavi)
export const myStoreTranslationChunksConfig: TranslationChunksConfig = {
  myStore: ['loyalty', 'giftWrap', 'loyaltyCheckout', 'brandTheme'],
};

export const myStoreTranslations: TranslationResources = {
  en: {
    myStore: {
      loyalty: { balance: 'You have {{points}} points' },
      giftWrap: { label: 'Gift wrapped' },
      loyaltyCheckout: { stepName: 'Loyalty points', skip: 'Do not use points' },
      brandTheme: { dark: 'Brand dark' },
    },
  },
  de: {
    myStore: {
      loyalty: { balance: 'Sie haben {{points}} Punkte' },
      giftWrap: { label: 'Als Geschenk verpackt' },
      loyaltyCheckout: { stepName: 'Treuepunkte', skip: 'Keine Punkte verwenden' },
      brandTheme: { dark: 'Marke dunkel' },
    },
  },
};

// --- src/app/spartacus/features/my-store-i18n.module.ts  (eager)
import { NgModule } from '@angular/core';
import { I18nConfig, provideConfig } from '@spartacus/core';
import {
  myStoreTranslationChunksConfig,
  myStoreTranslations,
} from '../../my-store/assets/translations';
```

```
@NgModule({
  providers: [
    provideConfig(<I18nConfig>{
      i18n: {
        resources: myStoreTranslations,
        chunks: myStoreTranslationChunksConfig,
        fallbackLang: 'en',
      },
    }),
  ],
})
export class MyStoreI18nModule {}
```

Catene di export: `TranslationChunksConfig`, `TranslationResources`: core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L17-L17 → core-libs/core/src/i18n/translation-resources.ts:L16-L16 / :L11-L11; `I18nConfig`: core-libs/core/public_api.ts:L23-L23 → core-libs/core/src/i18n/index.ts:L7-L7 → core-libs/core/src/i18n/config/i18n-config.ts:L18-L18; `provideConfig`: core-libs/core/public_api.ts:L15-L15 → core-libs/core/src/config/index.ts:L9-L9 → core-libs/core/src/config/config-providers.ts:L18-L18.

Perché funziona (file sorgente)

1. `i18n.chunks` mappa chunk → namespace; lo scopo dei chunk è il lazy loading — core-libs/core/src/i18n/config/i18n-config.ts:L84-L90; `TranslationChunksConfig` è chunk → `string[]` — core-libs/core/src/i18n/translation-resources.ts:L16-L18.
2. Modello: EDD usa un chunk con un solo namespace omonimo — feature-libs/estimated-delivery-date/assets/translations/translations.ts:L9-L12 — ed è fornito eager nel feature module, insieme al root module — projects/storefrontapp/src/app/spartacus/features/estimated-delivery-date/estimated-delivery-date-feature.module.ts:L18-L35.
3. `TranslationChunkService` costruisce la mappa namespace → chunk nel costruttore, con warning in dev per i duplicati — core-libs/core/src/i18n/translation-chunk.service.ts:L20-L37.
4. L'interpolazione `{{points}}` viene dai parametri passati alla pipe, come nel modello EDD — feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.html:L10-L17; la pipe `cxTranslate` è `TranslatePipe` — core-libs/core/src/i18n/translate.pipe.ts:L25-L29.
5. La chiave dello step (`loyaltyCheckout.stepName`) è tradotta dalla progress bar con `cxTranslate` — feature-libs/checkout/base/components/checkout-progress/checkout-progress.component.html:L27-L29; la chiave del tema (`brandTheme.dark`, campo `i18nNameKey`) è tradotta dal site theme switcher — core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.component.html:L15-L16.

Limiti e trappole

- Un namespace non elencato in `chunks` è trattato come chunk omonimo (si tenta di caricare un chunk con quel nome) — core-libs/core/src/i18n/translation-chunk.service.ts:L41-L49.
- Namespace duplicati tra chunk: vince il primo, warning solo in dev — core-libs/core/src/i18n/translation-chunk.service.ts:L22-L36. Il chunk core `common` contiene molti namespace (`common`, `miniCart`, `formErrors`, …): scegliere nomi che non collidano — core-libs/assets/src/translations/translations.ts:L7-L27.

- Due provider con lo stesso nome di chunk in `i18n.chunks` si sostituiscono (array) — core-libs/core/src/config/utils/deep-merge.ts:L43-L45.
- Le traduzioni vanno fornite eager: fornite nel chunk lazy (es. dentro *LoyaltyModule*) arriverebbero dopo l'init di i18next e dopo la costruzione di `TranslationChunkService`. DA VERIFICARE a runtime (core-libs/core/src/i18n/i18next/i18next-providers.ts:L13-L25, core-libs/core/src/i18n/translation-chunk.service.ts:L20-L37).
- Per caricare i chunk in lazy da asset locali la via raccomandata è `i18n.backend.loader` (dynamic import), non `loadPath` — core-libs/core/src/i18n/config/i18n-config.ts:L42-L71; con `resources` statiche tutte le lingue finiscono nel bundle iniziale.
- `CmsComponentMapping.i18nKeys` permette di dichiarare le chiavi di un componente CMS — core-libs/core/src/cms/config/cms-config.ts:L76-L76.

---

4. Diagrammi

4.1 Feature lib custom: dal bootstrap al componente nel chunk (Ricetta A)

■ TAVOLA 53.1 diagramma a tutta pagina, in fondo al capitolo

Ogni passaggio corrisponde a §3.1, "Perché funziona" punti 2-8 (core-libs/storefront/cms-structure/services/cms-features.service.ts:L56-L75, core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73, core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157, core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L38-L57).

4.2 Riga carrello estesa e add-to-cart (Ricette B1, B2, B3)

■ TAVOLA 53.2 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L278-L302, core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88, feature-libs/cart/base/core/event/cart-event.builder.ts:L47-L87, feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.html:L37-L37. Il passo "reload cart con fields" è una semplificazione: la catena effects → adapter non è riletta qui (vedi capitolo 25).

4.3 Nuovo step di checkout (Ricetta C1)

■ TAVOLA 53.3 diagramma a tutta pagina, in fondo al capitolo

Riferimenti: feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23, feature-libs/checkout/base/components/services/checkout-step.service.ts:L88-L100, feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L73-L132, feature-libs/checkout/base/components/checkout-progress/checkout-progress.module.ts:L37-L37, core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25.

4.4 Stili e traduzioni: dove si decide il risultato (Ricette D1, D2, E1, E2)

■ **TAVOLA 53.4** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: projects/storefrontapp/src/styles.scss:L1-L29, projects/storefrontapp/src/styles/lib-cart.scss:L1-L2, core-libs/styles/scss/_components.scss:L37-L53, core-libs/styles/scss/root.scss:L1-L6, core-libs/storefront/layout/theme/theme.service.ts:L52-L61, core-libs/core/src/config/config-tokens.ts:L37-L39, core-libs/core/src/i18n/i18next/i18next-providers.ts:L13-L25, core-libs/core/src/i18n/translation-chunk.service.ts:L20-L37.

---

5. Estratti di codice reale

Gli esempi della sezione 3 sono modellati sui seguenti estratti, copiati letteralmente.

Root module con featureModules senza `module` **e alias core** — feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L31-L50

TYPESCRIPT

```typescript
export function defaultCartSavedCartComponentsConfig(): CmsConfig {
  const config: CmsConfig = {
    featureModules: {
      [CART_SAVED_CART_FEATURE]: {
        cmsComponents: [
          'AddToSavedCartsComponent',
          'AccountSavedCartHistoryComponent',
          'SavedCartDetailsOverviewComponent',
          'SavedCartDetailsItemsComponent',
          'SavedCartDetailsActionComponent',
        ],
        dependencies: [CART_BASE_FEATURE],
      },

      // by default core is bundled together with components
      [CART_SAVED_CART_CORE_FEATURE]: CART_SAVED_CART_FEATURE,
    },
  };
  return config;
}
```

L'app fornisce `module` **e importa il root module** — projects/storefrontapp/src/app/spartacus/features/cart/cart-saved-cart-feature.module.ts:L21-L31

TYPESCRIPT

```typescript
@NgModule({
  imports: [SavedCartRootModule],
  providers: [
    provideConfig(<CmsConfig>{
      featureModules: {
        [CART_SAVED_CART_FEATURE]: {
```

```
          module: () =>
            import('@spartacus/cart/saved-cart').then((m) => m.SavedCartModule),
        },
      },
    }),
```

Facade proxy — feature-libs/cart/saved-cart/root/facade/saved-cart.facade.ts:L13-L22

```typescript
@Injectable({
  providedIn: 'root',
  useFactory: () =>
    facadeFactory({
      facade: SavedCartFacade,
      feature: CART_SAVED_CART_CORE_FEATURE,
      methods: [
        'editSavedCart',
        'deleteSavedCart',
        'getSavedCart',
// ...
```

Ri-fornitura della facade nel chunk — feature-libs/cart/saved-cart/core/facade/facade-providers.ts:L11-L17

```typescript
export const facadeProviders: Provider[] = [
  SavedCartService,
  {
    provide: SavedCartFacade,
    useExisting: SavedCartService,
  },
];
```

Proxy: resolver e sottoclasse anonima — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L52 e :L117-L126

```typescript
    if (!this.featureModules.isConfigured(feature)) {
      return throwError(
        () => new Error(`Feature ${feature} is not configured properly`)
      );
    }

    let facadeService$ = this.featureModules.resolveFeature(feature).pipe(
      map((moduleRef) => moduleRef.injector),
      map((injector) => injector.get(facadeClass))
    );
// ...
    const resolver$ = this.getResolver(feature, facade, async);
```

```
    const result: any = new (class extends (facade as any) {})();
    (methods ?? []).forEach((method) => {
      result[method] = (...args: any[]) =>
        this.call(resolver$, method as string, args);
    });
    (properties ?? []).forEach((property) => {
      result[property] = this.get(resolver$, property as string);
    });
```

Mapping della feature come default sotto la config statica — core-libs/storefront/cms-structure/services/cms-components.service.ts:L126-L133

TYPESCRIPT

```
          tap((featureComponentMapping) => {
            // We treat cms mapping configuration from a feature as a default,
            // that can be overridden by app/static configuration
            this.mappings[componentType] = deepMerge(
              {},
              featureComponentMapping,
              staticConfig
            );
```

Estensione della riga carrello (EDD) — feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L28-L37

TYPESCRIPT

```
  providers: [
    provideDefaultConfig(defaultOccCartWithEddConfig),
    provideOutlet({
      id: CartOutlets.ITEM_DETAILS,
      position: OutletPosition.AFTER,
      component: EstimatedDeliveryDateComponent,
    }),
  ],
})
export class ShowEstimatedDeliveryDateModule {}
```

Augmentation di `OrderEntry` — feature-libs/estimated-delivery-date/root/model/augmented-core.model.ts:L7-L15

TYPESCRIPT

```
import '@spartacus/cart/base/root';
import '@spartacus/order/root';
import { ArrivalSlot, ArrivalSlots } from './estimated-delivery-date.model';

declare module '@spartacus/cart/base/root' {
  interface OrderEntry {
    arrivalSlots?: ArrivalSlots[];
  }
}
```

Wrapper lazy del carrello — projects/storefrontapp/src/app/spartacus/features/cart/cart-base-wrapper.module.ts:L14-L25

TYPESCRIPT

```typescript
const extensions: Type<any>[] = [];

if (environment.estimatedDeliveryDate) {
  extensions.push(EstimatedDeliveryDateModule);
}
if (environment.cpq) {
  extensions.push(CpqQuoteModule);
}
@NgModule({
  imports: [CartBaseModule, ...extensions],
})
export class CartBaseWrapperModule {}
```

`addToCart()`**: entries,** `addEntry`**, evento UI** — feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L278-L302

TYPESCRIPT

```typescript
    this.activeCartService
      .getEntries()
      .pipe(take(1))
      .subscribe((cartEntries) => {
        this.activeCartService.addEntry(
          this.productCode,
          quantity,
          this.pickupStore
        );

        // A CartUiEventAddToCart is dispatched.  This event is intended for the UI
        // responsible to provide feedback about what was added to the cart, like
        // the added to cart dialog.
        //
        // Because we call activeCartService.getEntries() before, we can be sure the
        // cart library is loaded already and that the event listener exists.
        this.eventService.dispatch(
          this.createCartUiEventAddToCart(
            this.productCode,
            quantity,
            cartEntries.length,
            this.pickupStore
          )
        );
      });
```

Listener di eventi carrello — feature-libs/quote/core/event/quote-cart-event.listener.ts:L29-L34

```typescript
  protected listenToCartEvents(): void {
    this.subscription.add(
      this.eventService.get(CartAddEntrySuccessEvent).subscribe(() => {
        this.triggerQuoteReload();
      })
    );
```

Nuovo tipo di step (B2B) — feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23

```typescript
declare module '@spartacus/checkout/base/root' {
  enum CheckoutStepType {
    PAYMENT_TYPE = 'paymentType',
  }

  interface CheckoutState {
    costCenter?: CostCenter;
    purchaseOrderNumber?: string;
    paymentType?: PaymentType;
  }
}

(CheckoutStepType as any)['PAYMENT_TYPE'] = 'paymentType';
```

Lista step ridefinita (B2B) — feature-libs/checkout/b2b/root/config/default-b2b-checkout-config.ts:L13-L27

```typescript
export const defaultB2BCheckoutConfig: CheckoutConfig = {
  checkout: {
    steps: [
      {
        id: 'paymentType',
        name: 'checkoutB2B.progress.methodOfPayment',
        routeName: 'checkoutPaymentType',
        type: [CheckoutStepType.PAYMENT_TYPE],
      },
      {
        id: 'deliveryAddress',
        name: 'checkoutProgress.deliveryAddress',
        routeName: 'checkoutDeliveryAddress',
        type: [CheckoutStepType.DELIVERY_ADDRESS],
      },
// ...
```

`isStepSet` **del guard base** — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132

```typescript
protected isStepSet(step: CheckoutStep): Observable<GuardResult> {
  if (step && !step.disabled) {
    switch (step.type[0]) {
      case CheckoutStepType.DELIVERY_ADDRESS: {
        return this.isDeliveryAddress(step);
      }
      case CheckoutStepType.DELIVERY_MODE: {
        return this.isDeliveryModeSet(step);
      }
      // ...
      case CheckoutStepType.REVIEW_ORDER: {
        break;
      }
    }
  }
  return of(true);
}
```

Override del guard nel chunk (B2B) — feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37

```typescript
providers: [
  {
    provide: CheckoutAuthGuard,
    useExisting: CheckoutB2BAuthGuard,
  },
  {
    provide: CheckoutStepsSetGuard,
    useExisting: CheckoutB2BStepsSetGuard,
  },
],
```

Sostituzione di un componente di step (digital payments) — integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L32-L38

```typescript
  ConfigModule.withConfig({
    cmsComponents: {
      CheckoutPaymentDetails: {
        component: DpPaymentMethodComponent,
      },
    },
  }),
```

Skip degli stili per selettore — core-libs/styles/scss/_components.scss:L37-L44

```scss
@each $selector in $selectors {
  #{$selector} {
    // skip selectors if they're added to the $skipComponentStyles list
    @if (index($skipComponentStyles, $selector) ==null) {
      @extend %#{$selector} !optional;
    }
  }
}
```

Bundle stile di una lib nell'app demo — projects/storefrontapp/src/styles/lib-cart.scss:L1-L2

```scss
@import '../styles-config';
@import '@spartacus/cart';
```

Variabili tema `!default` **e no** — core-libs/styles/scss/theme/santorini/_variables.scss:L30-L32 e :L50-L52

```scss
// change theme-colors here
$primary: #1f7bc0 !default;
$primary-accent: #055f9f !default;
// ...
$visual-focus: #6d9df7 !default;
$background-focus: rgba(80, 176, 244, 0.1);
$border-focus: rgba(166, 204, 232, 0.6);
```

Custom property da `$theme-colors` — core-libs/styles/scss/root.scss:L1-L6

```scss
@import 'theme';

:root {
  @each $color, $value in $theme-colors {
    --cx-color-#{$color}: #{$value};
  }
```

Traduzioni di una lib fornite eager dall'app — projects/storefrontapp/src/app/spartacus/features/estimated-delivery-date/estimated-delivery-date-feature.module.ts:L18-L35

```typescript
@NgModule({
  imports: [EstimatedDeliveryDateRootModule],
  providers: [
    provideConfig(<I18nConfig>{
      i18n: {
        resources: {
          en: estimatedDeliveryDateTranslationsEn,
          ja: estimatedDeliveryDateTranslationsJa,
          de: estimatedDeliveryDateTranslationsDe,
          zh: estimatedDeliveryDateTranslationsZh,
```

```
        },
        chunks: estimatedDeliveryDateTranslationChunksConfig,
        fallbackLang: 'en',
      },
    }),
  ],
})
export class EstimatedDeliveryDateFeatureModule {}
```

Risorse statiche aggiunte all'init — core-libs/core/src/i18n/i18next/i18next-initializer.ts:L79-L92

TYPESCRIPT

```
  protected addTranslationResources(): void {
    const resources: TranslationResources = this.config.i18n?.resources ?? {};
    Object.keys(resources).forEach((lang) => {
      Object.keys(resources[lang]).forEach((chunkName) => {
        this.i18next.addResourceBundle(
          lang,
          chunkName,
          resources[lang][chunkName],
          true,
          true
        );
      });
    });
  }
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
featureModules[name].module	() => Promise<any> (NgModule)	— (la fornisce l'app)	core-libs/core/src/cms/config/cms-config.ts:L113-L113	core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35, :L42-L73
featureModules[name].cmsComponents	string[]	per lib (es. feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L35-L41)	core-libs/core/src/cms/config/cms-config.ts:L121-L121	core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L72
featureModules[name].dependencies	(fn | string)[]	es. [CART_BASE_FEATURE] (feature-libs/checkout/base/root/checkout-root.module.ts:L42-L42)	core-libs/core/src/cms/config/cms-config.ts:L117-L117	core-libs/core/src/lazy-loading/feature-modules.service.ts:L108-L123
featureModules[alias]	string	es. cartBaseCore → cartBase	core-libs/core/src/cms/config/cms-config.ts:L129-L129	core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73
cmsComponents[type].component/guards/data/i18nKeys	CmsComponentMapping	per modulo	core-libs/core/src/cms/config/cms-config.ts:L60-L93	core-libs/storefront/cms-structure/services/cms-components.service.ts:L117-L141, :L253-L261

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
checkout.steps	CheckoutStep[]	4 step	feature-libs/checkout/base/root/config/default-checkout-config.ts:L10-L42	feature-libs/checkout/base/components/services/checkout-step.service.ts:L88-L100
checkout.flows[paymentProvider]	CheckoutFlow	—	feature-libs/checkout/base/root/config/checkout-config.ts:L18-L55	feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L35-L44
routing.routes.<routeName>.paths	string[]	es. checkoutDeliveryAddress	feature-libs/checkout/base/root/config/default-checkout-routing-config.ts:L9-L20	feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L81-L84
backend.occ.endpoints.cart / carts / createCart	string (con fields)	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L15-L19	feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L9-L9	adapter OCC del carrello (DA VERIFICARE: righe non rilette per questo capitolo)
i18n.resources	TranslationResources	—	core-libs/core/src/i18n/config/i18n-config.ts:L78-L78	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L79-L92
i18n.chunks	{ [chunk]: string[] }	—	core-libs/core/src/i18n/config/i18n-config.ts:L88-L90	core-libs/core/src/i18n/translation-chunk.service.ts:L20-L37
i18n.fallbackLang	string | false	—	core-libs/core/src/i18n/config/i18n-config.ts:L23-L23	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L52-L52
i18n.backend.loader	(lng, chunk) => Promise	—	core-libs/core/src/i18n/config/i18n-config.ts:L68-L71	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L63-L68
context.theme	string[]	—	core-libs/core/src/site-context/config/site-context-config.ts:L15-L17	core-libs/core/src/site-theme/facade/site-theme.service.ts:L51-L58
siteTheme.optionalThemes	SiteTheme[]	2 temi high-contrast	core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22	core-libs/core/src/site-theme/facade/site-theme.service.ts:L86-L89
Sass $skipComponentStyles	lista	()	core-libs/styles/scss/_components.scss:L7-L7; feature-libs/cart/base/styles/components/_index.scss:L26-L26	core-libs/styles/scss/_components.scss:L40-L40, :L49-L49
Sass $cart-components-allowlist (e analoghe per area)	lista	per lib	feature-libs/cart/base/styles/components/_index.scss:L19-L24; feature-libs/checkout/base/styles/components/steps/_index.scss:L7-L35	loop @each negli stessi file

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
Sass `$page-template-allowlist` / `$page-template-blocklist`	lista	14 template / `()`	`core-libs/styles/scss/_page-template.scss:L9-L19`	`core-libs/styles/scss/_page-template.scss:L21-L27`
Sass `$primary` … `$visual-focus`, `$disabled*`	colore `!default`	santorini	`core-libs/styles/scss/theme/santorini/_variables.scss:L30-L58`	`$theme-colors` `core-libs/styles/scss/theme/santorini/_variables.scss:L60-L85` → `core-libs/styles/scss/root.scss:L3-L6`

7. Commenti e note notevoli dal sorgente

- `// by default core is bundled together with components` — sopra gli alias `*_CORE_FEATURE` dei root module: è il motivo per cui le facade puntano alla feature "core" e questa è un alias — `feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L45-L46`; per il cart il commento aggiunge "The cart lib should keep using this default" — `feature-libs/cart/base/root/cart-base-root.module.ts:L45-L53`.
- `// TODO: Inline this factory when we start releasing Ivy compiled libraries` — sul `provideDefaultConfigFactory` del root module saved-cart — `feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L30-L30`. In un'app (non libreria) la factory non serve: nell'esempio A uso `provideDefaultConfig`.
- `* Denotes that feature should have to be initialized with an async delay. Required to make lazy NgRx store feature ready.` — `core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L32-L36`.
- `// TODO:#deprecation-checkout - remove once we remove ngrx` sull'`async: true` delle facade checkout — `feature-libs/checkout/base/root/facade/checkout-delivery-modes.facade.ts:L26-L26`.
- `* Because we call activeCartService.getEntries() before, we can be sure the cart library is loaded already and that the event listener exists.` — `feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L292-L293`: chi sovrascrive `addToCart()` senza `super` deve preservare questo ordine.
- `/** @private We plan to drive the outlets by standard configuration */` — su `PROVIDE_OUTLET_OPTIONS` — `core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L10-L12`.
- `TODO:#9574 - we should be able to remove the spread of CHECKOUT_BASE_CMS_COMPONENTS` — `feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L23-L31`: lo spread nella Ricetta C1 ha la stessa origine.
- `// Import augmented types first` — `feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L8-L9`.
- `// we want to grab cms configuration available at config initialization phase / as lazy-loaded modules can affect global configuration resulting in / non-deterministic state` — `core-libs/storefront/cms-structure/services/cms-components.service.ts:L64-L66`.
- `// Don't use i18next's 'resources' config key for adding static translations, because it will disable loading chunks from backend.` — `core-libs/core/src/i18n/i18next/i18next-initializer.ts:L39-L40`.
- `* This option is NOT recommended for loading translations from local assets.` (su `backend.loadPath`) — `core-libs/core/src/i18n/config/i18n-config.ts:L42-L45`.
- `// ORDER IMPORTANT: Spartacus core first` / `// ORDER IMPORTANT: Spartacus styles last` — `projects/storefrontapp/src/styles.scss:L3-L4`, `:L24-L25`.

- `// the $skipComponentStyles is used as a list of component styles that can be ignore during the sass compilation.` — core-libs/styles/scss/_components.scss:L4-L6.
- `// You cannot: 1) Add new variables` — intestazione del tema santorini — core-libs/styles/scss/theme/santorini/_variables.scss:L8-L9.
- `* General rule is, that all config provided in libraries should be provided as default config.` — core-libs/core/src/config/config-tokens.ts:L58-L58: nelle ricette uso `provideDefaultConfig` per ciò che si comporta "da libreria" (root module, mapping nel chunk) e `provideConfig` per le decisioni dell'app.

8. Extension point

EXTENSION POINT	RICETTA	MECCANISMO	CAPITOLO DI RIFERIMENTO
`featureModules` (`module`, `cmsComponents`, `dependencies`, alias)	§3.1	core-libs/core/src/cms/config/cms-config.ts:L109-L134	12
`facadeFactory` + `FacadeDescriptor`	§3.1	core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24	20
`OrderEntry` augmentation + endpoint `cart` con `fields`	§3.2	feature-libs/cart/base/root/public_api.ts:L18-L20; feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L9-L19	25, ricette OCC
`CartOutlets` + `provideOutlet` + `CartItemContext`	§3.2	feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L31; core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L47	ricette CMS/outlet
Rimappatura `ProductAddToCartComponent` / sottoclasse `AddToCartComponent`	§3.3	feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts:L34-L44	25, ricette CMS
Eventi `CartAddEntrySuccessEvent`, `CartUiEventAddToCart`	§3.4	feature-libs/cart/base/root/events/cart.events.ts:L66-L136	20, ricette eventi
`checkout.steps`, `CheckoutStepType` augmentation, rotta, `CheckoutStepsSetGuard`	§3.5	feature-libs/checkout/base/root/config/checkout-config.ts:L18-L55; feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132	26
Rimappatura componenti di step (`CheckoutDeliveryMode`, `CheckoutPaymentDetails`, …)	§3.6	feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.module.ts:L36-L50	26
`$skipComponentStyles`, allowlist, placeholder `%cx-*`	§3.7	core-libs/styles/scss/_components.scss:L37-L53	38
Variabili tema `!default`, `--cx-color-*`, `siteTheme.optionalThemes`	§3.8	core-libs/styles/scss/theme/santorini/_variables.scss:L30-L50; core-libs/styles/scss/root.scss:L1-L6	38
`i18n.resources` (override)	§3.9	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L79-L92	37
`i18n.chunks` + `resources` (nuovo chunk), `backend.loader`	§3.10	core-libs/core/src/i18n/config/i18n-config.ts:L42-L92	37

Ricette correlate della Parte III: configurazione, DI, interceptor, eventi · componenti CMS, outlet, route e layout · OCC, adapter, converter, endpoint · API pubblica vs interna e upgrade.

9. Trappole, casi limite, comportamenti controintuitivi

1. **Root module senza `module` fornito dall'app**: il tipo CMS non viene mappato (warning) e la facade lancia "Feature X is not configured properly" — `core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L68`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47`.
2. **La config root dell'app vince su quella dei chunk lazy**: un `provideConfig` dentro un chunk non sovrascrive la config root fornita con `provideConfig` (RootConfig), ma vince su una `provideDefaultConfig` eager — `core-libs/core/src/config/services/configuration.service.ts:L92-L99`.
3. **Array sostituiti**: `checkout.steps`, `featureModules[x].cmsComponents`, `cmsComponents[x].guards`, `i18n.chunks[x]`, `siteTheme.optionalThemes` vanno ridefiniti per intero — `core-libs/core/src/config/utils/deep-merge.ts:L7-L9`, `:L43-L45`.
4. **Override statico di un componente di feature**: il chunk della feature viene comunque caricato e i campi non ridefiniti (`guards`, `data`) restano — `core-libs/storefront/cms-structure/services/cms-components.service.ts:L92-L99`, `:L126-L133`.
5. **Provider nel posto sbagliato**: un override di un servizio del carrello (adapter, guard) va nel chunk del carrello/checkout (wrapper), dopo il modulo base; un provider root è oscurato da quello del chunk. `DA VERIFICARE`: risoluzione DI dedotta, non eseguita — `feature-libs/cart/base/occ/cart-base-occ.module.ts:L50-L53`, `core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L97`.
6. **Outlet registrati da un chunk**: servono `OutletModule.forChild()` nello stesso injector e il caricamento del chunk — `core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88`.
7. **Endpoint con `fields`**: stringhe intere che due estensioni si contendono — `feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L10-L16`. Anche le lib OPF ridefiniscono endpoint cart (`integration-libs/opf/global-functions/root/config/default-occ-opf-cart-config.ts`, `integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts`).
8. **Nuovo step senza guard esteso**: `isStepSet` ritorna `true` per tipi sconosciuti — `feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132`; con il B2B attivo va esteso `CheckoutB2BStepsSetGuard` — `feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L57-L86`.
9. **Augmentation di enum senza valore runtime**: `CheckoutStepType.X` è `undefined` — `feature-libs/checkout/b2b/root/model/augmented-types.ts:L23-L23`.
10. **Guard funzionali scartati**: `isCanActivate` richiede un oggetto con `canActivate` — `core-libs/storefront/cms-structure/services/utils.ts:L9-L11`.
11. **Stili legati al selector**: un componente con selector diverso perde gli stili globali — `feature-libs/cart/base/styles/components/_index.scss:L28-L35`.
12. `$skipComponentStyles` **e allowlist assegnati dopo l'import**: nessun effetto (`!default`) — `core-libs/styles/scss/_components.scss:L1-L19`.
13. `$theme-colors` **non** `!default`: non ridefinibile da `styles-config` — `core-libs/styles/scss/theme/santorini/_variables.scss:L60-L85`.
14. **Traduzioni nel chunk lazy**: lette troppo tardi. `DA VERIFICARE` a runtime — `core-libs/core/src/i18n/i18next/i18next-providers.ts:L13-L25`, `core-libs/core/src/i18n/translation-chunk.service.ts:L20-L37`.
15. **Override i18n prima della lib**: perde — `core-libs/core/src/config/config-tokens.ts:L37-L39`.
16. `CartUiEventAddToCart` **non è un** `CartEvent` — `feature-libs/cart/base/root/events/cart.events.ts:L122-L122`.

17. **Chiamata di facade = download del chunk** anche senza subscribe — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L92.

---

10. Come fare debug

- **Componente CMS custom non renderizzato** → in dev cercare in console "No component implementation found for the CMS component type" — core-libs/storefront/cms-structure/services/cms-components.service.ts:L170-L176. Se compare: la feature non ha `module` o il tipo non è in `cmsComponents` (core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L72), oppure il mapping non è fornito nel chunk (core-libs/storefront/cms-structure/services/cms-features.service.ts:L188-L205).
- **Errore "Feature ... is not configured properly"** → la feature puntata dalla facade (dopo alias) non ha `module` — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L43-L47, core-libs/core/src/lazy-loading/feature-modules.service.ts:L33-L35.
- **Chunk caricato?** → nel tab Network del browser compare il file JS del chunk quando la pagina contiene il tipo CMS o alla prima chiamata della facade; `resolveFeature` è in cache, quindi una sola volta — core-libs/core/src/lazy-loading/feature-modules.service.ts:L42-L73. Dopo la creazione del modulo viene dispatchato `ModuleInitializedEvent` (ascoltabile con `EventService.get`) — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L78-L108.
- **Config effettiva** → `ConfigurationService.unifiedConfig$` emette la config unificata dopo ogni chunk lazy con config — core-libs/core/src/config/services/configuration.service.ts:L30-L30, :L92-L100: sottoscriverla in dev mostra se il proprio override c'è e chi vince.
- **Step di checkout che rimanda indietro** → in dev il guard logga "Missing step with route '...' in checkout configuration or this step is disabled." — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L93-L101. Confrontare la URL della pagina con `routing.routes.<routeName>.paths[0]`.
- **Guard sostituito attivo?** → mettere un breakpoint in `isStepSet` della sottoclasse; se non viene mai chiamato con il B2B attivo, è il guard B2B a essere in uso (feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L57-L86).
- **Traduzione mancante** → in dev il testo appare come `[chiave]` e la console riporta "Translation keys missing [...]. Attempted to load 'chiave' from chunk 'chunk'." — core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L100-L119: il chunk indicato rivela se la mappa namespace → chunk è quella attesa (core-libs/core/src/i18n/translation-chunk.service.ts:L41-L49). Namespace duplicati producono un warning dedicato in dev — core-libs/core/src/i18n/translation-chunk.service.ts:L34-L36.
- **Stile non applicato / non escluso** → ispezionare il CSS compilato del bundle della lib (es. bundle `cart` in projects/storefrontapp/project.json:L85-L87): con lo skip attivo non deve contenere regole generate per `cx-add-to-cart`; controllare che `styles-config` sia importato in testa al bundle (projects/storefrontapp/src/styles/lib-cart.scss:L1-L1).
- **Colori runtime** → negli strumenti del browser verificare quale classe tema è sul root component (core-libs/storefront/layout/theme/theme.service.ts:L52-L61) e da dove arriva il valore di `--cx-color-primary` (`:root` da core-libs/styles/scss/root.scss:L3-L6 o la classe tema).

11. Domande di autoverifica

1. **Perché il root module di una feature non dichiara** `module`**?** Perché è l'app a decidere da dove caricare il chunk (o se usare un wrapper con estensioni); il root module dichiara solo i tipi CMS coperti e gli alias. Senza `module` la feature non entra nella mappa dei componenti. — feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L31-L50; projects/storefrontapp/src/app/spartacus/features/cart/cart-saved-cart-feature.module.ts:L21-L31; core-libs/storefront/cms-structure/services/cms-features.service.ts:L65-L72.
2. **Cosa succede se chiamo sul proxy della facade un metodo non elencato in** `methods`**?** Il metodo non esiste sulla sottoclasse anonima (la classe è astratta): errore a runtime. — core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L119-L126.
3. **Perché nella Ricetta B1 il modulo con** `provideOutlet` **va importato nel wrapper del carrello e non in una feature lazy propria?** Perché gli outlet sono registrati dal `MODULE_INITIALIZER` di `OutletModule.forChild()`, letto con `self: true` nell'injector del modulo lazy; nel wrapper il modulo condivide l'injector di `CartBaseComponentsModule` che importa `forChild()`. — core-libs/core/src/lazy-loading/lazy-modules.service.ts:L150-L157; core-libs/storefront/cms-structure/outlet/outlet.module.ts:L72-L88; feature-libs/cart/base/components/cart-base-components.module.ts:L30-L30.
4. **Rimappando** `ProductAddToCartComponent` **nella config dell'app, il chunk** `addToCart` **viene ancora caricato?** Sì: per un tipo gestito da feature si passa sempre dal resolver della feature, e il mapping finale è `deepMerge(feature, statico)`. — core-libs/storefront/cms-structure/services/cms-components.service.ts:L92-L99, :L126-L133.
5. **Perché nell'augmentation di** `CheckoutStepType` **serve anche** `(CheckoutStepType as any)['X'] = 'x'`**?** Perché `declare module` aggiunge il membro solo al tipo; l'oggetto enum a runtime resta quello della libreria. — feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23.
6. **Aggiungo un nuovo step senza estendere** `CheckoutStepsSetGuard`**: cosa succede allo step successivo?** È accessibile anche se il nuovo step non è completato, perché `isStepSet` ritorna `of(true)` per i tipi che non conosce. — feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L106-L132.
7. **Ho esteso** `isStepSet` **ma con il B2B attivo non viene chiamato. Perché?** Il B2B registra `CheckoutB2BStepsSetGuard`, che ridefinisce `canActivate` e chiama `isB2BStepSet`. — feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L57-L86; feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L28-L37.
8. **Perché** `$skipComponentStyles` **va in** `styles-config.scss` **e non in** `styles.scss`**?** Perché è `!default` in ogni lib e ogni lib è un bundle Sass separato che importa `styles-config` in testa. — feature-libs/cart/base/styles/components/_index.scss:L26-L26; projects/storefrontapp/src/styles/lib-cart.scss:L1-L2; projects/storefrontapp/project.json:L85-L87.
9. **Posso cambiare** `$theme-colors` **da** `styles-config.scss`**?** No: non è `!default` e viene riassegnata dall'import del tema; si cambiano le singole variabili `!default` (`$primary`, …) oppure le custom property `--cx-color-*`. — core-libs/styles/scss/theme/santorini/_variables.scss:L30-L50, :L60-L85; core-libs/styles/scss/root.scss:L1-L6.
10. **Il mio override di** `cartDetails.proceedToCheckout` **non ha effetto. Cosa controllo?** Che sia sotto il chunk giusto (`en.cart.cartDetails…`, la chiave i18next è `cart:cartDetails.proceedToCheckout`) e che il suo `provideConfig` venga dopo quello della lib nell'ordine dei provider (es. in un modulo importato dopo `SpartacusFeaturesModule`). — feature-libs/cart/base/assets/translations/translations.ts:L9-L19; core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L121-L123; projects/storefrontapp/src/app/spartacus/spartacus.module.ts:L12-L20; core-libs/core/src/config/config-tokens.ts:L37-L39.

TAVOLA 53.1 **4.1 Feature lib custom: dal bootstrap al componente nel chunk (Ricetta A)**

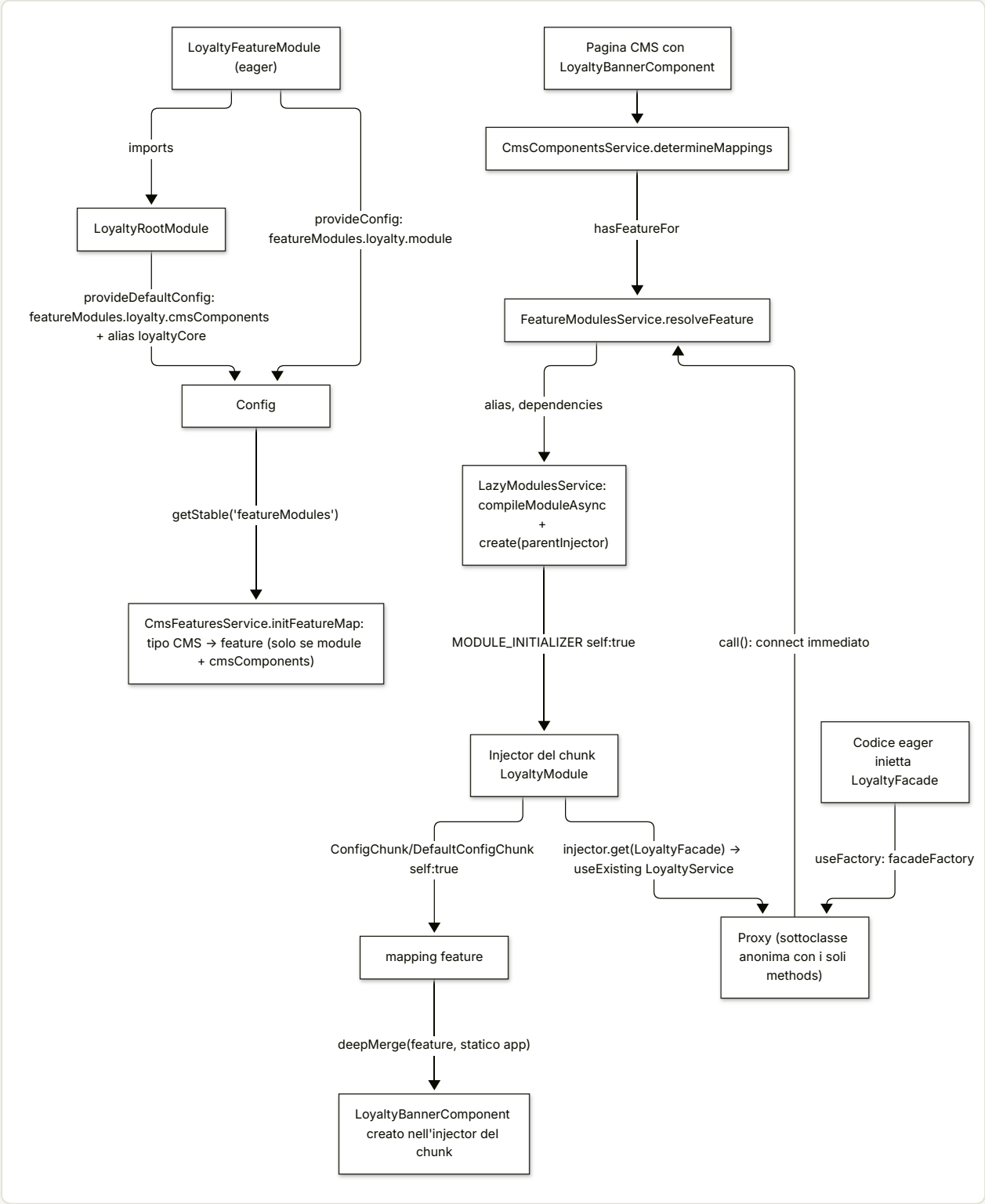


TAVOLA 53.2 **4.2 Riga carrello estesa e add-to-cart (Ricette B1, B2, B3)**

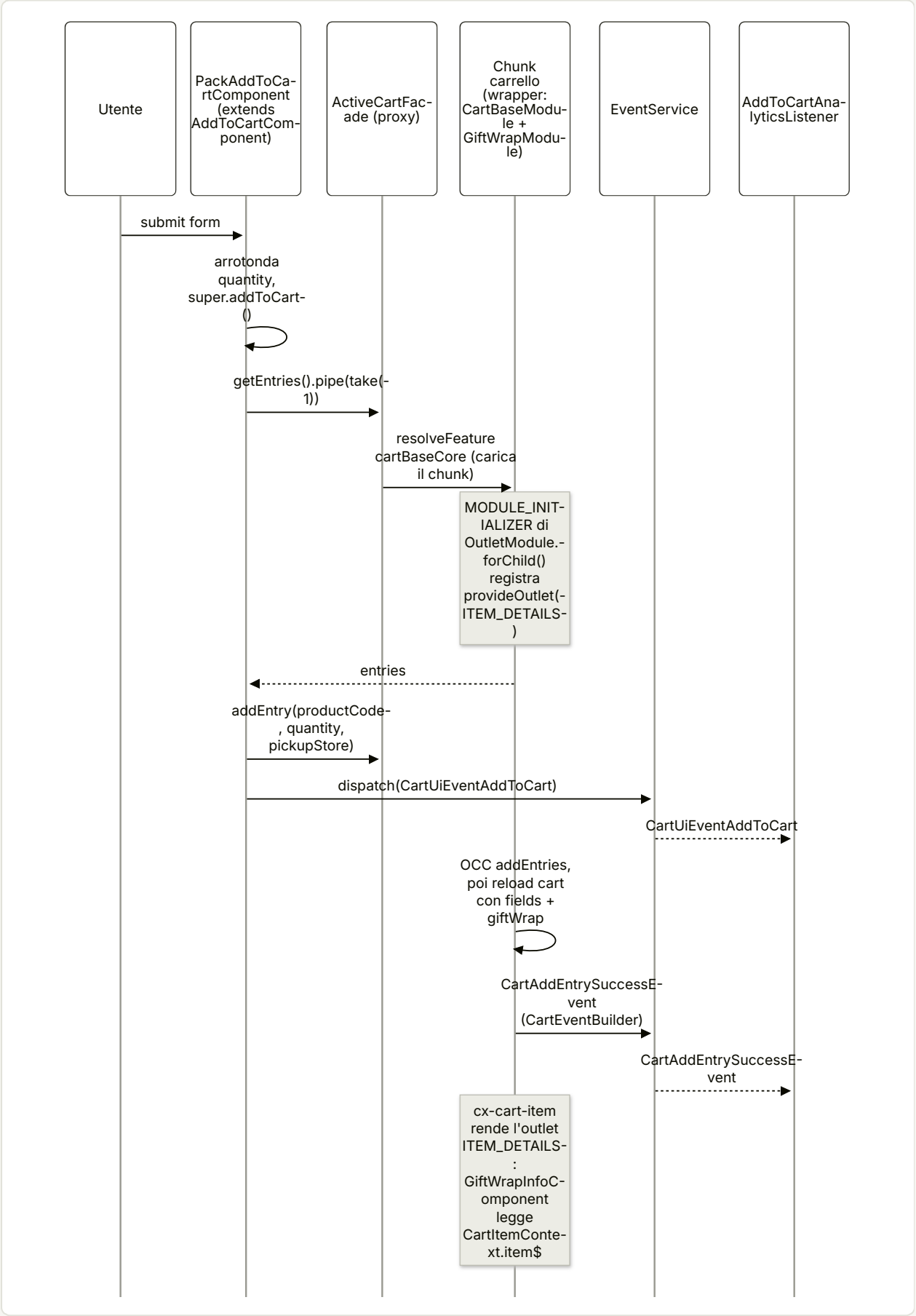


TAVOLA 53.3 **4.3 Nuovo step di checkout (Ricetta C1)**

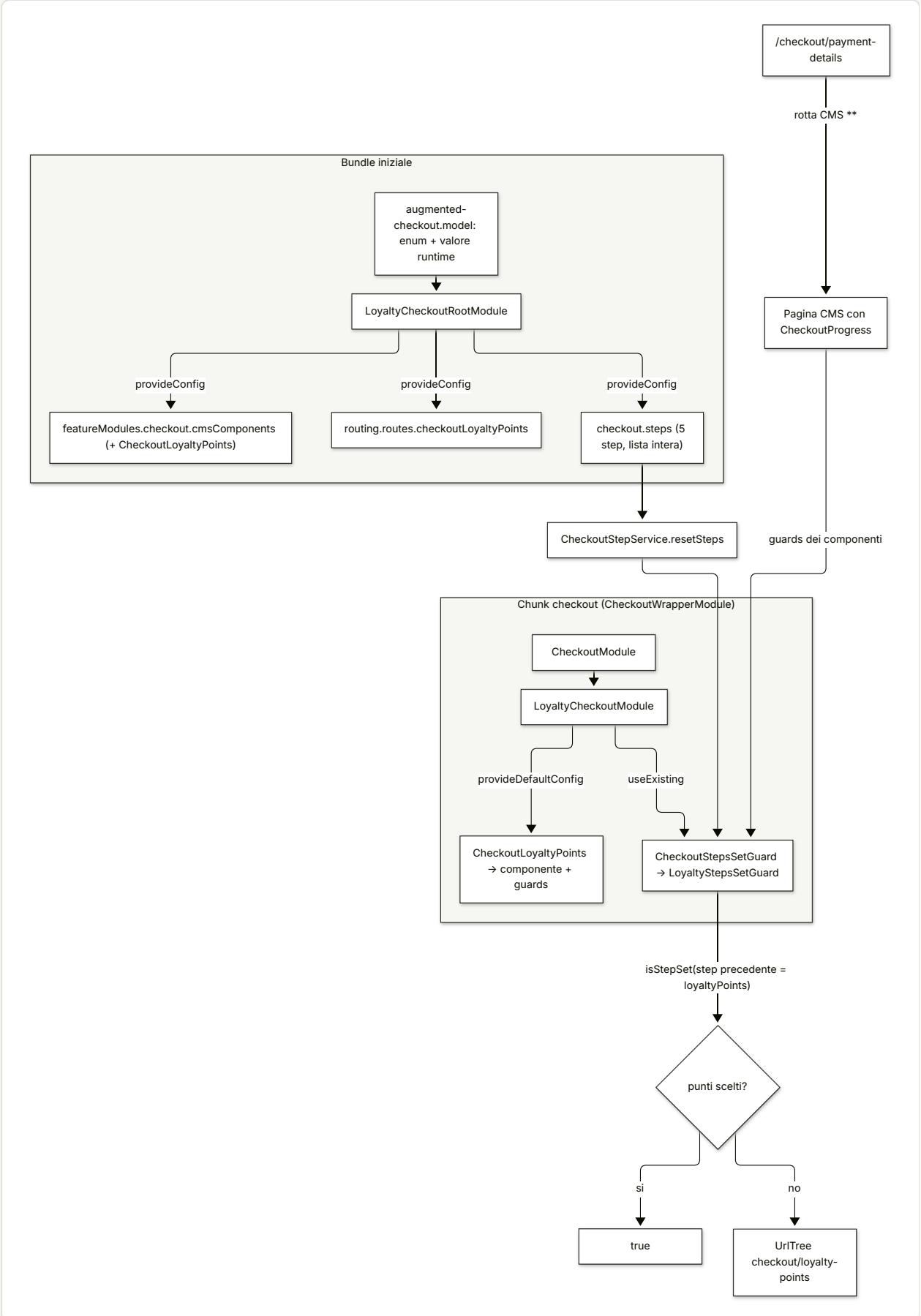
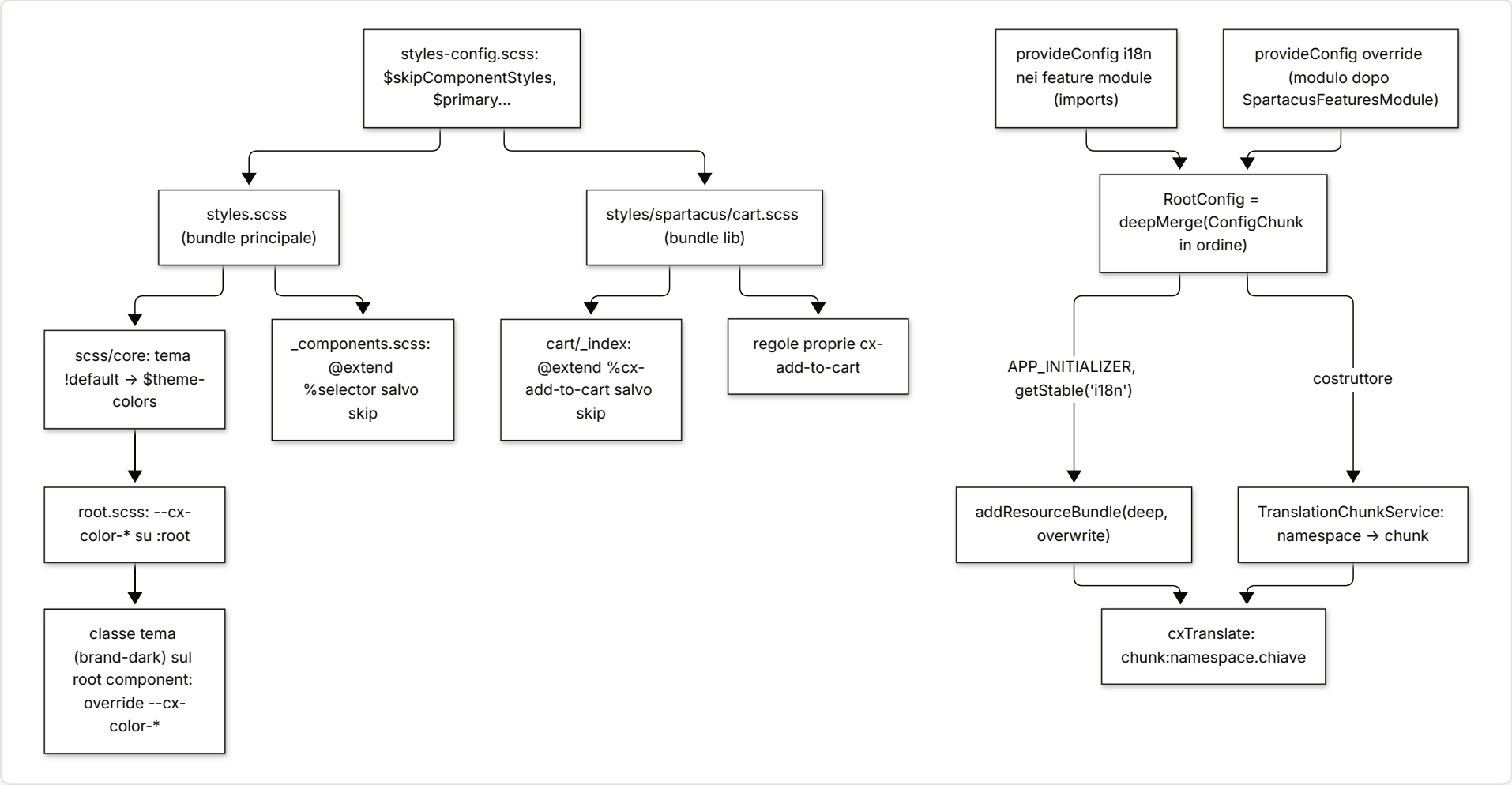


TAVOLA 53.4 **4.4 Stili e traduzioni: dove si decide il risultato (Ricette D1, D2, E1, E2)**



54

API pubblica vs interna, upgrade, anti-pattern, checklist upgrade-safe

La mia estensione sopravvivrà al prossimo `ng update`?

IN QUESTO CAPITOLO

1. A cosa serve e perché esiste
2. Mappa dei file
3. Come funziona passo per passo
4. Diagrammi
5. Estratti di codice reale
6. Configurazione
7. Commenti e note notevoli dal sorgente
8. Extension point
9. Trappole, casi limite, comportamenti controintuitivi
10. Come fare debug
11. Domande di autoverifica

Riferimenti allo SHA `5b7a1c2bf2808d7380bc20596d10f4fa710c3da1` (vedi 00-meta).

1. A cosa serve e perché esiste

Le ricette dei capitoli 50–53 insegnano a *estendere* Spartacus: config, DI, componenti CMS, adapter OCC, feature lazy. Questo capitolo chiude la Parte III e risponde alla domanda che viene dopo: **la mia estensione sopravvivrà al prossimo** `ng update`**?**

Il progetto si presenta come un insieme di librerie che "follows semantic versioning conventions ... upgradable for minor and patch releases" (`README.md:L31-L32`), con una "roll-forward update policy" (`docs/README.md:L5-L5`). La promessa vale però solo per una cosa precisa: la **superficie pubblica** dei pacchetti, cioè ciò che è raggiungibile dai file `public_api.ts` dichiarati come `entryFile` in ogni `ng-package.json` (`core-libs/core/ng-package.json:L1-L7`, `core-libs/core/public_api.ts:L7-L36`). Il resto, anche se sta in una classe `export`ata in un file sorgente, è implementazione e può cambiare senza preavviso.

Il repository rende questo confine verificabile con diversi strumenti:

- i barrel `index.ts`, che decidono cosa arriva al `public_api.ts` (F2, F28–F49 del dossier);
- un bot CI che fa il diff dei report di API Extractor a ogni PR (`.github/workflows/breaking-changes-detection-bot.yml:L1-L35`);
- un tool offline che confronta due versioni e genera i dati delle migrazioni `ng update` (`tools/breaking-changes/package.json:L1-L27`, `tools/breaking-changes/common.ts:L12-L19`);
- regole ESLint che proteggono l'estendibilità per sottoclasse (`eslint.config.mjs:L179-L192`, `tools/eslint-rules/index.ts:L86-L100`);
- feature toggle strict per introdurre gradualmente i breaking change (`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11`).

Il capitolo ha quattro parti operative: come riconoscere l'API pubblica, cosa si rompe davvero negli upgrade e perché, una serie di anti-pattern con la loro alternativa basata solo su API pubblica, e una checklist verificabile.

Capitoli collegati: il meccanismo dei toggle e delle deprecazioni è in 42 — Feature toggle e deprecazioni; il motore `ng update` e i mechanism di migrazione in 43 — Schematics e migrazioni. Le ricette che questo capitolo "mette in sicurezza" sono 50 — config, DI, interceptor, eventi, 51 — CMS, outlet, layout, 52 — OCC e dati e 53 — feature lazy, carrello/checkout, stili, i18n.

> Nota sulle versioni: i pacchetti libreria allo SHA sono `221121.22.0` (`core-libs/core/package.json:L3-L3`), la root del monorepo `2612.0.0` (`package.json:L3-L3`). Le migrazioni `ng update` usano la numerazione dei pacchetti (`core-libs/schematics/src/migrations/migrations.json:L103-L122`).

2. Mappa dei file

PATH	RUOLO
`core-libs/core/ng-package.json`	dichiara `entryFile: ./public_api.ts` e `dest` del pacchetto `@spartacus/core`
`core-libs/core/public_api.ts`	superficie pubblica di core: 27 barrel + blocco `AUGMENTABLE_TYPES_START..._END`

PATH	RUOLO
core-libs/storefront/public_api.ts	superficie pubblica di storefront: 11 barrel + ICON_TYPE, BREAKPOINT, LAUNCH_CALLER augmentabili
core-libs/setup/public_api.ts, core-libs/setup/ssr/public_api.ts	@spartacus/setup e l'entry point secondario SSR
feature-libs/**/ng-package.json, feature-libs/**/public_api.ts	entry point delle feature lib (@spartacus/<lib> e @spartacus/<lib>/<entry>): 217 ng-package.json sotto feature-libs; nel repo sono 314 in tutto (core-libs 5, feature-libs 217, integration-libs 92), entry point primari inclusi (conteggio con git ls-files <dir> | grep -c 'ng-package.json$')
docs/libs/library-structure.md	convenzione root/ (eager) vs core/, components/, occ/ (lazy)
tools/build-lib/declaration-merging/index.ts	builder usato dalle librerie: riscrive i declare module relativi nel .d.ts appiattito
tools/build-lib/augmented-types/index.ts	builder storico (registrato, non usato) per spostare il blocco augmentabile
tools/breaking-changes/*.ts, readme.md, api-extractor.json	pipeline offline: extract → parse → compare → gen dei dati di migrazione
tsconfig.api-extractor.json	tsconfig non-strict per API Extractor
.github/workflows/breaking-changes-detection-bot.yml, .github/api-extractor-action/**	bot PR che commenta i diff dell'API
core-libs/schematics/package.json	aggancio ng-update.migrations + packageGroup
core-libs/schematics/src/migrations/migrations.json	25 migrazioni registrate per versione
core-libs/schematics/src/migrations/221121_7/*/data/generated-*.ts	dati generati di deprecazione (costruttori, membri, API rimosse, rename)
core-libs/schematics/src/migrations/mechanism/**	motori generici delle migrazioni
core-libs/schematics/src/shared/utils/file-utils.ts	interfacce dei dati di deprecazione
core-libs/schematics/README.md	processo per aggiungere una migrazione
core-libs/core/src/features-config/**	feature toggle (dettaglio in cap. 42)
tools/eslint-rules/index.ts, tools/eslint-rules/rules/*.ts	regole custom legate all'estendibilità
eslint.config.mjs, projects/storefrontapp/eslint.config.mjs	wiring delle regole
core-libs/core/src/config/config-tokens.ts, core-libs/core/src/config/services/configuration.service.ts	costruzione e mutazione in-place del Config globale
docs/migration/221121_7/*.md	guide di migrazione manuali e generate
docs/libs/major-release-checklist.md	rimozione delle migrazioni della major precedente

3. Come funziona passo per passo

3.1 API pubblica vs interna

3.1.1 L'entry point è il contratto

Ogni pacchetto (o entry point secondario) ha un `ng-package.json` che punta a un `public_api.ts`:

- `@spartacus/core`: `entryFile: ./public_api.ts`, output `../../dist/core` (core-libs/core/ng-package.json:L1-L7).
- `@spartacus/assets` fa eccezione: `./src/public_api.ts` (core-libs/assets/ng-package.json:L1-L7).
- `@spartacus/setup` esporta solo `./recipes/index`; l'SSR è un entry point secondario separato con il proprio `public_api.ts` (core-libs/setup/public_api.ts:L7-L7, core-libs/setup/ssr/ng-package.json:L1-L6, core-libs/setup/ssr/public_api.ts:L7-L16).
- Le feature lib multi-entry hanno un `ng-package.json` per sotto-cartella, es. `@spartacus/cart/base/root` (feature-libs/cart/base/root/ng-package.json:L1-L6, feature-libs/cart/base/root/public_api.ts:L7-L16). Lo stesso vale per i singoli componenti: `@spartacus/cart/base/components/add-to-cart` ha `ng-package.json` e `public_api.ts` propri (feature-libs/cart/base/components/add-to-cart/ng-package.json:L1-L6, feature-libs/cart/base/components/add-to-cart/public_api.ts:L7-L8).

La convenzione di libreria è documentata: `root/` è EAGER (facade astratte, modelli, eventi, token, config), mentre `core/`, `components/`, `occ/` sono LAZY; "each entry point has own `public_api.ts`. Import as `@spartacus/<lib>/<entry-point>`" (docs/libs/library-structure.md:L4-L16). In sviluppo gli alias `@spartacus/*` del `tsconfig.json` puntano ai `public_api` sorgente, nella build di produzione dell'app a `dist/*` (tsconfig.json:L795-L796, projects/storefrontapp/tsconfig.app.prod.json:L508-L509).

Il `public_api.ts` di core è una lista di `export * from './src/<area>/index'` (core-libs/core/public_api.ts:L7-L36). Ne segue la regola pratica: **un simbolo è pubblico se e solo se la catena** `public_api.ts` **→** `index.ts` **→ … → file lo riesporta**. Un `export class` in un file non basta.

3.1.2 Come si riconosce l'interno

Spartacus *non* usa le convenzioni di Angular per marcare il privato:

- **Prefisso** `ɵ`: Angular riesporta simboli privati con alias `ɵ…` (angular:packages/core/src/core_private_export.ts:L8-L20). In Spartacus nessun simbolo lo usa; l'unica occorrenza non-spec è un separatore di stringa (core-libs/core/src/product/services/product-loading.service.ts:L88-L90).
- `@internal`: compare una sola volta, su `ConfigInitializerService.initialize()` "Not a part of a public API, used by APP_INITIALIZER", ma il metodo resta `public` nella classe esportata (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L130). È solo documentale.

I segnali reali sono tre.

(a) Assenza dal barrel. È il caso normale. Campione riverificato nel dossier (F36–F49):

SIMBOLO NON PUBBLICO	DEFINITO IN	BARREL CHE LO ESCLUDE	ALTERNATIVA PUBBLICA
`LazyLoadingModule`	`core-libs/core/src/lazy-loading/lazy-loading.module.ts`	core-libs/core/src/lazy-loading/index.ts:L7-L12	importato da `BaseCoreModule` (core-libs/core/src/base-core.module.ts:L41-L41)
`CombinedInjector`	core-libs/core/src/util/combined-injector.ts:L27-L27	core-libs/core/src/util/index.ts:L7-L29	—

SIMBOLO NON PUBBLICO	DEFINITO IN	BARREL CHE LO ESCLUDE	ALTERNATIVA PUBBLICA
TRANSFER_STATE_META_REDUCER, CX_KEY	CX_KEY in core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16; TRANSFER_STATE_META_REDUCER in core-libs/core/src/state/reducers/index.ts:L23-L25	core-libs/core/src/state/index.ts:L7-L11 (non riesporta reducers)	—
HttpErrorInterceptor	core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts	core-libs/core/src/global-message/http-interceptors/index.ts:L20-L22 (solo importato)	HttpErrorHandler + errorHandlers
provideMultiErrorHandler()	core-libs/core/src/error-handling/multi-error-handler/provide-multi-error-handler.ts:L11-L19	core-libs/core/src/error-handling/multi-error-handler/index.ts:L7-L8	MULTI_ERROR_HANDLER (core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L25-L25)
CouponSearchPageResolver	core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L33-L33	core-libs/core/src/product/services/index.ts:L7-L11	—
I18NEXT_LOGGER_PLUGIN	core-libs/core/src/i18n/i18next/i18next-plugins/i18next-logger-plugin.ts:L17-L17	core-libs/core/src/i18n/i18next/index.ts:L7-L10	—
WebComponentHandler	core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L30-L30	core-libs/storefront/cms-structure/page/component/handlers/index.ts:L7-L9	—
ContextToken	core-libs/storefront/context/context.model.ts:L7-L7	core-libs/storefront/context/index.ts:L7-L8	—
AsmBindCartService	feature-libs/asm/core/facade/asm-bind-cart.service.ts:L16-L16	feature-libs/asm/core/facade/index.ts:L7-L9	AsmBindCartFacade (feature-libs/asm/root/facade/index.ts:L7-L7)
UserIdHttpHeaderInterceptor	feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L30-L30	feature-libs/asm/root/public_api.ts:L7-L14	—
CheckoutCartInterceptor	feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L27-L27	feature-libs/checkout/base/root/public_api.ts:L7-L16	—
SegmentRefsConfig	integration-libs/segment-refs/root/config/segment-refs-config.ts:L14-L14	integration-libs/segment-refs/root/public_api.ts:L7-L7	—

(b) Commenti espliciti nel sorgente. Il repo li usa in modo non uniforme ma riconoscibile:

- `// PRIVATE API` su `MergingSubject` (core-libs/core/src/event/utils/merging-subject.ts:L10-L10) e su `SiteContextRoutesHandler` (core-libs/core/src/site-context/services/site-context-routes-handler.ts:L21-L25);
- "functions below should not be exposed in public API" su `siteContextParamsProviders`, che sostituisce `UrlSerializer` (core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L18);
- `CmsRoutesImplService` "should be exposed in public API only after the refactor planned in issue 7070" (core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L33);
- il barrel di keyboard-focus "we're NOT exposing all it to the public API" e lascia commentati dieci sotto-barrel (core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L7-L27);
- il barrel dei toggle "don't export from './config/feature-toggles' as it contains private API" (core-libs/core/src/features-config/feature-toggles/index.ts:L7-L9);
- `@spartacus/setup/ssr` testing esporta "only deliberate public api members" (core-libs/setup/ssr/testing/index.ts:L7-L10).

(c) Deprecazione "di confine". `locationInitializedFactory` è esportata ma marcata "@deprecated since 2211.22 - should not be a public API" (core-libs/core/src/config/config-initializer/config-initializer.module.ts:L32-L39): è pubblica per inerzia e verrà tolta.

Un caso particolare: `@spartacus/schematics` ha un `public_api` che "is not intended as a public API for external use. It may change without deprecation." (core-libs/schematics/index.ts:L7-L13). Le utility che esporta servono agli schematics delle altre librerie Spartacus, non al codice dei clienti.

3.1.3 Le "zone grigie": membri protected e tipi augmentabili

Due elementi sono pubblici pur non essendo simboli a sé.

Membri `protected` delle classi esportate. Fanno parte del `.d.ts` della classe, e API Extractor li confronta membro per membro: un membro sparito produce `DELETED` (tools/breaking-changes/compare.ts:L208-L239). Il progetto li considera un punto di estensione voluto: la regola `no-restricted-syntax` dice "Injected dependencies should use "protected" instead of "private" to allow customers to extend the class" (eslint.config.mjs:L179-L192). Le eccezioni sono `FeatureConfigService` e `FeatureToggles`, che DEVONO essere `private` (tools/eslint-rules/rules/feature-config-service-must-be-private.ts:L11-L19, tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L11-L26). Allo SHA ci sono 1226 `protected … = inject(` contro 133 `private … = inject(`, di cui 26 non coperti dalle eccezioni, es. `SearchboxService.actionsSubject` e `AddToCartComponent.productAvailabilityService` (core-libs/core/src/product/facade/searchbox.service.ts:L19-L20, feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L120-L120).

Tipi augmentabili. Il `public_api.ts` di core chiude con un blocco `/** AUGMENTABLE_TYPES_START */ … /** AUGMENTABLE_TYPES_END */` che riesporta per nome i tipi pensati per la module augmentation: `AuthToken`, `Config`, `Address`, `B2BUnit`, `B2BUser`, `B2BUserRole`, `CostCenter`, `Facet`, `ProductSearchPage`, `Price`, `Product`, `Stock`, `BackendConfig`, `OccEndpoints`, `ProductScope`, `RoutingConfigDefinition`, `OccHttpErrorReason`, `OccHttpErrorType` (core-libs/core/public_api.ts:L38-L58). In storefront: `ICON_TYPE`, `BREAKPOINT`, `LAUNCH_CALLER` (core-libs/storefront/public_api.ts:L32-L36). Blocchi simili esistono in 13 `public_api.ts`, es. `Cart`, `DeliveryMode`, `OrderEntry` in `@spartacus/cart/base/root` (feature-libs/cart/base/root/public_api.ts:L18-L20).

Le librerie stesse usano questo canale: `@spartacus/checkout/b2b/root` estende `enum CheckoutStepType` e `interface CheckoutState` di `@spartacus/checkout/base/root` e poi aggiunge il valore runtime con un cast (feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23); `@spartacus/order/root` fa lo stesso su `LAUNCH_CALLER` (feature-libs/order/root/model/augmented.model.ts:L7-L20). Gli enum augmentabili sono `enum` normali (core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L93-L93) e una regola ESLint vieta i `const enum` (tools/eslint-rules/rules/no-const-enum.ts:L11-L24).

Lato build, il builder usato da core, storefront e altre 16 librerie è `./tools/build-lib:declaration-merging` (core-libs/core/project.json:L8-L13): dopo ng-packagr riscrive in `declare module './<typings>'` i `declare module` del `.d.ts` appiattito il cui nome **non** inizia con uno dei caratteri `@spartacu` (la regex usa la classe di caratteri `[^\@spartacus]`, non un prefisso negato): in pratica vengono riscritti solo i path relativi (`./`, `../`), il che esclude anche i moduli `@spartacus/*` (tools/build-lib/declaration-merging/index.ts:L108-L126). Il builder storico `augmented-types`, che spostava il blocco augmentabile nel `.d.ts` radice, è registrato ma non usato (tools/build-lib/builders.json:L1-L14, tools/build-lib/augmented-types/index.ts:L20-L32).
DA VERIFICARE se con ng-packagr 21 il blocco `AUGMENTABLE_TYPES_START…_END` sia ancora necessario: senza `node_modules/dist` non si può controllare il `.d.ts` prodotto.

3.2 Cosa si rompe negli upgrade

3.2.1 Le regole del tool: cosa è breaking

Il tool `tools/breaking-changes` (pacchetto privato basato su `@microsoft/api-extractor`, tools/breaking-changes/package.json:L1-L27) definisce in codice cosa conta come breaking change. Estrae l'API da ogni `dist/**/public_api.d.ts` (tools/breaking-changes/extract.ts:L17-L50), quindi **un cambiamento a un simbolo non pubblico non viene nemmeno visto**. Poi confronta vecchia e nuova versione:

- elemento della vecchia API assente nella nuova: lookup dei rename, poi ricerca di un "move" per nome; esito `RENAMED`, `MOVED` (entry point diverso) o `DELETED` (tools/breaking-changes/compare.ts:L39-L92);
- confronto per tipo: Interface/Class → membri, Enum, TypeAlias, Variable, Function (tools/breaking-changes/compare.ts:L137-L162);
- costruttori e metodi: breaking rimuovere parametri, aggiungere parametri non opzionali, cambiare il tipo di uno esistente; non breaking aggiungere parametri opzionali in coda (tools/breaking-changes/compare.ts:L284-L307);
- enum: breaking solo la rimozione di un valore (tools/breaking-changes/compare.ts:L349-L366);
- membro di classe/interfaccia sparito: `DELETED` (tools/breaking-changes/compare.ts:L208-L239).

Il readme aggiunge le revisioni manuali: costruttori deprecati con overload, attributi opzionali aggiunti alle classi astratte Config ("di solito non breaking"), TypeAlias (tools/breaking-changes/readme.md:L77-L97).

Il processo interno conferma il confine: il README di schematics dice "Check whether any of the changed files are exported in the public API. If no, then no further action is required." (core-libs/schematics/README.md:L163-L174). **Se dipendi da un simbolo non pubblico, nessuna migrazione e nessuna documentazione ti avviserà.**

3.2.2 Il bot sulle PR

Il workflow `API extractor` gira su `pull_request` opened/synchronize/reopened (.github/workflows/breaking-changes-detection-bot.yml:L1-L35): builda head e base, esegue API Extractor su ogni entry point in `dist` (.github/api-extractor-action/src/index.ts:L36-L49, .github/api-extractor-action/src/api-extractor.ts:L154-L185) e commenta la PR con il diff del report, "Entry point removed. Are you sure it was intentional?" oppure "New entry point. Initial public api" (.github/api-extractor-action/src/comment.ts:L189-L247). Il report usa un tsconfig `strict: false` (tsconfig.api-extractor.json:L1-L9).

3.2.3 Le migrazioni `ng update`

`@spartacus/schematics` registra `ng-update.migrations: ./src/migrations/migrations.json` con un `packageGroup` di tutti i pacchetti `@spartacus/*` (core-libs/schematics/package.json:L46-L88). Allo SHA ci sono 25 migrazioni: 10 per `2211.19.0`, 4 per `2211.36.0`, 10 per `221121.7.0`, 1 per `221121.22.0` (core-libs/schematics/src/migrations/migrations.json:L3-L127). Le quattro dedicate ai breaking change TypeScript

della `221121.7.0` sono constructor-deprecations, methods-and-properties-deprecations, removed-public-api-deprecations, rename-symbol (`core-libs/schematics/src/migrations/migrations.json:L103-L122`).

I dati sono generati dal tool ("Generated file, don't update manually", `core-libs/schematics/src/migrations/221121_7/constructor-deprecations/data/generated-constructor.migration.ts:L7-L9`). Le interfacce sono `ConstructorDeprecation`, `MethodPropertyDeprecation`, `DeprecatedNode`, `ConfigDeprecation`, `RenamedSymbol`, `ComponentData` (`core-libs/schematics/src/shared/utils/file-utils.ts:L70-L117`). Cosa fanno, in concreto:

CATEGORIA	CORREGGE IL CODICE?	RIFERIMENTO
Costruttori deprecati	sì, aggiunge/rimuove parametri	`core-libs/schematics/src/migrations/migrations.json:L103-L107`
Metodi/proprietà rimossi	no, inserisce `// TODO:Spartacus -`	`core-libs/schematics/src/migrations/221121_7/methods-and-properties-deprecations/data/generated-methods-and-properties.migration.ts:L12-L19`
API pubblica rimossa	no, inserisce `// TODO:Spartacus -` sopra l'import	`core-libs/schematics/src/migrations/mechanism/removed-public-api-deprecations/removed-public-api-deprecation.ts:L19-L53`
Rename/move di simboli	sì, riscrive l'import	`core-libs/schematics/src/migrations/221121_7/rename-symbol/data/generated-rename-symbols.migration.ts:L11-L17`

Volumi della `221121_7`: 3 costruttori, 48 metodi/proprietà, 49 simboli rimossi, 1 rename (F73 del dossier); il documento generato riporta 49 voci "has been removed and is no longer part of the public API" (`grep -c "has been removed and is no longer part of the public API" docs/migration/221121_7/generated-typescript-changes-doc.md` → 49; prima occorrenza `docs/migration/221121_7/generated-typescript-changes-doc.md:L13-L13`). Il marcatore comune è `TODO_SPARTACUS = 'TODO:Spartacus -'` (`core-libs/schematics/src/shared/constants.ts:L1145-L1145`).

La guida `221121_7` prescrive di aggiornare prima Angular a 20 e poi a 21, quindi `ng update @spartacus/schematics@221121.7`, con fallback manuale (`docs/migration/221121_7/migration.md:L16-L16`, `docs/migration/221121_7/migration.md:L96-L96`, `docs/migration/221121_7/migration.md:L108-L114`). A ogni major le migrazioni della major precedente vengono rimosse (`docs/libs/major-release-checklist.md:L15-L16`). DA VERIFICARE il comportamento di `ng update` su un salto di più major: la logica di selezione delle migrazioni è nell'Angular CLI, che non è nel repo.

3.2.4 Il caso di studio: `FeatureConfigService` reso private

Il manuale `221121_7` descrive il breaking change più istruttivo per chi estende per ereditarietà. `FeatureConfigService` "has been made **private** across all Spartacus components and services"; le sottoclassi che lo usavano come `protected` non compilano più e "should be injected directly within that subclass" (`docs/migration/221121_7/221121_7-typescript-manual.doc.md:L208-L213`). Esempi puntuali: `UnitAddressItemService` e `OrgUnitEffects` perdono `protected featureConfigService` (`docs/migration/221121_7/221121_7-typescript-manual.doc.md:L144-L150`); `AsmBindCartComponent` perde `featureConfig` e la migrazione inserisce solo un TODO (`core-libs/schematics/src/migrations/221121_7/methods-and-properties-deprecations/data/generated-methods-and-properties.migration.ts:L12-L19`).

Il motivo è scritto nelle regole ESLint introdotte di conseguenza: quando un toggle arriva a fine vita, "removing the property would become a Typescript breaking change for customers who reference it from a subclass" (`tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L11-L26`). In altre parole: Spartacus si toglie dalla superficie pubblica tutto ciò che sa di dover rimuovere.

3.2.5 Feature toggle: breaking change "a rate"

Le correzioni che cambiano DOM, stili o comportamento vengono introdotte dietro un toggle: "Properties for gradual adaptation to breaking changes in Spartacus via feature toggles. Each toggle has its own lifespan defined in the official docs." (core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L10-L25). L'interfaccia è volutamente STRICT perché "customers using a property that was recently removed, will know they have to adapt their code" (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11). In sorgente ci sono TODO come "Remove feature flag next major release." (feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.ts:L233-L233). DA VERIFICARE la tabella delle scadenze dei toggle: il JSDoc rimanda alla documentazione ufficiale, che non è nel repo.

Le API legacy `FeatureConfigService`, `features.level`, `FeatureLevelDirective` sono deprecate a favore di `FeatureToggles` (dettaglio in 42); `import/no-deprecated` segnala in lint l'uso di simboli `@deprecated` (eslint.config.mjs:L124-L124).

3.2.6 Sintesi: cosa si rompe

TIPO DI DIPENDENZA DELL'ESTENSIONE	RILEVATO DAL TOOL?	MIGRAZIONE?	RISCHIO
Simbolo pubblico rinominato/spostato	sì (`RENAMED`/`MOVED`)	sì, riscrive l'import	basso
Costruttore pubblico cambiato	sì	sì	basso
Metodo/proprietà pubblica o protected rimossa	sì (`DELETED`)	solo TODO	medio
Simbolo pubblico rimosso	sì	solo TODO	medio
Toggle rimosso	errore di compilazione (interfaccia strict)	no	medio
Simbolo non esportato (deep import, copia, `useClass` di una classe interna)	**no**	**no**	alto
Membro `private` o tipo non esportato usato da sottoclasse	**no**	**no**	alto
Mutazione del `Config` iniettato	no	no	alto (silenzioso)
CSS / DOM interno di un componente	via toggle, non via tool	no	medio

3.3 Anti-pattern

Gli esempi "sbagliato" sono codice cliente illustrativo, non preso dal repo. Le alternative usano solo simboli esportati, con il riferimento al barrel che li rende pubblici.

AP1 — Deep import da path interni

```ts
// SBAGLIATO: MergingSubject è "PRIVATE API" e non è nel barrel event/index.ts
import { MergingSubject } from '@spartacus/core/src/event/utils/merging-subject';

const merged = new MergingSubject<MyEvent>();
merged.add(source$);
```

Perché si rompe: il file è marcato `// PRIVATE API` (core-libs/core/src/event/utils/merging-subject.ts:L10-L10), il barrel esporta solo `cx-event` ed `EventService` (core-libs/core/src/event/index.ts:L7-L8), quindi né il

bot né le migrazioni lo seguono. In più il path `src/...` esiste solo nei sorgenti: nella build di produzione gli alias puntano a `dist/*` (projects/storefrontapp/tsconfig.app.prod.json:L508-L509). DA VERIFICARE se il `package.json` pubblicato da ng-packagr 21 abbia un campo `exports` che blocca del tutto i deep import (il sorgente di `@spartacus/core` non ne ha, core-libs/core/package.json:L1-L36).

```ts
// CORRETTO: EventService è pubblico e usa MergingSubject al suo interno
import { inject } from '@angular/core';
import { EventService } from '@spartacus/core';

const eventService = inject(EventService);
const unregister = eventService.register(MyEvent, source$);
// ... quando la sorgente non serve più:
unregister();
```

`EventService.register()` è il metodo pubblico per aggiungere sorgenti e restituisce la funzione di teardown (core-libs/core/src/event/event.service.ts:L60-L68); internamente usa proprio `MergingSubject` (core-libs/core/src/event/event.service.ts:L19-L19). Ricetta: 50 — eventi.

AP2 — Sostituire o estendere una classe non esportata

```ts
// SBAGLIATO: HttpErrorInterceptor non è pubblico
import { HttpErrorInterceptor } from '@spartacus/core/src/global-message/http-interceptors/http-error.interceptor';

@Injectable()
export class MyErrorInterceptor extends HttpErrorInterceptor { /* ... */ }
```

`HttpErrorInterceptor` è solo importato da `http-interceptors/index.ts`, che espone invece `errorHandlers` e `httpErrorInterceptors` (core-libs/core/src/global-message/http-interceptors/index.ts:L20-L22). Il punto di estensione progettato è `HttpErrorHandler`, una classe astratta pubblica con `responseStatus`, `abstract handleError(...)`, `hasMatch()` e `abstract getPriority?()` (core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L15-L44, core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L61-L61), registrata come multi provider (core-libs/core/src/global-message/http-interceptors/index.ts:L22-L33).

```ts
// CORRETTO: un handler in più sul multi provider pubblico
import { Injectable } from '@angular/core';
import {
  GlobalMessageType,
  HttpErrorHandler,
  HttpResponseStatus,
  Priority,
} from '@spartacus/core';

@Injectable({ providedIn: 'root' })
export class MyGatewayTimeoutHandler extends HttpErrorHandler {
  responseStatus = HttpResponseStatus.GATEWAY_TIMEOUT;
```

```
  handleError() {
    this.globalMessageService.add(
      { key: 'myHttpHandlers.gatewayTimeout' },
      GlobalMessageType.MSG_TYPE_ERROR
    );
  }

  getPriority(): Priority {
    return Priority.NORMAL; // > Priority.LOW dell'handler OOTB
  }
}

// providers dell'app:
{ provide: HttpErrorHandler, useExisting: MyGatewayTimeoutHandler, multi: true }
```

Lo schema ricalca `GatewayTimeoutHandler` (core-libs/core/src/global-message/http-interceptors/handlers/gateway/gateway-timeout.handler.ts:L13-L29); `Priority` e la risoluzione per priorità stanno in core-libs/core/src/util/applicable.ts:L7-L7 e core-libs/core/src/util/applicable.ts:L47-L87 (JSDoc della regola di priorità a L47-L61, implementazione di `resolveApplicable` a L63-L87). Il dettaglio della scelta dell'handler è in 41 — Errori e global message; ricetta: 50 — interceptor.

AP3 — Dipendere dall'implementazione concreta di una facade

```ts
// SBAGLIATO: AsmBindCartService è l'implementazione, non è nel barrel asm/core/facade
import { AsmBindCartService } from '@spartacus/asm/core/facade/asm-bind-cart.service';
```

`AsmBindCartService` non è esportato da feature-libs/asm/core/facade/index.ts:L7-L9; è registrato come `useExisting` della facade pubblica (feature-libs/asm/core/facade/facade-providers.ts:L23-L27). Il contratto è la facade astratta in `root/`, che si auto-fornisce con `facadeFactory` e carica la feature lazy al primo uso (feature-libs/asm/root/facade/asm-bind-cart.facade.ts:L13-L26).

```ts
// CORRETTO: la facade astratta di root/
import { inject } from '@angular/core';
import { AsmBindCartFacade } from '@spartacus/asm/root';

const asmBindCart = inject(AsmBindCartFacade);
asmBindCart.bindCart(cartId).subscribe();
```

Pattern facade/command/query: 20; feature lazy e facade: 12, ricetta 53.

AP4 — Usare membri ereditati che non sono contratto

```ts
// SBAGLIATO: usare dipendenze del padre che sono (o diventeranno) private
export class MyAddToCartComponent extends AddToCartComponent {
  ngOnInit() {
    super.ngOnInit();
    // private nel padre: non compila
    this.productAvailabilityService /* ... */;
```

```
  }
}

export class MyUnitAddressItemService extends UnitAddressItemService {
  // protected fino a 221121.7, poi rimosso: non compila più
  isX = this.featureConfigService.isEnabled('...');
}
```

`AddToCartComponent.productAvailabilityService` è `private` (feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L120-L120); `featureConfigService` è sparito dalle sottoclassi con la `221121_7` (docs/migration/221121_7/221121_7-typescript-manual.doc.md:L144-L150).

```ts
// CORRETTO: la sottoclasse inietta da sé ciò che le serve, dal public API
import { inject } from '@angular/core';
import { FeatureToggles, ProductAvailabilityService } from '@spartacus/core';
import { AddToCartComponent } from '@spartacus/cart/base/components/add-to-cart';

export class MyAddToCartComponent extends AddToCartComponent {
  protected myAvailability = inject(ProductAvailabilityService);
  private featureToggles = inject(FeatureToggles); // private, come nel core
}
```

`ProductAvailabilityService` è pubblico via core-libs/core/src/product/facade/index.ts:L14-L14; `FeatureToggles` via core-libs/core/src/features-config/index.ts:L10-L10. È esattamente l'azione richiesta dal manuale: "inject `FeatureConfigService` directly within that subclass as a `private` property" (docs/migration/221121_7/221121_7-typescript-manual.doc.md:L147-L147), oggi preferibilmente con `FeatureToggles`, letto come proprietà come fa `PaginationComponent` (core-libs/storefront/shared/components/list-navigation/pagination/pagination.component.ts:L64-L67, core-libs/storefront/shared/components/list-navigation/pagination/pagination.component.ts:L120-L120). Ricetta per sostituire componenti CMS: 51.

Variante della stessa trappola: `LaunchDialogService` (pubblico) espone `protected autoFocusService` di tipo `AutoFocusService`, che non è esportato (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L17-L17, core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L30-L30, core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L17-L17). La sottoclasse vede il membro ma non può nominarne il tipo dal public API: va trattato come non contrattuale.

AP5 — Mutare il `Config` iniettato

```ts
// SBAGLIATO: muta l'oggetto globale condiviso
const config = inject(Config);
config.backend!.occ!.baseUrl = 'https://api.example.com';
```

`Config` è costruito una volta con `deepMerge({}, DefaultConfig, RootConfig)` (core-libs/core/src/config/config-tokens.ts:L10-L21) ed è lo stesso oggetto che `ConfigurationService` muta in-place con `deepMerge(this.config, newConfig)` quando arrivano chunk lazy, a meno che `disableConfigUpdates` sia attivo (core-libs/core/src/config/services/configuration.service.ts:L92-L106). Una mutazione applicativa può essere sovrascritta da un chunk lazy successivo, o diventare inefficace quando il "compatibility mechanism" verrà tolto: non c'è nessun contratto né migrazione che la copra.

```ts
// CORRETTO: contribuire config con i provider pubblici
import { provideConfig, provideConfigFactory } from '@spartacus/core';

providers: [
  provideConfig({ backend: { occ: { baseUrl: 'https://api.example.com' } } }),
  // oppure, se dipende da altre dipendenze:
  provideConfigFactory(() => ({ backend: { occ: { prefix: '/occ/v2/' } } })),
]
```

`provideConfig`/`provideConfigFactory` producono multi provider su `ConfigChunk` (core-libs/core/src/config/config-providers.ts:L18-L28, core-libs/core/src/config/config-providers.ts:L37-L49); `ConfigChunk` "Should not be used directly, use `provideConfig`..." (core-libs/core/src/config/config-tokens.ts:L49-L53). Le chiavi `backend.occ.baseUrl`/`prefix` sono in core-libs/core/src/occ/config/occ-config.ts:L13-L16. Per *leggere* la config che cambia con i moduli lazy si usa `ConfigurationService.unifiedConfig$` (core-libs/core/src/config/services/configuration.service.ts:L24-L34). Meccanica completa in 11 — Configurazione; ricetta: 50.

AP6 — `provideConfig` / `provideFeatureToggles` dentro una libreria riusabile

```ts
// SBAGLIATO (in una libreria condivisa tra più app)
@NgModule({ providers: [provideConfig({ myLib: { pageSize: 10 } })] })
export class MyLibModule {}
```

Una libreria che usa `provideConfig` scrive nella config *root*, togliendo all'app la possibilità di sovrascriverla in modo prevedibile. Lo dicono il JSDoc ("To provide default configuration in libraries provideDefaultConfig should be used instead.", core-libs/core/src/config/config-providers.ts:L11-L17), il token (core-libs/core/src/config/config-tokens.ts:L54-L59) e la regola `use-provide-default-config`, `error` su `**/*.module.ts` (tools/eslint-rules/rules/use-provide-default-config.ts:L33-L36, eslint.config.mjs:L231-L240). Lo stesso per i toggle (core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L16-L31).

```ts
// CORRETTO
import { provideDefaultConfig, provideDefaultFeatureToggles } from '@spartacus/core';

@NgModule({
  providers: [
    provideDefaultConfig({ myLib: { pageSize: 10 } }),
    provideDefaultFeatureToggles({ /* toggle della libreria */ }),
  ],
})
export class MyLibModule {}
```

(`myLib` va dichiarato con module augmentation di `Config`, vedi AP9.) Ricetta: 50.

AP7 — Chiamare API marcate `@internal`

```ts
// SBAGLIATO: initialize() è @internal, usato dall'APP_INITIALIZER
await inject(ConfigInitializerService).initialize([myInitializer]);
```

Il metodo è pubblico solo per necessità tecniche: "Not a part of a public API, used by APP_INITIALIZER to initialize all provided CONFIG_INITIALIZERS" (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L130).

```ts
// CORRETTO: registrare un ConfigInitializer e attendere la stabilità
import { inject, Injectable } from '@angular/core';
import {
  ConfigInitializer,
  ConfigInitializerService,
  provideConfigInitializer,
} from '@spartacus/core';

@Injectable({ providedIn: 'root' })
export class MyConfigInitializer implements ConfigInitializer {
  scopes = ['myLib.settings'];
  configFactory = async () => ({ myLib: { settings: await loadSettings() } });
}

providers: [provideConfigInitializer(MyConfigInitializer)];

// dove serve leggere la config già risolta:
inject(ConfigInitializerService).getStable('myLib.settings').subscribe(/* ... */);
```

`ConfigInitializer` (`scopes`, `configFactory`) e `provideConfigInitializer` sono in core-libs/core/src/config/config-initializer/config-initializer.ts:L17-L33 e core-libs/core/src/config/config-initializer/config-initializer.ts:L45-L51, esportati da core-libs/core/src/config/config-initializer/index.ts:L7-L14; `getStable()` in core-libs/core/src/config/config-initializer/config-initializer.service.ts:L50-L66. Bootstrap e initializer: 10.

AP8 — Aggirare la strictness dei feature toggle

```ts
// SBAGLIATO: silenziare l'errore su un toggle rimosso
provideFeatureToggles({ someRemovedToggle: true } as any);
```

Il cast annulla l'unico avviso che Spartacus dà quando un toggle arriva a fine vita: l'interfaccia è strict proprio perché "customers using a property that was recently removed, will know they have to adapt their code" (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11). Dopo la rimozione il comportamento "nuovo" è l'unico rimasto, quindi la chiave non fa più nulla.

```ts
// CORRETTO: togliere la chiave e adeguare l'estensione al nuovo comportamento
import { provideFeatureToggles } from '@spartacus/core';

provideFeatureToggles({
  // solo chiavi ancora presenti in FeatureToggles: il compilatore lo verifica
  a11yPaginationKeyboardNavigation: true,
});
```

`provideFeatureToggles` è per l'app e va nel root injector (core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L16-L31); l'elenco delle chiavi è in A09.

AP9 — Augmentation su path interni o di tipi non previsti

```ts
// SBAGLIATO: path interno e const enum
declare module '@spartacus/storefront/layout/launch-dialog/config/launch-config' {
  const enum LAUNCH_CALLER { MY_DIALOG = 'MY_DIALOG' }
}
```

Il path non è un entry point: non esiste un `ng-package.json` per quel path, quindi TypeScript non risolve il modulo e l'augmentation non si fonde con il `LAUNCH_CALLER` pubblico, che è esportato dall'entry point `@spartacus/storefront` nel blocco augmentabile (core-libs/storefront/public_api.ts:L32-L36). Il builder `declaration-merging` riguarda solo i `declare module` relativi interni al `.d.ts` della libreria, in fase di build (tools/build-lib/declaration-merging/index.ts:L108-L126), e non tocca il codice del cliente. I `const enum` sono vietati perché incompatibili con `isolatedModules` (tools/eslint-rules/rules/no-const-enum.ts:L11-L24).

```ts
// CORRETTO: entry point pubblico + valore runtime, come fanno le librerie Spartacus
import { LAUNCH_CALLER } from '@spartacus/storefront';

declare module '@spartacus/storefront' {
  enum LAUNCH_CALLER {
    MY_DIALOG = 'MY_DIALOG',
  }
}

(LAUNCH_CALLER as any)['MY_DIALOG'] = 'MY_DIALOG';
```

È lo schema di feature-libs/order/root/model/augmented.model.ts:L7-L20 e feature-libs/quote/root/model/augmented-types.ts:L10-L23. Augmenta solo i tipi nei blocchi `AUGMENTABLE_TYPES_START..._END` (core-libs/core/public_api.ts:L38-L58, core-libs/storefront/public_api.ts:L32-L36). Per `Config` l'equivalente è `declare module '@spartacus/core' { interface Config extends MyLibConfig {} }`; le librerie interne usano un path relativo (core-libs/core/src/occ/config/occ-config.ts:L46-L52) solo perché stanno *dentro* il pacchetto. Ricette: 50, 52 — endpoint OCC.

AP10 — Sostituire provider "di infrastruttura" non esportati

```ts
// SBAGLIATO: replicare a mano i provider interni del site context
providers: [
  { provide: UrlSerializer, useExisting: MyCopyOfSiteContextUrlSerializer },
]
```

`siteContextParamsProviders` sostituisce già `UrlSerializer` con `SiteContextUrlSerializer` ed è esplicitamente fuori dal public API (core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L18, core-libs/core/src/site-context/providers/index.ts:L7-L8). Una copia del comportamento interno diverge silenziosamente a ogni upgrade.

L'alternativa corretta è non toccare l'infrastruttura e agire sui punti pubblici: la configurazione del site context (`provideConfig({ context: { … } })`, vedi 13 — Site context e theme) e la ricetta di routing 51. Se un punto di estensione manca davvero, va chiesto al progetto (issue) invece di copiare codice interno: è il caso

di `CmsRoutesImplService`, che il team stesso dichiara "should be exposed in public API only after the refactor" (`core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L33`).

3.4 Checklist finale per estensioni upgrade-safe

Ogni voce è verificabile con un comando o una lettura puntuale.

Import e simboli

- [ ] Nessun import contiene `/src/`, `/lib/` o path di file: `grep -rnE "from '@spartacus/[^']*/(src|lib)/" src/` restituisce vuoto.
- [ ] Ogni simbolo importato da `@spartacus/<lib>[/<entry>]` è raggiungibile dal `public_api.ts` di quell'entry point (catena di barrel, §3.1.1).
- [ ] Nessun simbolo usato è marcato `// PRIVATE API`, "should not be exposed in public API", `@internal` o "@deprecated ... should not be a public API" nel sorgente (§3.1.2).
- [ ] Nessun import da `@spartacus/schematics` nel codice applicativo (`core-libs/schematics/index.ts:L7-L13`).
- [ ] `import/no-deprecated` attivo nel lint dell'app e zero warning sui simboli Spartacus (`eslint.config.mjs:L124-L124`).

DI ed estensione

- [ ] Le sostituzioni passano da classi/token pubblici: facade astratte di `root/`, `HttpErrorHandler`, `MULTI_ERROR_HANDLER`, adapter/converter pubblici (AP2, AP3).
- [ ] Le sottoclassi usano solo membri `protected`/`public` della classe padre e iniettano da sé tutto il resto (AP4).
- [ ] Nessuna sottoclasse usa membri della classe padre il cui tipo non sia importabile dal public API (`LaunchDialogService.autoFocusService`, AP4).
- [ ] `FeatureToggles` e `FeatureConfigService` iniettati nelle proprie classi come `private` (`eslint.config.mjs:L255-L256`, regole `@nx/workspace-feature-config-service-must-be-private` e `@nx/workspace-feature-toggles-must-be-private` a severità `error`; motivazione in `tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L11-L26`).
- [ ] Gli override di costruttore usano `inject()` o firme identiche all'ultima versione: dopo l'upgrade nessun `// TODO:Spartacus -` resta irrisolto (`grep -rn "TODO:Spartacus" src/` vuoto).

Configurazione e toggle

- [ ] Nessuna assegnazione su oggetti ottenuti da `inject(Config)` o `ConfigurationService.config` (AP5).
- [ ] Nelle librerie interne del progetto: solo `provideDefaultConfig*` e `provideDefaultFeatureToggles*`; nell'app: `provideConfig*` e `provideFeatureToggles` (AP6).
- [ ] Nessun `as any` / `@ts-ignore` su oggetti passati a `provideFeatureToggles` o `provideConfig` (AP8).
- [ ] Inizializzazione asincrona solo via `provideConfigInitializer` / `getStable()` (AP7).

Tipi

- [ ] Module augmentation solo su `declare module '@spartacus/<entry>'` e solo su tipi nei blocchi `AUGMENTABLE_TYPES_START..._END` (AP9).
- [ ] Nessun `const enum`; i valori aggiunti agli enum augmentati sono assegnati anche a runtime (AP9).

Processo di upgrade

- [ ] Aggiornamento in sequenza: prima Angular, poi `ng update @spartacus/schematics@<versione>` (docs/migration/221121_7/migration.md:L106-L114). Poiché a ogni major le migrazioni precedenti vengono rimosse (docs/libs/major-release-checklist.md:L15-L15), conviene passare per ogni major `DA VERIFICARE`: la logica con cui `ng update` seleziona le migrazioni su un salto di più major sta nella Angular CLI, non in questo repo.
- [ ] Dopo `ng update`: risolti tutti i `// TODO:Spartacus -`, letti il manuale TypeScript e il documento generato della versione (`docs/migration/<v>/`).
- [ ] Rivisti i toggle disattivati esplicitamente nell'app: ognuno ha un owner e un piano di attivazione prima della major che lo rimuoverà.
- [ ] Test (unit + e2e) delle estensioni eseguiti con tutti i toggle a `true`, come fa la storefrontapp (tools/eslint-rules/rules/no-storefrontapp-false-feature-toggles.ts:L33-L40).

---

4. Diagrammi

4.1 Dal sorgente alla superficie pubblica

■ **TAVOLA 54.1** diagramma a tutta pagina, in fondo al capitolo

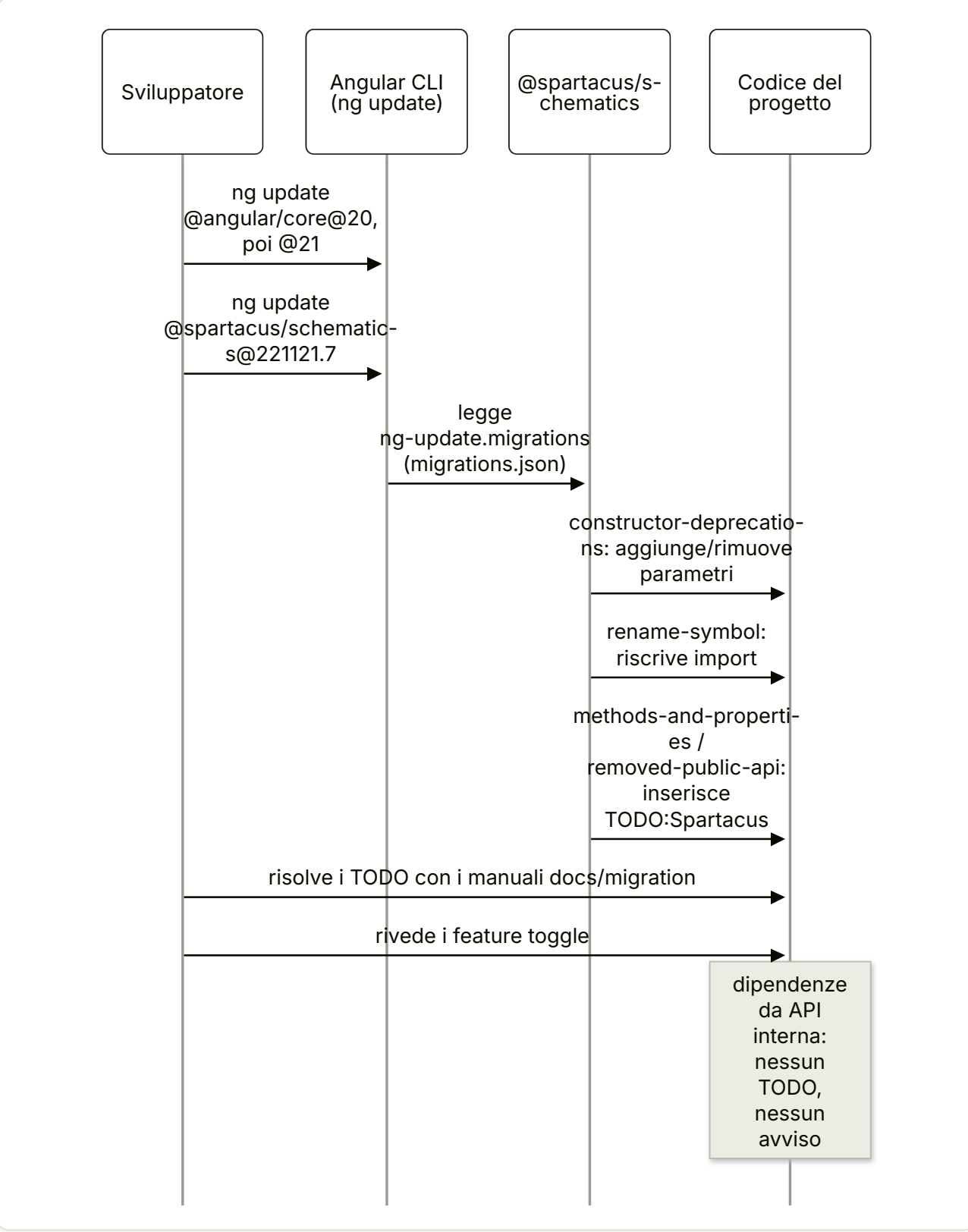
Riferimenti: barrel e `public_api.ts` (core-libs/core/public_api.ts:L7-L36), entry file (core-libs/core/ng-package.json:L1-L7), executor (core-libs/core/project.json:L8-L13, tools/build-lib/declaration-merging/index.ts:L108-L126), estrazione (tools/breaking-changes/extract.ts:L17-L50), generazione (tools/breaking-changes/common.ts:L12-L19), regola "nessuna azione se non è public API" (core-libs/schematics/README.md:L163-L174).

4.2 Classificazione dei cambiamenti nel tool di confronto

■ **TAVOLA 54.2** diagramma a tutta pagina, in fondo al capitolo

Riferimenti: tools/breaking-changes/compare.ts:L39-L92, tools/breaking-changes/compare.ts:L208-L239, tools/breaking-changes/compare.ts:L284-L307, tools/breaking-changes/compare.ts:L349-L366.

4.3 Upgrade lato progetto



Riferimenti: docs/migration/221121_7/migration.md:L16-L16, docs/migration/221121_7/migration.md:L108-L114, core-libs/schematics/package.json:L46-L88, core-libs/schematics/src/migrations/migrations.json:L103-L122, core-libs/schematics/src/migrations/mechanism/removed-public-api-deprecations/

removed-public-api-deprecation.ts:L19-L53. La logica con cui la CLI seleziona le migrazioni per versione è DA VERIFICARE (Angular CLI non nel repo).

---

5. Estratti di codice reale

Il blocco augmentabile di core — core-libs/core/public_api.ts:L38-L58

```ts
/** AUGMENTABLE_TYPES_START */
export { AuthToken } from './src/auth/user-auth/models/auth-token.model';
export { Config } from './src/config/config-tokens';
export { Address } from './src/model/address.model';
// ...
export { OccEndpoints } from './src/occ/occ-models/occ-endpoints.model';
export { ProductScope } from './src/product/model/product-scope';
export { RoutingConfigDefinition } from './src/routing/configurable-routes/config/routing-config';
export {
  OccHttpErrorReason,
  OccHttpErrorType,
} from './src/util/occ-http-error-constants';
/** AUGMENTABLE_TYPES_END */
```

Il barrel che decide cosa non pubblicare — core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L7-L27

```ts
// given that we're likely going to refactor the directives, we're
// NOT exposing all it to the public API.
export * from './focus.directive';
export { FocusConfig, TrapFocus, TrapFocusType } from './keyboard-focus.model';
export * from './skip-focus.directive';
export * from './keyboard-focus.module';
export * from './focus-testing.module';
export * from './services/index';
export * from './keyboard-focus.utils';

// export * from './autofocus/index';
// export * from './base/index';
// ...
// export * from './keyboard-focus.model';
```

`@internal` **documentale su un metodo pubblico** — core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L130

```ts
  /**
   * @internal
   *
   * Not a part of a public API, used by APP_INITIALIZER to initialize all provided CONFIG_INITIALIZERS
```

```
   *
   */
  async initialize(initializers?: ConfigInitializer[]) {
```

Regola sui parametri — tools/breaking-changes/compare.ts:L284-L307

```ts
function isParametersBreakingChangeDetected(
  oldParameters: any[],
  newParameters: any[]
): boolean {
  // Removed params is a breaking change
  if (oldParameters.length > newParameters.length) {
    return true;
  }
  // Were parameter(s) added?
  // If they are not optional, it's a breaking change
  if (oldParameters.length < newParameters.length) {
    const extraParams = newParameters.slice(oldParameters.length);
    if (!extraParams.every((param: any) => param.isOptional)) {
      return true;
    }
  }
  // Detect changes in the existing parameters.
  for (let index = 0; index < oldParameters.length; index++) {
    if (!isSameTypeParameter(oldParameters[index], newParameters[index])) {
      return true;
    }
  }
  return false;
}
```

La migrazione "removed public api" commenta, non corregge — core-libs/schematics/src/migrations/mechanism/removed-public-api-deprecations/removed-public-api-deprecation.ts:L31-L53

```ts
    for (const removedNode of removedNodes) {
      // 'source' has to be reloaded after each committed change
      const source = getTsSourceFile(tree, sourcePath);
      const changes = insertCommentAboveImportIdentifier(
        sourcePath,
        source,
        removedNode.node,
        removedNode.importPath,
        buildComment(
          removedNode.comment ??
            `'${removedNode.node}' is no longer part of the public API. Please look into migration guide for more information`
        )
      );
```

```
      commitChanges(tree, sourcePath, changes, InsertDirection.RIGHT);
    }
  }

  return tree;
}

function buildComment(content: string): string {
  return `// ${TODO_SPARTACUS} ${content}\n`;
}
```

Protected per estendere, private per ciò che si rimuoverà — eslint.config.mjs:L179-L192

```
      'no-restricted-syntax': [
        'warn',
        {
          selector: 'MemberExpression[object.property.name="nativeElement"]',
          // ...
        },
        {
          selector:
            'PropertyDefinition[accessibility="private"][value.type="CallExpression"]
[value.callee.name="inject"]:not([value.arguments.0.name="FeatureConfigService"]):not([value.arguments.0
.name="FeatureToggles"])',
          message: `[Spartacus] Injected dependencies should use "protected" instead of "private" to
allow customers to extend the class.
            (Exceptions: FeatureConfigService, FeatureToggles — those MUST be private; see
@nx/workspace-feature-config-service-must-be-private and @nx/workspace-feature-toggles-must-be-
private.)`,
        },
      ],
```

La mutazione in-place del Config — core-libs/core/src/config/services/configuration.service.ts:L92-L106

```
  private emitUnifiedConfig(): void {
    const newConfig: Config = deepMerge(
      {},
      this.defaultConfig,
      this.ambientDefaultConfig,
      this.ambientConfig,
      this.rootConfig
    );
    (this.unifiedConfig$ as BehaviorSubject<Config>).next(newConfig);

    // compatibility mechanism, can be disabled with feature toggle
    if (!isFeatureEnabled(this.config, 'disableConfigUpdates')) {
```

```
      deepMerge(this.config as Record<string, unknown>, newConfig);
    }
  }
```

Augmentation da una feature lib — `feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23`

```ts
declare module '@spartacus/checkout/base/root' {
  enum CheckoutStepType {
    PAYMENT_TYPE = 'paymentType',
  }

  interface CheckoutState {
    costCenter?: CostCenter;
    purchaseOrderNumber?: string;
    paymentType?: PaymentType;
  }
}

(CheckoutStepType as any)['PAYMENT_TYPE'] = 'paymentType';
```

6. Configurazione

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
`lib.entryFile`	string	`./public_api.ts` (assets: `./src/public_api.ts`)	`core-libs/core/ng-package.json:L4-L6`, `core-libs/assets/ng-package.json:L4-L6`	ng-packagr (**DA VERIFICARE**: `node_modules` assente)
`dest`	string	`../../dist/core`	`core-libs/core/ng-package.json:L3-L3`	`tools/build-lib/declaration-merging/index.ts:L78-L81`
`ng-update.migrations`	path	`./src/migrations/migrations.json`	`core-libs/schematics/package.json:L46-L47`	Angular CLI `ng update` (**DA VERIFICARE**: CLI non nel repo)
`ng-update.packageGroup`	string[]	tutti i `@spartacus/*`	`core-libs/schematics/package.json:L48-L88`	Angular CLI
`NEW_VERSION`	string	`'221121_7'`	`tools/breaking-changes/common.ts:L12-L12`	`tools/breaking-changes/common.ts:L13-L18`
`SCHEMATICS_COMMENT_PREFIX`	string	`'// TODO:Spartacus -'`	`tools/breaking-changes/common.ts:L19-L19`	generatori `tools/breaking-changes/generate-*.ts`
`mainEntryPointFilePath`	path	`./public_api.d.ts`	`tools/breaking-changes/api-extractor.json:L48-L48`	API Extractor
`compiler.tsconfigFilePath`	path	`<projectFolder>/tsconfig.api-extractor.json`	`tools/breaking-changes/api-extractor.json:L66-L66`	API Extractor
`strict` (tsconfig API Extractor)	boolean	`false`	`tsconfig.api-extractor.json:L4-L6`	API Extractor

CHIAVE	TIPO	DEFAULT	DEFINITA IN	LETTA IN
costanti BASE_BRANCH_DIR, BUILD_DIR, REPORT_DIR	string	`branch-clone` / `dist` / `etc`	`.github/api-extractor-action/src/const.ts:L7-L9`	`.github/api-extractor-action/src/api-extractor.ts:L158-L161`
`features.disableConfigUpdates`	boolean (legacy)	non definito → mutazioni in-place attive	`core-libs/core/src/features-config/config/features-config.ts:L116-L116`; chiave accettata da `isFeatureEnabled` in `core-libs/core/src/features-config/utils/feature-config-utils.ts:L50-L50` (cap. 42)	`core-libs/core/src/config/services/configuration.service.ts:L103-L105`
`import/no-deprecated`	severità ESLint	`warn`	`eslint.config.mjs:L124-L124`	ESLint
`no-restricted-syntax` (inject protected)	severità ESLint	`warn`	`eslint.config.mjs:L179-L192`	ESLint
`@nx/workspace-feature-config-service-must-be-private`, `@nx/workspace-feature-toggles-must-be-private`	severità ESLint	`error`	`eslint.config.mjs:L255-L256`	ESLint
`@nx/workspace-no-self-public-api-import`	severità ESLint	`warn`	`eslint.config.mjs:L257-L257`	ESLint

Le regole ESLint custom sono parte del monorepo (`tools/eslint-rules/index.ts:L86-L100`): un progetto cliente non le eredita automaticamente.

---

7. Commenti e note notevoli dal sorgente

- "Public API Surface of core" — `core-libs/core/public_api.ts:L7-L9`.
- "@internal … Not a part of a public API, used by APP_INITIALIZER" — `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L128`.
- "@deprecated since 2211.22 - should not be a public API" — `core-libs/core/src/config/config-initializer/config-initializer.module.ts:L32-L34`.
- "// PRIVATE API" — `core-libs/core/src/event/utils/merging-subject.ts:L10-L10`, `core-libs/core/src/site-context/services/site-context-routes-handler.ts:L21-L21`.
- "functions below should not be exposed in public API" — `core-libs/core/src/site-context/providers/site-context-params-providers.ts:L12-L12`.
- "This service should be exposed in public API only after the refactor planned in issue 7070" — `core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L31-L31`.
- "we're NOT exposing all it to the public API" — `core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L7-L8`.
- "don't export from './config/feature-toggles' as it contains private API" — `core-libs/core/src/features-config/feature-toggles/index.ts:L7-L7`.
- "We want this interface to be STRICT and cause a compilation error when a removed property is used." — `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L9-L10`.
- "…not intended as a public API for external use. It may change without deprecation." — `core-libs/schematics/index.ts:L9-L10`.
- "Generated file, don't update manually." — `core-libs/schematics/src/migrations/221121_7/constructor-deprecations/data/generated-constructor.migration.ts:L7-L9`.

- "It's a workaround to make TS types augmentable" — tools/build-lib/augmented-types/index.ts:L29-L31.
- "To make sure we get an accurate diff, do not update the dependencies of this tool..." — tools/breaking-changes/readme.md:L3-L3.
- "//TODO: Remove file and reference to it when everything is adjusted to strict mode" — tsconfig.api-extractor.json:L2-L2.
- "compatibility mechanism, can be disabled with feature toggle" — core-libs/core/src/config/services/configuration.service.ts:L102-L102.

---

8. Extension point

Gli extension point "sicuri" sono quelli che il tool di confronto vede e che le migrazioni coprono. Riepilogo con la ricetta della Parte III che li applica:

EXTENSION POINT PUBBLICO	RIFERIMENTO	AL POSTO DI	RICETTA
`provideConfig` / `provideConfigFactory` / `provideDefaultConfig*`	core-libs/core/src/config/config-providers.ts:L18-L28, core-libs/core/src/config/config-providers.ts:L55-L55	mutare `Config` (AP5), `provideConfig` in libreria (AP6)	50
`provideConfigInitializer` + `getStable()`	core-libs/core/src/config/config-initializer/config-initializer.ts:L45-L51	`initialize()` `@internal` (AP7)	50
`EventService.register/get/dispatch`	core-libs/core/src/event/event.service.ts:L68-L68	`MergingSubject` (AP1)	50
`HttpErrorHandler` + multi provider	core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L15-L15	`HttpErrorInterceptor` (AP2)	50
`MULTI_ERROR_HANDLER`	core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L25-L25	`provideMultiErrorHandler()`	50
Facade astratte di `root/` (es. `AsmBindCartFacade`)	feature-libs/asm/root/facade/asm-bind-cart.facade.ts:L13-L26	servizi concreti di `core/` (AP3)	53
Componenti per entry point (`@spartacus/cart/base/components/add-to-cart`) + membri `protected`	feature-libs/cart/base/components/add-to-cart/public_api.ts:L7-L8	membri `private` / tipi non esportati (AP4)	51
Tipi del blocco `AUGMENTABLE_TYPES_START` (`Config`, `OccEndpoints`, `Product`, `LAUNCH_CALLER`, ...)	core-libs/core/public_api.ts:L38-L58, core-libs/storefront/public_api.ts:L32-L36	augmentation su path interni (AP9)	50, 52
`provideFeatureToggles` / `provideDefaultFeatureToggles`	core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L16-L31	cast su toggle rimossi (AP8)	53, 42
Migrazioni `ng update` e `// TODO:Spartacus -`	core-libs/schematics/src/migrations/migrations.json:L103-L122	aggiornare "a mano" saltando le major	43

9. Trappole, casi limite, comportamenti controintuitivi

1. **"È esportato dal file" non vuol dire pubblico.** `TRANSFER_STATE_META_REDUCER` è esportato da `state/reducers/index.ts`, ma `state/index.ts` non riesporta `reducers` (core-libs/core/src/state/reducers/index.ts:L20-L27, core-libs/core/src/state/index.ts:L7-L11). L'autocompletamento dell'IDE in un monorepo con alias ai sorgenti può comunque proporlo.
2. `@internal` **non nasconde nulla.** `initialize()` resta chiamabile (core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L130).
3. **Un membro protected può avere un tipo privato.** `LaunchDialogService.autoFocusService: AutoFocusService` (core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L30-L30).
4. **La regola "inject protected" è solo** `warn`**.** Esistono 26 dipendenze `private` non coperte dalle eccezioni, es. `ProductSearchByCodeEffect.productSearchConnector` (core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L25-L25, eslint.config.mjs:L179-L180). Non si può contare sul fatto che "tutto è protected".
5. **Da protected a private è un breaking change che la migrazione non corregge.** Inserisce solo un TODO (core-libs/schematics/src/migrations/221121_7/methods-and-properties-deprecations/data/generated-methods-and-properties.migration.ts:L12-L19).
6. **API Extractor lavora non-strict e vede solo le firme overload.** I costruttori deprecati devono dichiarare tutti gli overload (tsconfig.api-extractor.json:L4-L6, tools/breaking-changes/readme.md:L81-L87).
7. **Falsi positivi sui Config.** Un attributo opzionale aggiunto a una classe Config viene segnalato come breaking e va tolto a mano dal JSON (tools/breaking-changes/readme.md:L89-L91).
8. **Input mancanti non bloccano il tool.** Se mancano i JSON in `docs/migration/<v>/`, il tool usa set vuoti e lo segnala solo nel log (tools/breaking-changes/common.ts:L22-L31); allo SHA `docs/migration/221121_7/` non li contiene (T4 del dossier). I commenti custom nelle migrazioni dipendono da quei file.
9. **Documentazione interna disallineata.** Il readme del tool mostra `NEW_VERSION = '2211.19'` e `renamed-api.json` (tools/breaking-changes/readme.md:L9-L11, tools/breaking-changes/readme.md:L37-L37) mentre il codice usa `'221121_7'` e `renamed-api-mappings.json` (tools/breaking-changes/common.ts:L12-L17); il README di schematics cita cartelle `3_0` che non esistono più (core-libs/schematics/README.md:L138-L142). Fa fede il codice.
10. **Le utility di test sono metà pubbliche.** `I18nTestingModule` e `KeyboardFocusTestingModule` sono nel public API (core-libs/core/src/i18n/index.ts:L13-L13, core-libs/storefront/layout/a11y/keyboard-focus/index.ts:L13-L13); altre sono raggiungibili solo via alias di spec (core-libs/storefront/tsconfig.spec.json:L16-L31). Gli spec del progetto non devono importarle da `@spartacus/core/testing/*`.
11. **Il** `Config` **cambia dopo il bootstrap.** Chi ne tiene un riferimento lo vede cambiare quando arrivano chunk lazy (core-libs/core/src/config/services/configuration.service.ts:L92-L106); con `disableConfigUpdates` no: il comportamento dipende da un toggle.
12. **Salti di più major.** Le migrazioni della major precedente vengono tolte (docs/libs/major-release-checklist.md:L15-L16); DA VERIFICARE se `ng update` su un salto multiplo le esegua (logica nella CLI).

10. Come fare debug

- **"Questo simbolo è pubblico?"** Nel repo Spartacus, risali la catena dei barrel: `grep -rlw --include=index.ts --include=public_api.ts NomeSimbolo core-libs feature-libs integration-libs`. Poi controlla che

ogni `index.ts` trovato sia a sua volta riesportato, fino al `public_api.ts` (§3.1.1). Nel progetto cliente: apri il `.d.ts` del pacchetto in `node_modules/@spartacus/<lib>` e cerca il nome.

- **Trovare deep import nel progetto:** `grep -rnE "from '@spartacus/[^']*/(src|lib)/" src/`.
- **Trovare dipendenze da membri ereditati:** per ogni `extends <ClasseSpartacus>`, confronta i membri usati via `this.` con la dichiarazione della classe nella versione di destinazione; i membri `private` producono errore di compilazione, quelli rimossi pure.
- **Prima di un upgrade:** leggi `docs/migration/<v>/generated-typescript-changes-doc.md` e il manuale TypeScript della versione e cerca i nomi delle classi che estendi (es. `docs/migration/221121_7/221121_7-typescript-manual.doc.md:L77-L79`).
- **Dopo `ng update`:** `grep -rn "TODO:Spartacus" src/` deve tornare vuoto prima del merge (marcatore in `core-libs/schematics/src/shared/constants.ts:L1145-L1145`).
- **Capire perché un upstream ha rotto qualcosa:** cerca la PR della modifica; il commento del bot `API extractor` mostra il diff del report per entry point (`.github/api-extractor-action/src/comment.ts:L189-L247`). Se il bot non lo segnala, la modifica toccava API interna.
- **Mutazioni del Config:** metti un breakpoint (o un `Object.freeze` temporaneo in sviluppo) sull'oggetto restituito da `inject(Config)` per scoprire chi lo modifica; ricorda che Spartacus stesso lo modifica in `emitUnifiedConfig()` (`core-libs/core/src/config/services/configuration.service.ts:L92-L106`).
- **Toggle:** un errore TS su una chiave passata a `provideFeatureToggles` indica di norma un toggle rimosso (`core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11`); cerca la chiave nel manuale della versione e in A09.

---

11. Domande di autoverifica

1. **Una classe ha `export class` nel suo file. È API pubblica di `@spartacus/core`?** Non necessariamente: lo è solo se la catena di barrel la porta fino a `public_api.ts` (`core-libs/core/public_api.ts:L7-L36`). Esempio contrario: `MergingSubject` (`core-libs/core/src/event/utils/merging-subject.ts:L24-L24`, `core-libs/core/src/event/index.ts:L7-L8`).
2. **Spartacus usa il prefisso `ɵ` o `@internal` per marcare il privato?** No in pratica: nessun simbolo con `ɵ` (`core-libs/core/src/product/services/product-loading.service.ts:L88-L90`) e un solo `@internal`, puramente documentale (`core-libs/core/src/config/config-initializer/config-initializer.service.ts:L124-L130`). Il segnale reale è l'assenza dal barrel.
3. **Aggiungere un parametro opzionale in coda a un costruttore pubblico è breaking per il tool?** No; lo è rimuovere parametri, aggiungerne di obbligatori o cambiarne il tipo (`tools/breaking-changes/compare.ts:L284-L307`).
4. **Cosa fa la migrazione "removed public api" sul codice del cliente?** Inserisce sopra l'import un commento `// TODO:Spartacus - '<X>' is no longer part of the public API…`, senza correggere il codice (`core-libs/schematics/src/migrations/mechanism/removed-public-api-deprecations/removed-public-api-deprecation.ts:L31-L53`).
5. **Perché `FeatureToggles` e `FeatureConfigService` devono essere `private`, mentre le altre dipendenze iniettate `protected`?** Le seconde sono punti di estensione per le sottoclassi (`eslint.config.mjs:L179-L192`); le prime verranno rimosse quando i toggle scadono, e rimuovere una proprietà protected sarebbe un breaking change TS (`tools/eslint-rules/rules/feature-toggles-must-be-private.ts:L11-L26`).
6. **Ho una sottoclasse che usava `this.featureConfigService` del padre e dopo l'upgrade a 221121.7 non compila. Cosa faccio?** Inietto il servizio (oggi preferibilmente `FeatureToggles`) direttamente nella sotto-

classe come `private` (docs/migration/221121_7/221121_7-typescript-manual.doc.md:L208-L213, core-libs/storefront/shared/components/list-navigation/pagination/pagination.component.ts:L120-L120).

7. **Perché mutare l'oggetto `Config` iniettato è fragile?** Perché è lo stesso oggetto che `ConfigurationService` muta con `deepMerge` all'arrivo di chunk lazy, a seconda di `disableConfigUpdates` (core-libs/core/src/config/services/configuration.service.ts:L92-L106). La strada pubblica è `provideConfig`/`provideConfigFactory` (core-libs/core/src/config/config-providers.ts:L18-L49).
8. **Come aggiungo un valore a `LAUNCH_CALLER` in modo upgrade-safe?** `declare module '@spartacus/storefront' { enum LAUNCH_CALLER { … } }` più l'assegnazione runtime con cast, come `@spartacus/order/root` (feature-libs/order/root/model/augmented.model.ts:L7-L20); `LAUNCH_CALLER` è nel blocco augmentabile (core-libs/storefront/public_api.ts:L32-L36).
9. **Perché `provideFeatureToggles({ x: true } as any)` è un anti-pattern?** Annulla l'errore di compilazione che l'interfaccia strict produce apposta quando un toggle viene rimosso (core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L7-L11).
10. **Una modifica a un file non esportato richiede una migrazione secondo il processo interno?** No: "Check whether any of the changed files are exported in the public API. If no, then no further action is required." (core-libs/schematics/README.md:L163-L174). Per questo chi dipende da API interna non riceve né avvisi né migrazioni.

TAVOLA 54.1 **4.1 Dal sorgente alla superficie pubblica**

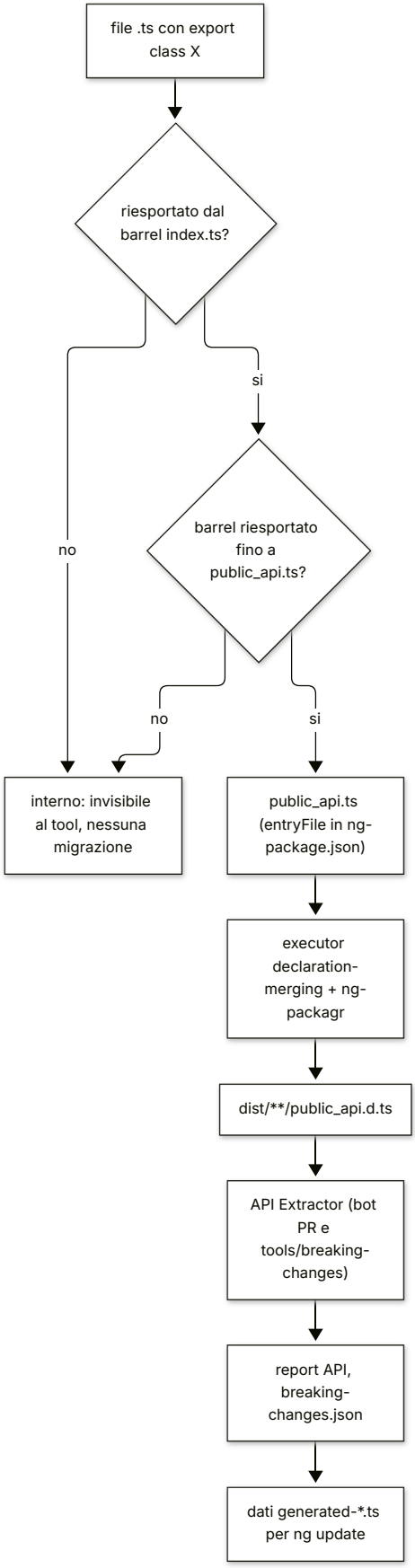
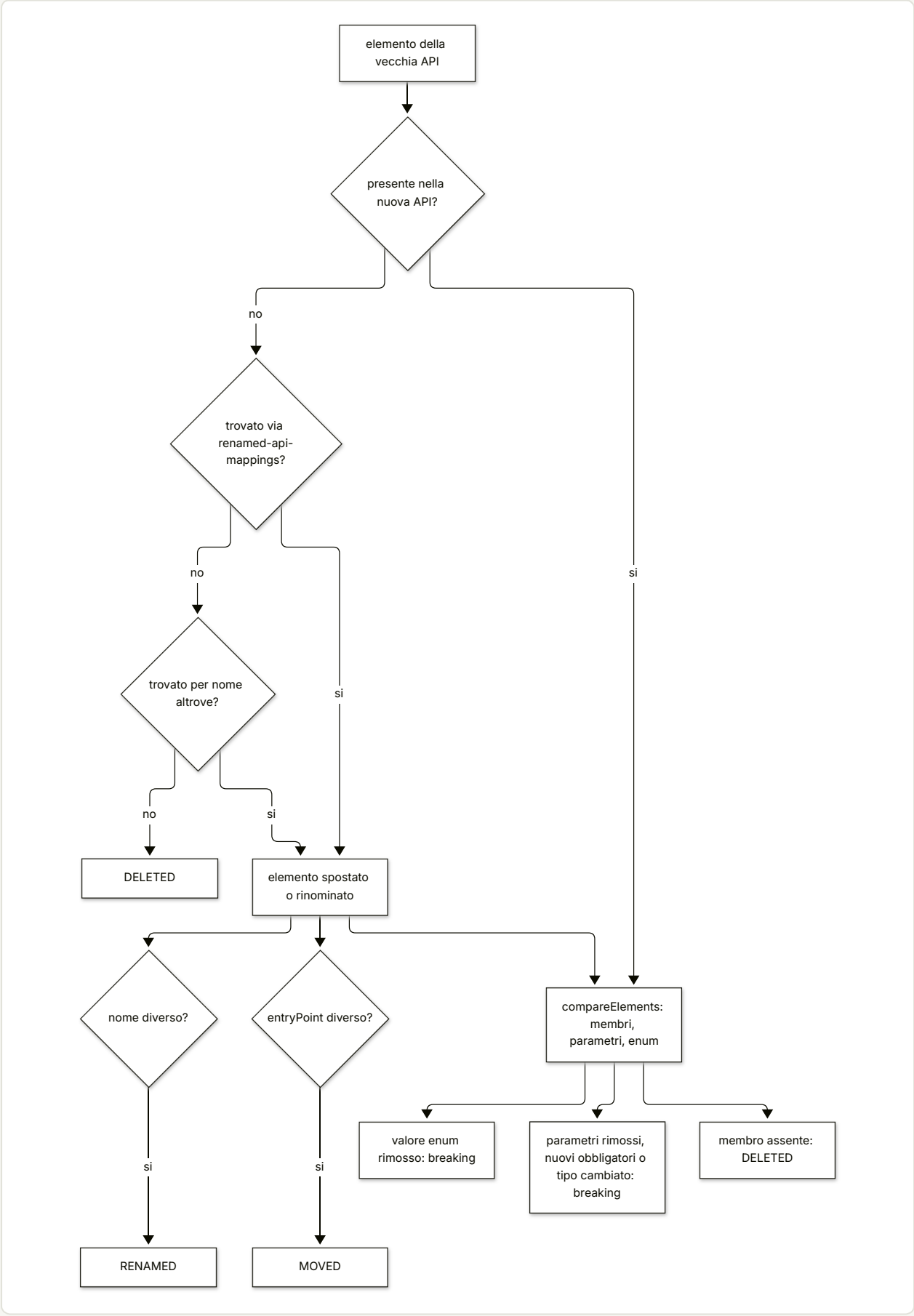


TAVOLA 54.2 **4.2 Classificazione dei cambiamenti nel tool di confronto**



Appendici

A01 InjectionToken

A02 Configurazione (`*Config`)

A03 Componenti CMS (`cmsComponents`)

A04 Endpoint OCC

A05 Guard

A06 HTTP interceptor

A07 Eventi (`CxEvent`)

A08 Outlet

A09 Feature toggle

A10 API deprecate (`@deprecated`)

A11 Glossario

A12 Indice dei simboli

A01

InjectionToken

Token DI dichiarati nel codice

IN QUESTO CAPITOLO

Conteggi per libreria

Elenco

Generata da `tools/extract/a01-injection-tokens.js` allo SHA 5b7a1c2b (`5b7a1c2bf2808d7380bc20596d10f4fa710c3da1`). Non modificare a mano.
Righe di tabella: **219**.
Rigenerazione: `cd docs/spartacus-book/tools/extract && npm install && node a01-injection-tokens.js`
Colonna *public API*: `sì` se la `const` è raggiungibile (anche via `export *`) dall'`entryFile` di almeno un `ng-package.json` (risoluzione ts-morph `getExportedDeclarations`). *export*: la `const` ha la keyword `export` nel file. Ambito: file `.ts` tracciati in core-libs, feature-libs, integration-libs esclusi spec, testing, mock e schematics.

Conteggi per libreria

LIBRERIA	TOKEN
@spartacus/asm	4
@spartacus/cart	13
@spartacus/cdc	1
@spartacus/cds	1
@spartacus/checkout	5
@spartacus/core	68
@spartacus/customer-ticketing	7
@spartacus/digital-payments	2
@spartacus/epd-visualization	2
@spartacus/opf	9
@spartacus/order	14
@spartacus/organization	29
@spartacus/pdf-invoices	1
@spartacus/pickup-in-store	3
@spartacus/product	2
@spartacus/product-configurator	20
@spartacus/punchout	2
@spartacus/quote	7
@spartacus/setup	4
@spartacus/storefinder	3
@spartacus/storefront	10
@spartacus/user	12
Totale	**219** (public API: 186)

Elenco

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
1	ASM_CUSTOMER_360_NORMALIZER	AsmCustomer360 Normalizer	Converter<any, AsmCustomer360Response>	@spartacus/asm	sì	sì	feature-libs/asm/customer-360/core/connectors/converters.ts:L11-L13
2	CUSTOMER_LISTS_NORMALIZER	CustomerLists Normalizer	Converter<any, CustomerListsPage>	@spartacus/asm	sì	sì	feature-libs/asm/core/connectors/converters.ts:L15-L17
3	CUSTOMER_SEARCH_PAGE_NORMALIZER	CustomerSearchPage Normalizer	Converter<any, CustomerSearchPage>	@spartacus/asm	sì	sì	feature-libs/asm/core/connectors/converters.ts:L11-L13
4	reducerToken	AsmReducers	ActionReducerMap<AsmState>	@spartacus/asm	sì	no	feature-libs/asm/core/store/reducers/index.ts:L36-L37
5	ActiveCartOrderEntriesContextToken	ActiveCartOrderEntriesContext		@spartacus/cart	sì	sì	feature-libs/cart/base/root/tokens/context.ts:L9-L11
6	CART_ACCESS_CODE_NORMALIZER	CartAccessCode Normalizer	Converter<any, string>	@spartacus/cart	sì	sì	feature-libs/cart/base/core/connectors/access-code/converters.ts:L10-L12
7	CART_GUEST_USER_NORMALIZER	CartGuestUser Normalizer	Converter<any, CartGuestUser>	@spartacus/cart	sì	sì	feature-libs/cart/base/core/connectors/guest-user/converters.ts:L11-L13
8	CART_MODIFICATION_NORMALIZER	CartModification Normalizer	Converter<any, CartModification>	@spartacus/cart	sì	sì	feature-libs/cart/base/root/tokens/converters.ts:L25-L27
9	CART_NORMALIZER	CartNormalizer	Converter<any, Cart>	@spartacus/cart	sì	sì	feature-libs/cart/base/root/tokens/converters.ts:L17-L19
10	CART_VALIDATION_NORMALIZER	CartValidation Normalizer	Converter<any, CartModificationList>	@spartacus/cart	sì	sì	feature-libs/cart/base/core/connectors/validation/converters.ts:L11-L13
11	CART_VOUCHER_NORMALIZER	CartVoucher Normalizer	Converter<any, Voucher>	@spartacus/cart	sì	sì	feature-libs/cart/base/root/tokens/converters.ts:L33-L35
12	multiCartReducerToken	MultiCartReducers	ActionReducerMap<MultiCartState>	@spartacus/cart	sì	no	feature-libs/cart/base/core/store/reducers/index.ts:L30-L32
13	NewSavedCartOrderEntriesContextToken	NewSavedCartOrderEntriesContext		@spartacus/cart	sì	sì	feature-libs/cart/saved-cart/root/tokens/context.ts:L13-L15
14	ORDER_ENTRY_PROMOTIONS_NORMALIZER	OrderEntry Promotions Normalizer	Converter<any, PromotionResult[]>	@spartacus/cart	sì	sì	feature-libs/cart/base/root/tokens/converters.ts:L21-L23
15	QuickOrderOrderEntriesContextToken	QuickOrderOrderEntriesContext		@spartacus/cart	sì	sì	feature-libs/cart/quick-order/root/tokens/context.ts:L9-L11

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
16	SAVE_CART_ NORMALIZER	SaveCartNormalizer	Converter<any, SaveCart Result>	@spartacus/cart	sì	sì	feature-libs/cart/base/root/tokens/converters.ts:L29-L31
17	SavedCartOrder EntriesContext Token	SavedCartOrder EntriesContext		@spartacus/cart	sì	sì	feature-libs/cart/saved-cart/root/tokens/context.ts:L9-L11
18	CDC_PREFERENCE_ SERIALIZER	CdcPreference Serializer	Converter<CdcConsent[], any>	@spartacus/cdc	sì	sì	integration-libs/cdc/root/consent-management/converters/converter.ts:L11-L13
19	LOGIN_EVENTS	LOGIN_EVENTS	Observable<LoginEvent Envelope>	@spartacus/cds	sì	no	integration-libs/cds/src/profiletag/tokens/login-events.token.ts:L20-L22
20	CHECKOUT_ NORMALIZER	CheckoutNormalizer	Converter<any, Checkout State>	@spartacus/checkout	sì	sì	feature-libs/checkout/base/core/connectors/checkout/converters.ts:L11-L13
21	CHECKOUT_PAYMENT_ TYPE_NORMALIZER	CheckoutPaymentType Normalizer	Converter<any, Payment Type>	@spartacus/checkout	sì	sì	feature-libs/checkout/b2b/core/connectors/checkout-payment-type/converters.ts:L11-L13
22	DELIVERY_MODE_ NORMALIZER	DeliveryMode Normalizer	Converter<any, Delivery Mode>	@spartacus/checkout	sì	sì	feature-libs/checkout/base/core/connectors/checkout-delivery-modes/converters.ts:L11-L13
23	PAYMENT_CARD_TYPE_ NORMALIZER	PaymentCardType Normalizer	Converter<any, CardType>	@spartacus/checkout	sì	sì	feature-libs/checkout/base/core/connectors/checkout-payment/converters.ts:L14-L16
24	PAYMENT_DETAILS_ SERIALIZER	PaymentDetails Serializer	Converter<PaymentDetails, any>	@spartacus/checkout	sì	sì	feature-libs/checkout/base/core/connectors/checkout-payment/converters.ts:L10-L12
25	ADDRESS_LIST_ NORMALIZER	AddressesNormalizer	Converter<any, Entities Model<Address>>	@spartacus/core	sì	sì	core-libs/core/src/user/connectors/address/converters.ts:L16-L18
26	ADDRESS_NORMALIZER	AddressNormalizer	Converter<any, Address>	@spartacus/core	sì	sì	core-libs/core/src/user/connectors/address/converters.ts:L12-L14
27	ADDRESS_SERIALIZER	AddressSerializer	Converter<Address, any>	@spartacus/core	sì	sì	core-libs/core/src/user/connectors/address/converters.ts:L20-L22
28	ADDRESS_ VALIDATION_ NORMALIZER	AddressValidation Normalizer	Converter<any, Address Validation>	@spartacus/core	sì	sì	core-libs/core/src/user/connectors/address/converters.ts:L24-L26
29	ANONYMOUS_CONSENT_ NORMALIZER	AnonymousConsent Normalizer	Converter<string, AnonymousConsent[]>	@spartacus/core	sì	sì	core-libs/core/src/anonymous-consents/connectors/converters.ts:L11-L13

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
30	BASE_SITE_NORMALIZER	BaseSiteNormalizer	Converter<any, BaseSite>	@spartacus/core	sì	sì	core-libs/core/src/site-context/connectors/converters.ts:L36-L38
31	CITY_DISTRICT_NORMALIZER	CityDistrict Normalizer	Converter<any, City District>	@spartacus/core	sì	sì	core-libs/core/src/site-context/connectors/converters.ts:L32-L34
32	CITY_NORMALIZER	CityNormalizer	Converter<any, City>	@spartacus/core	sì	sì	core-libs/core/src/site-context/connectors/converters.ts:L28-L30
33	CMS_COMPONENT_NORMALIZER	CmsComponent Normalizer	Converter<any, Cms Component>	@spartacus/core	sì	sì	core-libs/core/src/cms/connectors/component/converters.ts:L11-L13
34	CMS_PAGE_NORMALIZER	CmsPageNormalizer	Converter<any, Cms StructureModel>	@spartacus/core	sì	sì	core-libs/core/src/cms/connectors/page/converters.ts:L11-L13
35	CONFIG_INITIALIZER	ConfigInitializer	ConfigInitializer | null	@spartacus/core	sì	sì	core-libs/core/src/config/config-initializer/config-initializer.ts:L33-L35
36	CONFIG_INITIALIZER_FORROOT_GUARD	CONFIG_INITIALIZER_FORROOT_GUARD	void	@spartacus/core	sì	no	core-libs/core/src/config/config-initializer/config-initializer.ts:L37-L39
37	ConfigChunk	ConfigurationChunk	Config[]	@spartacus/core	sì	sì	core-libs/core/src/config/config-tokens.ts:L53-L53
38	ConfigValidator Token	Configuration Validator		@spartacus/core	sì	sì	core-libs/core/src/config/config-validator/config-validator.ts:L11-L13
39	CONSENT_TEMPLATE_NORMALIZER	ConsentTemplate Normalizer	Converter<any, Consent Template>	@spartacus/core	sì	sì	core-libs/core/src/user/connectors/consent/converters.ts:L11-L13
40	COST_CENTER_NORMALIZER	CostCenter Normalizer	Converter<any, CostCenter>	@spartacus/core	sì	sì	core-libs/core/src/cost-center/connectors/cost-center/converters.ts:L12-L14
41	COST_CENTER_SERIALIZER	CostCenter Serializer	Converter<CostCenter, any>	@spartacus/core	sì	sì	core-libs/core/src/cost-center/connectors/cost-center/converters.ts:L20-L22
42	COST_CENTERS_NORMALIZER	CostCentersList Normalizer	Converter<any, Entities Model<CostCenter>>	@spartacus/core	sì	sì	core-libs/core/src/cost-center/connectors/cost-center/converters.ts:L16-L18
43	COUNTRY_NORMALIZER	CountryNormalizer	Converter<any, Country>	@spartacus/core	sì	sì	core-libs/core/src/site-context/connectors/converters.ts:L20-L22
44	CURRENCY_NORMALIZER	CurrencyNormalizer	Converter<any, Currency>	@spartacus/core	sì	sì	core-libs/core/src/site-context/connectors/converters.ts:L16-L18

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
45	CUSTOMER_COUPON_SEARCH_RESULT_NORMALIZER	CustomerCoupon SearchResult Normalizer	Converter<any, CustomerCouponSearchResult>	@spartacus/core	sì	sì	core-libs/core/src/user/connectors/customer-coupon/converters.ts:L11-L13
46	DEFAULT_AUTH_HTTP_HEADER_SERVICE	DEFAULT_AUTH_HTTP_HEADER_SERVICE	AuthHttpHeaderService	@spartacus/core	sì	sì	core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L55-L71
47	DEFAULT_URL_MATCHER	DEFAULT_URL_MATCHER	UrlMatcherFactory	@spartacus/core	sì	sì	core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L35-L45
48	DefaultConfig	Default Configuration		@spartacus/core	sì	sì	core-libs/core/src/config/config-tokens.ts:L32-L35
49	DefaultConfigChunk	Default ConfigurationChunk	Config[]	@spartacus/core	sì	sì	core-libs/core/src/config/config-tokens.ts:L60-L62
50	DefaultFeature Toggles	DefaultFeature Toggles	FeatureToggles	@spartacus/core	sì	sì	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L33-L43
51	DefaultFeature TogglesChunk	DefaultFeature TogglesChunk	FeatureToggles[]	@spartacus/core	sì	sì	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L75-L78
52	DELEGATED_AUTH_HTTP_HEADER_SERVICE	DELEGATED_AUTH_HTTP_HEADER_SERVICE	AuthHttpHeaderService	@spartacus/core	sì	sì	core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L79-L82
53	EXPIRED_REFRESH_TOKEN_HANDLERS	EXPIRED_REFRESH_TOKEN_HANDLERS	ExpiredRefreshToken Handler[]	@spartacus/core	sì	sì	core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts:L21-L23
54	FeatureToggles Chunk	FeatureTogglesChunk	FeatureToggles[]	@spartacus/core	sì	sì	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L64-L67
55	I18NEXT_HTTP_BACKEND_CLIENT	I18NEXT_HTTP_BACKEND_CLIENT	HttpBackend Options['request']	@spartacus/core	sì	sì	core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L16-L44
56	I18NEXT_INSTANCE	I18NEXT_INSTANCE	i18n	@spartacus/core	sì	sì	core-libs/core/src/i18n/i18next/i18next-instance.ts:L19-L22
57	I18NEXT_LOGGER_PLUGIN	I18NEXT_LOGGER_PLUGIN	LoggerModule	@spartacus/core	sì	no	core-libs/core/src/i18n/i18next/i18next-plugins/i18next-logger-plugin.ts:L17-L40

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
58	LANGUAGE_NORMALIZER	LanguageNormalizer	Converter<any, Language>	@spartacus/core	sì	sì	core-libs/core/src/site-context/connectors/converters.ts:L12-L14
59	LOCATION_INITIALIZED_MULTI	MULTI_LOCATION_INITIALIZED	LocationInitializer[]	@spartacus/core	sì	sì	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L36-L38
60	MODULE_INITIALIZER	MODULE_INITIALIZER	(() => any)[]	@spartacus/core	sì	sì	core-libs/core/src/lazy-loading/tokens.ts:L19-L20
61	MULTI_ERROR_HANDLER	MULTI_ERROR_HANDLER	MultiErrorHandler[]	@spartacus/core	sì	sì	core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L25-L27
62	NOTIFICATION_PREFERENCE_NORMALIZER	Notification Preference Normalizer	Converter<any, NotificationPreference>	@spartacus/core	sì	no	core-libs/core/src/user/connectors/notification-preference/converters.ts:L15-L17
63	NOTIFICATION_PREFERENCE_SERIALIZER	Notification Preference Serializer	Converter<NotificationPreference[], any>	@spartacus/core	sì	no	core-libs/core/src/user/connectors/notification-preference/converters.ts:L11-L13
64	OCC_USER_ID_CONSTANTS	List of OCC constants that pass for user IDs.	{ [identifier: string]: string; }	@spartacus/core	sì	sì	core-libs/core/src/occ/utils/index.ts:L19-L24
65	PAYMENT_DETAILS_NORMALIZER	PaymentDetails Normalizer	Converter<any, PaymentDetails>	@spartacus/core	sì	sì	core-libs/core/src/checkout/connectors/payment/converters.ts:L11-L13
66	POINT_OF_SERVICE_NORMALIZER	PointOfService Normalizer	Converter<any, PointOfService>	@spartacus/core	sì	sì	core-libs/core/src/checkout/connectors/checkout/converters.ts:L11-L13
67	PRODUCT_INTERESTS_NORMALIZER	ProductInterests Normalizer	Converter<any, ProductInterestSearchResult>	@spartacus/core	sì	sì	core-libs/core/src/user/connectors/interests/converters.ts:L11-L13
68	PRODUCT_NORMALIZER	ProductNormalizer	Converter<any, Product>	@spartacus/core	sì	sì	core-libs/core/src/product/connectors/product/converters.ts:L11-L13
69	PRODUCT_REFERENCES_NORMALIZER	ProductReferences ListNormalizer	Converter<any, ProductReference[]>	@spartacus/core	sì	sì	core-libs/core/src/product/connectors/references/converters.ts:L11-L13
70	PRODUCT_REVIEW_NORMALIZER	ProductReview Normalizer	Converter<any, Review>	@spartacus/core	sì	sì	core-libs/core/src/product/connectors/reviews/converters.ts:L11-L13
71	PRODUCT_REVIEW_SERIALIZER	ProductReview Serializer	Converter<Review, any>	@spartacus/core	sì	sì	core-libs/core/src/product/connectors/reviews/converters.ts:L15-L17

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
72	PRODUCT_SEARCH_PAGE_NORMALIZER	ProductSearchPage Normalizer	Converter<any, ProductSearchPage>	@spartacus/core	sì	sì	core-libs/core/src/product/connectors/search/converters.ts:L14-L16
73	PRODUCT_SUGGESTION_NORMALIZER	ProductSuggestion Normalizer	Converter<any, Suggestion>	@spartacus/core	sì	sì	core-libs/core/src/product/connectors/search/converters.ts:L18-L20
74	reducerToken	AnonymousConsents Reducers	ActionReducerMap<AnonymousConsentsState>	@spartacus/core	sì	no	core-libs/core/src/anonymous-consents/store/reducers/index.ts:L38-L42
75	reducerToken	ClientAuthReducers	ActionReducerMap<ClientAuthState>	@spartacus/core	sì	no	core-libs/core/src/auth/client-auth/store/reducers/index.ts:L19-L20
76	reducerToken	CmsReducers	ActionReducerMap<CmsState>	@spartacus/core	sì	no	core-libs/core/src/cms/store/reducers/index.ts:L67-L68
77	reducerToken	GlobalMessage Reducers	ActionReducer<GlobalMessageState>	@spartacus/core	sì	no	core-libs/core/src/global-message/store/reducers/index.ts:L16-L19
78	reducerToken	ProcessReducers	ActionReducerMap<EntityLoaderState<any>>	@spartacus/core	sì	no	core-libs/core/src/process/store/reducers/index.ts:L17-L21
79	reducerToken	ProductReducers	ActionReducerMap<ProductsState>	@spartacus/core	sì	no	core-libs/core/src/product/store/reducers/index.ts:L37-L38
80	reducerToken	RouterReducers	ActionReducerMap<State>	@spartacus/core	sì	no	core-libs/core/src/routing/store/reducers/router.reducer.ts:L97-L98
81	reducerToken	SiteContextReducers	ActionReducerMap<SiteContextState>	@spartacus/core	sì	no	core-libs/core/src/site-context/store/reducers/index.ts:L22-L23
82	reducerToken	SiteThemeReducers	ActionReducerMap<SiteThemeState>	@spartacus/core	sì	no	core-libs/core/src/site-theme/store/reducers/index.ts:L19-L20
83	reducerToken	UserReducers	ActionReducerMap<UserState>	@spartacus/core	sì	no	core-libs/core/src/user/store/reducers/index.ts:L91-L92
84	REGION_NORMALIZER	RegionNormalizer	Converter<any, Region>	@spartacus/core	sì	sì	core-libs/core/src/site-context/connectors/converters.ts:L24-L26
85	RootConfig	RootConfiguration		@spartacus/core	sì	sì	core-libs/core/src/config/config-tokens.ts:L44-L47
86	RootFeatureToggles	RootFeatureToggles	FeatureToggles	@spartacus/core	sì	sì	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L48-L58

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
87	SERVER_REQUEST_ORIGIN	SERVER_REQUEST_ORIGIN	string	@spartacus/core	sì	sì	core-libs/core/src/util/ssr.tokens.ts:L19-L21
88	SERVER_REQUEST_URL	SERVER_REQUEST_URL	string	@spartacus/core	sì	sì	core-libs/core/src/util/ssr.tokens.ts:L12-L14
89	TEST_CONFIG	TEST_CONFIG	any	@spartacus/core	sì	sì	core-libs/core/src/config/test-config.module.ts:L42-L55
90	TEST_CONFIG_COOKIE_NAME	TEST_CONFIG_COOKIE_NAME	string	@spartacus/core	sì	sì	core-libs/core/src/config/test-config.module.ts:L31-L33
91	TRANSFER_STATE_META_REDUCER	TransferStateMetaReducer		@spartacus/core	sì	no	core-libs/core/src/state/reducers/index.ts:L23-L25
92	USE_AUTHORIZATION_CODE_FLOW_BY_DEFAULT	USE_AUTHORIZATION_CODE_FLOW_BY_DEFAULT	boolean	@spartacus/core	sì	sì	core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L15-L19
93	CUSTOMER_TICKETING_ASSOCIATED_OBJECTS_NORMALIZER	CustomerTicketingAssociatedObjectsNormalizer	Converter<any, AssociatedObject>	@spartacus/customer-ticketing	sì	sì	feature-libs/customer-ticketing/core/connectors/converters.ts:L38-L41
94	CUSTOMER_TICKETING_CATEGORY_NORMALIZER	CustomerTicketingCategoryNormalizer	Converter<any, Category>	@spartacus/customer-ticketing	sì	sì	feature-libs/customer-ticketing/core/connectors/converters.ts:L34-L36
95	CUSTOMER_TICKETING_CREATE_NORMALIZER	CustomerTicketingCreateNormalizer	Converter<any, TicketStarter>	@spartacus/customer-ticketing	sì	sì	feature-libs/customer-ticketing/core/connectors/converters.ts:L43-L45
96	CUSTOMER_TICKETING_DETAILS_NORMALIZER	CustomerTicketingDetailsNormalizer	Converter<any, TicketDetails>	@spartacus/customer-ticketing	sì	sì	feature-libs/customer-ticketing/core/connectors/converters.ts:L18-L20
97	CUSTOMER_TICKETING_EVENT_NORMALIZER	CustomerTicketingEventNormalizer	Converter<any, TicketEvent>	@spartacus/customer-ticketing	sì	sì	feature-libs/customer-ticketing/core/connectors/converters.ts:L22-L24
98	CUSTOMER_TICKETING_FILE_NORMALIZER	CustomerTicketingFileNormalizer	Converter<any, File>	@spartacus/customer-ticketing	sì	sì	feature-libs/customer-ticketing/core/connectors/converters.ts:L26-L28
99	CUSTOMER_TICKETING_LIST_NORMALIZER	CustomerTicketingListNormalizer	Converter<any, TicketList>	@spartacus/customer-ticketing	sì	sì	feature-libs/customer-ticketing/core/connectors/converters.ts:L30-L32
100	DP_DETAILS_NORMALIZER	DpDetailsNormalizer	Converter<any, PaymentDetails>	@spartacus/digital-payments	sì	no	integration-libs/digital-payments/src/checkout/adapters/converters.ts:L11-L13
101	DP_REQUEST_NORMALIZER	DpRequestNormalizer	Converter<any, DpPaymentRequest>	@spartacus/digital-payments	sì	no	integration-libs/digital-payments/src/checkout/adapters/converters.ts:L15-L17

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
102	LOOKUP_VISUALIZATIONS_RESPONSE_NORMALIZER	Lookup Visualizations ResponseNormalizer	Converter<any, LookupVisualizationsResponse>	@spartacus/epd-visualization	sì	sì	integration-libs/epd-visualization/core/connectors/visualization/converters.ts:L14-L16
103	NODES_RESPONSE_NORMALIZER	NodesResponse Normalizer	Converter<any, NodesResponse>	@spartacus/epd-visualization	sì	sì	integration-libs/epd-visualization/core/connectors/scene/converters.ts:L14-L16
104	OPF_ACTIVE_CONFIGURATIONS_NORMALIZER	OpfActive Configurations Normalizer	Converter<any, OpfActiveConfigurationsResponse>	@spartacus/opf	sì	sì	integration-libs/opf/base/core/tokens/tokens.ts:L11-L13
105	OPF_AFTER_REDIRECT_SCRIPTS_NORMALIZER	OpfAfterRedirect ScriptsNormalizer	Converter<any, OpfPaymentAfterRedirectScriptResponse>	@spartacus/opf	sì	sì	integration-libs/opf/payment/core/tokens/tokens.ts:L28-L30
106	OPF_APPLE_PAY_WEB_SESSION_NORMALIZER	OpfApplePayWeb Session	Converter<any, ApplePaySessionVerificationResponse>	@spartacus/opf	sì	sì	integration-libs/opf/quick-buy/core/tokens/tokens.ts:L11-L13
107	OPF_CTA_SCRIPTS_NORMALIZER	OpfCtaScripts Normalizer	Converter<any, OpfCtaScriptsResponse>	@spartacus/opf	sì	sì	integration-libs/opf/cta/core/tokens/tokens.ts:L11-L13
108	OPF_GET_CART_USER_EMAIL_NORMALIZER	OpfGetCartUserEmail Normalizer	Converter<any, CartUserEmailResponse>	@spartacus/opf	sì	no	integration-libs/opf/checkout/root/tokens/tokens.ts:L11-L13
109	OPF_PAYMENT_CONFIG_SERIALIZER	OpfPaymentConfig Serializer	Converter<OpfPaymentInitiationConfig, any>	@spartacus/opf	sì	sì	integration-libs/opf/payment/core/connectors/converters.ts:L11-L13
110	OPF_PAYMENT_SUBMIT_COMPLETE_NORMALIZER	OpfPaymentSubmit CompleteNormalizer	Converter<any, OpfPaymentSubmitCompleteResponse>	@spartacus/opf	sì	sì	integration-libs/opf/payment/core/tokens/tokens.ts:L24-L26
111	OPF_PAYMENT_SUBMIT_NORMALIZER	OpfPaymentSubmit Normalizer	Converter<any, OpfPaymentSubmitResponse>	@spartacus/opf	sì	sì	integration-libs/opf/payment/core/tokens/tokens.ts:L20-L22
112	OPF_PAYMENT_VERIFICATION_NORMALIZER	OpfPayment Verification Normalizer	Converter<any, OpfPaymentVerificationResponse>	@spartacus/opf	sì	sì	integration-libs/opf/payment/core/tokens/tokens.ts:L16-L18
113	CONSIGNMENT_TRACKING_NORMALIZER	ConsignmentTracking Normalizer	Converter<any, ConsignmentTracking>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L40-L42
114	ORDER_HISTORY_NORMALIZER	OrderHistory Normalizer	Converter<any, OrderHistoryList>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L36-L38
115	ORDER_NORMALIZER	OrderNormalizer	Converter<any, Order>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L24-L26
116	ORDER_RETURN_REQUEST_INPUT_SERIALIZER	OrderReturnRequest InputSerializer	Converter<ReturnRequestEntryInputList, any>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L48-L50
117	ORDER_RETURN_REQUEST_NORMALIZER	OrderReturnRequest Normalizer	Converter<any, ReturnRequest>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L44-L46
118	ORDER_RETURNS_NORMALIZER	OrderReturns Normalizer	Converter<any, ReturnRequestList>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L52-L54

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
119	OrderConfirmation OrderEntries ContextToken	OrderConfirmation OrderEntriesContext		@spartacus/order	sì	sì	feature-libs/order/root/tokens/context.ts:L13-L15
120	OrderDetailsOrder EntriesContext Token	OrderDetailsOrder EntriesContext		@spartacus/order	sì	sì	feature-libs/order/root/tokens/context.ts:L9-L11
121	reducerToken	OrderReducers	ActionReducerMap<Order State>	@spartacus/order	sì	no	feature-libs/order/core/store/reducers/index.ts:L79-L80
122	REORDER_ORDER_ NORMALIZER	ReorderOrder Normalizer	Converter<any, Cart ModificationList>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L32-L34
123	REPLENISHMENT_ ORDER_FORM_ SERIALIZER	ReplenishmentOrder FormSerializer	Converter<Schedule ReplenishmentForm, any>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L60-L62
124	REPLENISHMENT_ ORDER_HISTORY_ NORMALIZER	ReplenishmentOrder HistoryNormalizer	Converter<any, ReplenishmentOrderList>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L56-L58
125	REPLENISHMENT_ ORDER_NORMALIZER	ReplenishmentOrder Normalizer	Converter<any, ReplenishmentOrder>	@spartacus/order	sì	sì	feature-libs/order/root/tokens/converters.ts:L28-L30
126	USE_MY_ACCOUNT_V2_ ORDER	feature flag to enable enhanced UI for Order related pages under My-Account	boolean	@spartacus/order	sì	sì	feature-libs/order/root/tokens/context.ts:L17-L20
127	ACCOUNT_SUMMARY_ DOCUMENT_ NORMALIZER	AccountSummary DocumentNormalizer	Converter<any, Account SummaryList>	@spartacus/organization	sì	sì	feature-libs/organization/account-summary/core/connectors/converters.ts:L18-L20
128	ACCOUNT_SUMMARY_ NORMALIZER	AccountSummary Normalizer	Converter<any, Account SummaryDetails>	@spartacus/organization	sì	sì	feature-libs/organization/account-summary/core/connectors/converters.ts:L14-L16
129	B2B_USER_ NORMALIZER	B2BUserNormalizer	Converter<any, B2BUser>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/b2b-user/converters.ts:L10-L12
130	B2B_USER_ SERIALIZER	B2BUserSerializer	Converter<B2BUser, any>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/b2b-user/converters.ts:L14-L16
131	B2B_USERS_ NORMALIZER	UserListNormalizer	Converter<any, Entities Model<B2BUser>>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/b2b-user/converters.ts:L18-L20
132	B2BUNIT_APPROVAL_ PROCESSES_ NORMALIZER	B2BUnitApproval ProcessNormalizer	Converter<any, B2BApproval Process[]>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/org-unit/converters.ts:L27-L29

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
133	B2BUNIT_NODE_LIST_NORMALIZER	B2BUnitNodeList Normalizer	Converter<any, B2BUnit Node[]>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/org-unit/converters.ts:L15-L17
134	B2BUNIT_NODE_NORMALIZER	B2BUnitNode Normalizer	Converter<any, B2BUnit Node>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/org-unit/converters.ts:L11-L13
135	B2BUNIT_NORMALIZER	B2BUnitNormalizer	Converter<any, B2BUnit>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/org-unit/converters.ts:L19-L21
136	B2BUNIT_SERIALIZER	B2BUnitSerializer	Converter<B2BUnit, any>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/org-unit/converters.ts:L23-L25
137	BUDGET_NORMALIZER	BudgetNormalizer	Converter<any, Budget>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/budget/converters.ts:L11-L13
138	BUDGET_SERIALIZER	BudgetSerializer	Converter<Budget, any>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/budget/converters.ts:L18-L20
139	BUDGETS_NORMALIZER	BudgetsList Normalizer	Converter<any, EntitiesModel<Budget>>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/budget/converters.ts:L14-L16
140	ORDER_APPROVAL_DECISION_NORMALIZER	OrderApproval DecisionNormalizer	Converter<any, OrderApprovalDecision>	@spartacus/organization	sì	sì	feature-libs/organization/order-approval/core/connectors/converters.ts:L22-L24
141	ORDER_APPROVAL_NORMALIZER	OrderApproval Normalizer	Converter<any, OrderApproval>	@spartacus/organization	sì	sì	feature-libs/organization/order-approval/core/connectors/converters.ts:L14-L16
142	ORDER_APPROVALS_NORMALIZER	OrderApprovalsList Normalizer	Converter<any, EntitiesModel<OrderApproval>>	@spartacus/organization	sì	sì	feature-libs/organization/order-approval/core/connectors/converters.ts:L18-L20
143	ORGANIZATION_USER_REGISTRATION_SERIALIZER	OrganizationUser Registration Serializer	Converter<OrganizationUserRegistration, any>	@spartacus/organization	sì	sì	feature-libs/organization/user-registration/core/connectors/converters.ts:L11-L13
144	PERMISSION_NORMALIZER	Permission Normalizer	Converter<any, Permission>	@spartacus/organization	sì	sì	feature-libs/organization/administration/core/connectors/permission/converters.ts:L15-L17

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
145	PERMISSION_ SERIALIZER	Permission Serializer	Converter<Permission, any>	@spartacus/organization	sì	sì	feature-libs/ organization/ administration/core/ connectors/permission/ converters.ts:L31-L33
146	PERMISSION_TYPE_ NORMALIZER	PermissionType Normalizer	Converter<any, Order ApprovalPermissionType>	@spartacus/organization	sì	sì	feature-libs/ organization/ administration/core/ connectors/permission/ converters.ts:L23-L25
147	PERMISSION_TYPES_ NORMALIZER	PermissionTypesList Normalizer	Converter<any, Order ApprovalPermissionType[]>	@spartacus/organization	sì	sì	feature-libs/ organization/ administration/core/ connectors/permission/ converters.ts:L27-L29
148	PERMISSIONS_ NORMALIZER	PermissionsList Normalizer	Converter<any, Entities Model<Permission>>	@spartacus/organization	sì	sì	feature-libs/ organization/ administration/core/ connectors/permission/ converters.ts:L19-L21
149	reducerToken	Organization Reducers	ActionReducerMap< OrganizationState>	@spartacus/organization	sì	no	feature-libs/ organization/ administration/core/ store/reducers/index. ts:L199-L202
150	reducerToken	Organization Reducers	ActionReducerMap<Order ApprovalState>	@spartacus/organization	sì	no	feature-libs/ organization/order-approval/core/store/ reducers/index.ts:L43-L47
151	reducerToken	UnitOrderReducers	ActionReducerMap<UnitOrder State>	@spartacus/organization	sì	no	feature-libs/ organization/unit-order/core/store/ reducers/index.ts:L32-L33
152	UnitOrderDetails OrderEntries ContextToken	OrderDetailsOrder EntriesContext		@spartacus/organization	sì	sì	feature-libs/ organization/unit-order/root/context/ context.ts:L9-L11
153	USER_GROUP_ NORMALIZER	UserGroupNormalizer	Converter<any, UserGroup>	@spartacus/organization	sì	sì	feature-libs/ organization/ administration/core/ connectors/user-group/ converters.ts:L11-L13
154	USER_GROUP_ SERIALIZER	UserGroupSerializer	Converter<UserGroup, any>	@spartacus/organization	sì	sì	feature-libs/ organization/ administration/core/ connectors/user-group/ converters.ts:L15-L17
155	USER_GROUPS_ NORMALIZER	UserGroupList Normalizer	Converter<any, Entities Model<UserGroup>>	@spartacus/organization	sì	sì	feature-libs/ organization/ administration/core/ connectors/user-group/ converters.ts:L19-L21
156	PDF_INVOICES_LIST_ INVOICES_ NORMALIZER	PDFInvoicesList Invoices	Converter<any, Order InvoiceList>	@spartacus/pdf-invoices	sì	sì	feature-libs/pdf-invoices/core/ connectors/converters. ts:L11-L13

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
157	pickupLocations ReducersToken	PickupLocations Reducers	ActionReducerMap<PickupLocationsState>	@spartacus/pickup-in-store	sì	sì	feature-libs/pickup-in-store/core/store/reducers/pickup-locations/index.ts:L23-L27
158	pickupOption ReducersToken	PickupOption Reducers	ActionReducerMap<PickupOptionState>	@spartacus/pickup-in-store	sì	sì	feature-libs/pickup-in-store/core/store/reducers/pickup-option/index.ts:L21-L25
159	stockReducersToken	StockReducers	ActionReducerMap<StockState>	@spartacus/pickup-in-store	sì	sì	feature-libs/pickup-in-store/core/store/reducers/stock/index.ts:L34-L35
160	FUTURE_STOCK_LIST_NORMALIZER	FutureStockList Normalizer	Converter<any, Product FutureStockList>	@spartacus/product	sì	sì	feature-libs/product/future-stock/core/connectors/converters.ts:L18-L20
161	FUTURE_STOCK_NORMALIZER	FutureStock Normalizer	Converter<any, Product FutureStock>	@spartacus/product	sì	sì	feature-libs/product/future-stock/core/connectors/converters.ts:L14-L16
162	(inline) Rulebased Configurator Connector.CONFIGURATOR_ADAPTER_LIST	ConfiguratorAdapterList	RulebasedConfiguratorAdapter[]	@spartacus/product-configurator	no	n/d	feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L21-L23
163	CONFIGURATION_TEXTFIELD_ADD_TO_CART_SERIALIZER	ConfigurationAddToCartSerializer	Converter<ConfiguratorTextfield.AddToCartParameters, any>	@spartacus/product-configurator	sì	no	feature-libs/product-configurator/textfield/core/connectors/converters.ts:L15-L18
164	CONFIGURATION_TEXTFIELD_NORMALIZER	Configuration Normalizer	Converter<any, ConfiguratorTextfield.Configuration>	@spartacus/product-configurator	sì	no	feature-libs/product-configurator/textfield/core/connectors/converters.ts:L11-L13
165	CONFIGURATION_TEXTFIELD_UPDATE_CART_ENTRY_SERIALIZER	ConfigurationUpdateCartEntrySerializer	Converter<ConfiguratorTextfield.UpdateCartEntryParameters, any>	@spartacus/product-configurator	sì	no	feature-libs/product-configurator/textfield/core/connectors/converters.ts:L20-L23
166	configurator ReducerToken	Configurator Reducers	ActionReducerMap<ConfiguratorState>	@spartacus/product-configurator	sì	no	feature-libs/product-configurator/rulebased/core/state/reducers/index.ts:L26-L30
167	configurator TextfieldReducerToken	Configurator Reducers	ActionReducerMap<ConfigurationTextfieldState>	@spartacus/product-configurator	sì	no	feature-libs/product-configurator/textfield/core/state/reducers/index.ts:L27-L31
168	CPQ_CONFIGURATOR_ADD_TO_CART_SERIALIZER	CpqConfiguratorAddToCartSerializer	Converter< Configurator.AddToCartParameters, Occ CpqConfigurator.AddToCartParameters >	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/cpq/occ/converters/cpq-configurator-occ.converters.ts:L12-L17
169	CPQ_CONFIGURATOR_NORMALIZER	CpqConfigurator Normalizer	Converter<Cpq.Configuration, Configurator.Configuration>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator.converters.ts:L12-L14

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
170	CPQ_CONFIGURATOR_OVERVIEW_NORMALIZER	CpqConfigurator OverviewNormalizer	Converter<Cpq.Configuration, Configurator.Overview>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator.converters.ts:L24-L26
171	CPQ_CONFIGURATOR_QUANTITY_SERIALIZER	CpqConfigurator ValueSerializer	Converter<Configurator.Configuration, Cpq.UpdateValue>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator.converters.ts:L20-L22
172	CPQ_CONFIGURATOR_SERIALIZER	CpqConfigurator Serializer	Converter<Configurator.Configuration, Cpq.UpdateAttribute>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator.converters.ts:L16-L18
173	CPQ_CONFIGURATOR_UPDATE_CART_ENTRY_SERIALIZER	CpqConfigurator UpdateCartEntry Serializer	Converter< Configurator.AddToCartParameters, Occ CpqConfigurator.UpdateConfigurationForCartEntryParameters >	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/cpq/occ/converters/cpq-configurator-occ.converters.ts:L19-L24
174	VARIANT_CONFIGURATOR_ADD_TO_CART_SERIALIZER	VariantConfigurator AddToCartSerializer	Converter< Configurator.AddToCartParameters, Occ Configurator.AddToCartParameters >	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L28-L33
175	VARIANT_CONFIGURATOR_NORMALIZER	VariantConfigurator Normalizer	Converter<OccConfigurator.Configuration, Configurator.Configuration>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L12-L14
176	VARIANT_CONFIGURATOR_OVERVIEW_NORMALIZER	VariantConfigurator OverviewNormalizer	Converter<OccConfigurator.Overview, Configurator.Overview>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L43-L45
177	VARIANT_CONFIGURATOR_OVERVIEW_SERIALIZER	VariantConfigurator OverviewSerializer	Converter<Configurator.Overview, OccConfigurator.Overview>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L47-L49
178	VARIANT_CONFIGURATOR_PRICE_NORMALIZER	VariantConfigurator PriceNormalizer	Converter<OccConfigurator.PriceSummary, Configurator.Configuration>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L24-L26
179	VARIANT_CONFIGURATOR_PRICE_SUMMARY_NORMALIZER	VariantConfigurator PriceSummary Normalizer	Converter<OccConfigurator.PriceSummary, Configurator.PriceSummary>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L20-L22
180	VARIANT_CONFIGURATOR_SERIALIZER	VariantConfigurator Serializer	Converter<Configurator.Configuration, Occ Configurator.Configuration>	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L16-L18
181	VARIANT_CONFIGURATOR_UPDATE_CART_ENTRY_SERIALIZER	VariantConfigurator UpdateCartEntry Serializer	Converter< Configurator.UpdateConfigurationForCartEntryParameters, Occ Configurator.UpdateConfigurationForCartEntryParameters >	@spartacus/product-configurator	sì	sì	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L35-L41

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
182	PUNCHOUT_REQUISITION_NORMALIZER	PunchoutRequisitionNormalizer	Converter<any, PunchoutRequisition>	@spartacus/punchout	sì	sì	integration-libs/punchout/core/connectors/converters.ts:L15-L17
183	PUNCHOUT_SESSION_NORMALIZER	PunchoutSessionNormalizer	Converter<any, PunchoutSession>	@spartacus/punchout	sì	sì	integration-libs/punchout/core/connectors/converters.ts:L11-L13
184	QUOTE_ACTION_SERIALIZER	QuoteActionSerializer	Converter<any, QuoteActionType>	@spartacus/quote	sì	sì	feature-libs/quote/core/connectors/converters.ts:L36-L38
185	QUOTE_COMMENT_SERIALIZER	QuoteCommentSerializer	Converter<any, QuoteComment>	@spartacus/quote	sì	sì	feature-libs/quote/core/connectors/converters.ts:L40-L42
186	QUOTE_DISCOUNT_SERIALIZER	QuoteDiscountSerializer	Converter<any, QuoteDiscount>	@spartacus/quote	sì	sì	feature-libs/quote/core/connectors/converters.ts:L44-L46
187	QUOTE_LIST_NORMALIZER	QuoteListNormalizer	Converter<any, QuoteList>	@spartacus/quote	sì	sì	feature-libs/quote/core/connectors/converters.ts:L24-L26
188	QUOTE_METADATA_SERIALIZER	QuoteMetadataSerializer	Converter<any, QuoteMetadata>	@spartacus/quote	sì	sì	feature-libs/quote/core/connectors/converters.ts:L32-L34
189	QUOTE_NORMALIZER	QuoteNormalizer	Converter<OccQuote, Quote>	@spartacus/quote	sì	sì	feature-libs/quote/core/connectors/converters.ts:L20-L22
190	QUOTE_STARTER_SERIALIZER	QuoteStarterSerializer	Converter<any, QuoteStarter>	@spartacus/quote	sì	sì	feature-libs/quote/core/connectors/converters.ts:L28-L30
191	EXPRESS_SERVER_LOGGER	EXPRESS_SERVER_LOGGER	ExpressServerLogger	@spartacus/setup	sì	sì	core-libs/setup/ssr/logger/loggers/express-server-logger.ts:L46-L48
192	PROPAGATE_ERROR_TO_SERVER	PROPAGATE_ERROR_TO_SERVER	(error: unknown) => void	@spartacus/setup	sì	sì	core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L20-L22
193	REQUEST	REQUEST	Request	@spartacus/setup	sì	sì	core-libs/setup/ssr/tokens/express.tokens.ts:L15-L17
194	RESPONSE	RESPONSE	Response	@spartacus/setup	sì	sì	core-libs/setup/ssr/tokens/express.tokens.ts:L24-L26
195	reducerToken	StoreFinderReducers	ActionReducerMap<StoresState>	@spartacus/storefinder	sì	no	feature-libs/storefinder/core/store/reducers/index.ts:L31-L32
196	STORE_COUNT_NORMALIZER	StoreCountNormalizer	Converter<any, StoreCount>	@spartacus/storefinder	sì	sì	feature-libs/storefinder/core/connectors/converters.ts:L15-L17
197	STORE_FINDER_SEARCH_PAGE_NORMALIZER	StoreFinderSearchPageNormalizer	Converter<any, StoreFinderSearchPage>	@spartacus/storefinder	sì	sì	feature-libs/storefinder/core/connectors/converters.ts:L11-L13

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
198	BEFORE_CMS_PAGE_GUARD	BEFORE_CMS_PAGE_GUARD	{ canActivate: CanActivateFn }[]	@spartacus/storefront	sì	sì	core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L22-L24
199	JSONLD_PRODUCT_BUILDER	JsonLdProductBuilderToken		@spartacus/storefront	sì	sì	core-libs/storefront/cms-structure/seo/structured-data/builders/tokens.ts:L21-L23
200	LCP_PRESENCE	LCP_PRESENCE	Observable<LcpPresence>	@spartacus/storefront	sì	sì	core-libs/storefront/shared/lcp-context/lcp-presence.token.ts:L26-L32
201	PAGE_LAYOUT_HANDLER	PageLayoutHandler	PageLayoutHandler	@spartacus/storefront	sì	sì	core-libs/storefront/cms-structure/page/page-layout/page-layout-handler.ts:L11-L13
202	PRODUCT_DETAILS_URL_MATCHER	PRODUCT_DETAILS_URL_MATCHER	UrlMatcherFactory	@spartacus/storefront	sì	sì	core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L42-L50
203	PRODUCT_LISTING_URL_MATCHER	PRODUCT_LISTING_URL_MATCHER	UrlMatcherFactory	@spartacus/storefront	sì	sì	core-libs/storefront/cms-pages/product-listing-page/product-listing-url-matcher.ts:L42-L50
204	PROVIDE_OUTLET_OPTIONS	PROVIDE_OUTLET_OPTIONS	ProvideOutletOptions	@spartacus/storefront	sì	no	core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L13-L15
205	SCHEMA_BUILDER	SchemaBuilderToken		@spartacus/storefront	sì	sì	core-libs/storefront/cms-structure/seo/structured-data/builders/tokens.ts:L15-L15
206	USE_MY_ACCOUNT_V2_CONSENT	feature flag to enable enhanced UI for Consent Management pages under My-Account	boolean	@spartacus/storefront	sì	sì	core-libs/storefront/cms-components/myaccount/my-account-v2/use-my-account-v2-consent-notification-perference.ts:L15-L18
207	USE_MY_ACCOUNT_V2_NOTIFICATION_PREFERENCE	feature flag to enable enhanced UI for notification preference pages under My-Account	boolean	@spartacus/storefront	sì	sì	core-libs/storefront/cms-components/myaccount/my-account-v2/use-my-account-v2-consent-notification-perference.ts:L9-L13
208	LOGIN_FORM_SERIALIZER	LoginFormSerializer	Converter<VerificationTokenCreation, any>	@spartacus/user	sì	sì	feature-libs/user/account/core/connectors/converters.ts:L27-L29
209	TITLE_NORMALIZER	TitleNormalizer	Converter<any, Title>	@spartacus/user	sì	sì	feature-libs/user/profile/core/connectors/converters.ts:L28-L30

#	NOME CONST	DESCRIZIONE (STRINGA)	TIPO GENERICO	LIBRERIA	EXPORT	PUBLIC API	RIFERIMENTO
210	USE_MY_ACCOUNT_V2_EMAIL	feature flag to enable enhanced UI for email related pages under My-Account	boolean	@spartacus/user	sì	sì	feature-libs/user/profile/components/update-email/use-my-account-v2-email.ts:L9-L12
211	USE_MY_ACCOUNT_V2_PASSWORD	feature flag to enable enhanced UI for Update Password related pages under My-Account	boolean	@spartacus/user	sì	sì	feature-libs/user/profile/components/update-password/use-my-account-v2-password.ts:L9-L12
212	USE_MY_ACCOUNT_V2_PROFILE	feature flag to enable enhanced UI for profile related pages under My-Account	boolean	@spartacus/user	sì	sì	feature-libs/user/profile/components/update-profile/use-my-account-v2-profile.ts:L9-L12
213	USER_ACCOUNT_NORMALIZER	UserAccount Normalizer	Converter<any, User>	@spartacus/user	sì	sì	feature-libs/user/account/core/connectors/converters.ts:L15-L17
214	USER_ACCOUNT_SERIALIZER	UserAccount Serializer	Converter<User, any>	@spartacus/user	sì	sì	feature-libs/user/account/core/connectors/converters.ts:L19-L21
215	USER_PROFILE_NORMALIZER	UserProfile Normalizer	Converter<User, any>	@spartacus/user	sì	sì	feature-libs/user/profile/core/connectors/converters.ts:L12-L14
216	USER_PROFILE_SERIALIZER	UserProfile Serializer	Converter<User, any>	@spartacus/user	sì	sì	feature-libs/user/profile/core/connectors/converters.ts:L16-L18
217	USER_SERIALIZER	UserSerializer	Converter<User, any>	@spartacus/user	sì	sì	feature-libs/user/profile/core/connectors/converters.ts:L20-L22
218	USER_SIGN_UP_SERIALIZER	UserSignUp Serializer	Converter<UserSignUp, any>	@spartacus/user	sì	sì	feature-libs/user/profile/core/connectors/converters.ts:L24-L26
219	VERIFICATION_TOKEN_NORMALIZER	VerificationToken Normalizer	Converter<any, VerificationToken>	@spartacus/user	sì	sì	feature-libs/user/account/core/connectors/converters.ts:L23-L25

A02

Configurazione (*Config)

Interfacce e chiavi di configurazione

IN QUESTO CAPITOLO

Riepilogo per libreria
@spartacus/asm
@spartacus/assets
@spartacus/cart
@spartacus/cdc
@spartacus/cds
@spartacus/checkout
@spartacus/core
@spartacus/customer-ticketing
@spartacus/digital-payments
@spartacus/epd-visualization
@spartacus/estimated-delivery-date
@spartacus/omf
@spartacus/opf

Generata da `tools/extract/a02-config.js` allo SHA 5b7a1c2b (`5b7a1c2bf2808d7380bc20596d10f4fa710c3da1`). Non modificare a mano.
Righe di tabella: **519**.
Rigenerazione: `cd docs/spartacus-book/tools/extract && npm install && node a02-config.js`
Per ogni libreria: (1) tabella chiavi di ogni classe/interfaccia il cui nome termina in `Config` (proprietà dichiarate direttamente, non ereditate; tipo dal codice; opzionalità = `?`); (2) tabella delle `const default*Config` con valore letterale troncato a 120 caratteri. Il default è associato a una config quando la radice coincide (es. `CartConfig` ↔ `defaultCartConfig`, confronto case-insensitive su nome senza `default`/`Config`). *public API*: dichiarazione raggiungibile dall'`entryFile` di un `ng-package.json`. Le interfacce `Config` dentro `declare module` sono module augmentation.

Riepilogo per libreria

LIBRERIA	CLASSI/INTERFACCE *CONFIG	CHIAVI (RIGHE)	DEFAULT*CONFIG
@spartacus/asm	7	10	12
@spartacus/assets	1	1	0
@spartacus/cart	8	13	9
@spartacus/cdc	2	2	2
@spartacus/cds	5	16	0
@spartacus/checkout	2	2	7
@spartacus/core	58	113	14
@spartacus/customer-ticketing	3	5	4
@spartacus/digital-payments	2	2	2
@spartacus/epd-visualization	7	11	0
@spartacus/estimated-delivery-date	0	0	1
@spartacus/omf	2	2	1
@spartacus/opf	12	26	22
@spartacus/opps	2	2	1
@spartacus/order	2	3	9
@spartacus/organization	2	2	14
@spartacus/pdf-invoices	0	0	1
@spartacus/pickup-in-store	2	2	4
@spartacus/product	0	0	4
@spartacus/product-configurator	13	19	12
@spartacus/product-multi-dimensional	2	2	2
@spartacus/punchout	2	2	4
@spartacus/qualtrics	2	2	1
@spartacus/quote	7	8	7
@spartacus/requested-delivery-date	0	0	1

LIBRERIA	CLASSI/INTERFACCE *CONFIG	CHIAVI (RIGHE)	DEFAULT*CONFIG
@spartacus/s4-service	3	4	5
@spartacus/segment-refs	2	2	1
@spartacus/setup	0	0	1
@spartacus/smartedit	2	2	1
@spartacus/storefinder	2	2	4
@spartacus/storefront	44	72	17
@spartacus/subscription-billing	2	5	2
@spartacus/tracking	7	10	3
@spartacus/user	2	2	7
Totale	**207**	**344**	**175**

@spartacus/asm

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
AsmConfig	abstract class; @Injectable; public API	asm	{ agentSessionTimer?: { startingDelayIn Seconds?: number; }; customerSearch?: { max Results?: number; }; customerList?: { page Size?: number; show Avatar?: boolean…	sì	defaultAsmConfig = { asm: { agentSessionTimer: { starting DelayInSeconds: 600, }, customerSearch: { maxResults: 20, }, asmSessionSupport: {… (feature-libs/asm/root/config/default-asm-config.ts:L11-L84)	feature-libs/asm/root/config/asm-config.ts:L11-L55
Asm Customer360 Config	abstract class; @Injectable; public API	asm Customer360	AsmCustomer360Tabs Config	sì	defaultAsmCustomer360Config = { asmCustomer360: { date Format: 'MM-dd-yyyy', date TimeFormat: 'dd-MM-yy hh:mm a', tabs: [{ i18nNameKey: 'asmCustomer36… (feature-libs/asm/customer-360/components/config/default-asm-customer-360-config.ts:L21-L142)	feature-libs/asm/customer-360/components/config/asm-customer-360-config.ts:L11-L17
Asm Customer360 Section Config	abstract class; public API	pageSize	number	sì		feature-libs/asm/customer-360/root/model/asm-customer-360-section-config.ts:L7-L9
Asm Customer360 TabConfig	abstract class; public API	components	Array<AsmCustomer360 TabComponent>	no		feature-libs/asm/customer-360/root/model/asm-customer-360-tab-config.ts:L18-L21
〃		i18nNameKey	string	no		

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE DEFAULT (STESSA RADICE)	RIFERIMENTO
AsmCustomer360TabsConfig	abstract class; public API	tabs	Array<AsmCustomer360TabConfig>	sì	feature-libs/asm/customer-360/root/model/asm-customer-360-tabs-config.ts:L9-L13
〃		dateFormat	string	sì	
〃		dateTimeFormat	string	sì	
Config	interface; extends AsmCustomer360Config; declare module '@spartacus/core'	(nessuna chiave propria)			feature-libs/asm/customer-360/components/config/asm-customer-360-config.ts:L20-L20
Config	interface; extends AsmConfig; declare module '@spartacus/core'	(nessuna chiave propria)			feature-libs/asm/root/config/asm-config.ts:L58-L58

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultAsmConfig	AsmConfig	{ asm: { agentSessionTimer: { startingDelayInSeconds: 600, }, customerSearch: { maxResults: 20, }, asmSessionSupport: {…	no	feature-libs/asm/root/config/default-asm-config.ts:L11-L84
defaultAsmCreateCustomerFormLayoutConfig	LayoutConfig	{ launch: { ASM_CREATE_CUSTOMER_FORM: { inlineRoot: true, component: AsmCreateCustomerFormComponent, dialogType: DIALOG…	no	feature-libs/asm/components/asm-create-customer-form/default-asm-create-customer-form-layout.config.ts:L10-L18
defaultAsmCustomer360Config	AsmCustomer360Config	{ asmCustomer360: { dateFormat: 'MM-dd-yyyy', dateTimeFormat: 'dd-MM-yy hh:mm a', tabs: [{ i18nNameKey: 'asmCustomer36…	sì	feature-libs/asm/customer-360/components/config/default-asm-customer-360-config.ts:L21-L142
defaultAsmCustomer360LayoutConfig	LayoutConfig	{ launch: { [LAUNCH_CALLER.ASM_CUSTOMER_360]: { inlineRoot: true, component: AsmCustomer360Component, dialogType: DIALO…	sì	feature-libs/asm/customer-360/components/default-asm-customer-360-layout.config.ts:L14-L22
defaultAsmLayoutConfig	LayoutConfig	{ launch: { ASM: { outlet: 'cx-storefront', component: AsmMainUiComponent, }, }, }	no	feature-libs/asm/components/default-asm-layout.config.ts:L10-L17

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultAsmPaginationConfig	PaginationConfig	{ pagination: { rangeCount: 2, addPrevious: true, addNext: true, addStart: false, addEnd: false, addFirst: true, addLas…	no	feature-libs/asm/components/default-asm-pagination.config.ts:L9-L22
defaultBindCartLayoutConfig	LayoutConfig	{ launch: { [LAUNCH_CALLER.ASM_BIND_CART]: { inlineRoot: true, component: AsmBindCartDialogComponent, dialogType: DIALO…	no	feature-libs/asm/components/default-bind-cart-layout.config.ts:L14-L22
defaultCustomerListLayoutConfig	LayoutConfig	{ launch: { ASM_CUSTOMER_LIST: { inlineRoot: true, component: CustomerListComponent, dialogType: DIALOG_TYPE.DIALOG, },…	no	feature-libs/asm/components/customer-list/default-customer-list-layout.config.ts:L10-L18
defaultOccAsmConfig	OccConfig	{ backend: { occ: { endpoints: { asmCustomerSearch: '/assistedservicewebservices/customers/search', asmCustomerLists: '…	no	feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L9-L22
defaultOccAsmCustomer360Config	OccConfig	{ backend: { occ: { endpoints: { asmCustomer360: '/assistedservicewebservices/${baseSiteId}/users/${userId}/customer360…	no	feature-libs/asm/customer-360/occ/adapters/default-occ-asm-customer-360-config.ts:L9-L18
defaultSaveCartLayoutConfig	LayoutConfig	{ launch: { [LAUNCH_CALLER.ASM_SAVE_CART]: { inlineRoot: true, component: AsmSaveCartDialogComponent, dialogType: DIALO…	no	feature-libs/asm/components/default-save-cart-layout.config.ts:L14-L22
defaultSwitchCustomerLayoutConfig	LayoutConfig	{ launch: { [LAUNCH_CALLER.ASM_SWITCH_CUSTOMER]: { inlineRoot: true, component: AsmSwitchCustomerDialogComponent, dialo…	no	feature-libs/asm/components/default-switch-customer-layout.config.ts:L14-L22

@spartacus/assets

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
TranslationChunksConfig	interface; public API	[chunk: string]	string[]	-		core-libs/assets/src/translations/translation-chunks-config.ts:L7-L9

@spartacus/cart

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
CartConfig	abstract class; @Injectable; public API	cart	{ selectiveCart?: { enabled?: boolean; }; validation?: { enabled?: boolean; }; showRealTimeStockInPDP?: { enabled?: boolean; }; }	sì	defaultCartConfig = { cart: { validation: { enabled: false, }, selectiveCart: { enabled: false, }, showRealTimeStockInPDP: { enabled: false… (feature-libs/cart/base/root/config/default-cart-config.ts:L9-L21)	feature-libs/cart/base/root/config/cart-config.ts:L12-L28
Config	interface; extends CartConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/cart/base/root/config/cart-config.ts:L31-L31
Config	interface; extends ImportExportConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/cart/import-export/core/config/import-export-config.ts:L25-L25
Config	interface; extends QuickOrderConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/cart/quick-order/root/config/quick-order-config.ts:L28-L28
ExportConfig	interface; public API	additionalColumns	ExportColumn[]	sì		feature-libs/cart/import-export/core/model/export-entries.model.ts:L23-L49
〃		messageEnabled	boolean	sì		
〃		downloadDelay	number	sì		
〃		fileOptions	ExportFileOptions	no		
〃		maxEntries	number	sì		
ImportConfig	interface; public API	fileValidity	FileValidity	sì		feature-libs/cart/import-export/core/model/import-entries.config.ts:L69-L72
〃		cartNameGeneration	CartNameGeneration	sì		
ImportExportConfig	abstract class; @Injectable; public API	cartImportExport	{ file: { separator: string }; import?: ImportConfig; export: ExportConfig; }	sì	defaultImportExportConfig = { cartImportExport: { file: { separator: ',', }, import: { fileValidity: { maxSize: 1, maxEntries: { [OrderEntriesSourc… (feature-libs/cart/import-export/core/config/default-import-export-config.ts:L11-L61)	feature-libs/cart/import-export/core/config/import-export-config.ts:L12-L22

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
QuickOrderConfig	abstract class; @Injectable; public API	quickOrder	{ searchForm?: { displayProductImages: boolean; maxProducts: number; minCharactersBeforeRequest: number; }; list?: { hardDeleteTimeout: number; }; }	sì	defaultQuickOrderConfig = { quickOrder: { searchForm: { displayProductImages: true, maxProducts: 5, minCharactersBeforeRequest: 3, }, list: { har… (feature-libs/cart/quick-order/root/config/default-quick-order.config.ts:L9-L20)	feature-libs/cart/quick-order/root/config/quick-order-config.ts:L10-L25

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultAddedToCartLayoutConfig	LayoutConfig	{ launch: { ADDED_TO_CART: { inlineRoot: true, component: AddedToCartDialogComponent, dialogType: DIALOG_TYPE.DIALOG, }…	no	feature-libs/cart/base/components/added-to-cart-dialog/default-added-to-cart-layout.config.ts:L10-L18
defaultCartConfig	CartConfig	{ cart: { validation: { enabled: false, }, selectiveCart: { enabled: false, }, showRealTimeStockInPDP: { enabled: false…	no	feature-libs/cart/base/root/config/default-cart-config.ts:L9-L21
defaultCartRoutingConfig	RoutingConfig	{ routing: { routes: { cart: { paths: ['cart'] }, }, }, }	no	feature-libs/cart/base/root/config/default-cart-routing-config.ts:L9-L15
defaultClearCartLayoutConfig	LayoutConfig	{ launch: { CLEAR_CART: { inline: true, component: ClearCartDialogComponent, dialogType: DIALOG_TYPE.DIALOG, }, }, }	sì	feature-libs/cart/base/components/clear-cart/clear-cart-dialog/default-clear-cart-layout.config.ts:L10-L18
defaultImportEntriesLayoutConfig	LayoutConfig	{ launch: { IMPORT_TO_CART: { inlineRoot: true, component: ImportEntriesDialogComponent, dialogType: DIALOG_TYPE.DIALOG…	no	feature-libs/cart/import-export/components/import-to-cart/default-import-entries-layout.config.ts:L10-L18
defaultImportExportConfig	ImportExportConfig	{ cartImportExport: { file: { separator: ',', }, import: { fileValidity: { maxSize: 1, maxEntries: { [OrderEntriesSourc…	sì	feature-libs/cart/import-export/core/config/default-import-export-config.ts:L11-L61
defaultQuickOrderConfig	QuickOrderConfig	{ quickOrder: { searchForm: { displayProductImages: true, maxProducts: 5, minCharactersBeforeRequest: 3, }, list: { har…	sì	feature-libs/cart/quick-order/root/config/default-quick-order.config.ts:L9-L20
defaultQuickOrderRoutingConfig	RoutingConfig	{ routing: { routes: { quickOrder: { paths: ['my-account/quick-order'], }, }, }, }	sì	feature-libs/cart/quick-order/root/quick-order-root.module.ts:L36-L44

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultSavedCartFormLayoutConfig	LayoutConfig	{ launch: { SAVED_CART: { inline: true, component: SavedCartFormDialogComponent, dialogType: DIALOG_TYPE.DIALOG, }, }, }	sì	feature-libs/cart/saved-cart/components/saved-cart-form-dialog/default-saved-cart-form-layout.config.ts:L10-L18

@spartacus/cdc

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
CdcConfig	abstract class; @Injectable; public API	cdc	{ baseSite: string; javascriptUrl: string; sessionExpiration: number; }[]	sì		integration-libs/cdc/root/config/cdc-config.ts:L10-L20
Config	interface; extends CdcConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/cdc/root/config/cdc-config.ts:L23-L23

Costanti default*Config

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultCdcReconsentLayoutConfig	LayoutConfig	{ launch: { CDC_RECONSENT: { inlineRoot: true, component: CdcReconsentComponent, dialogType: DIALOG_TYPE.DIALOG, }, }, }	sì	integration-libs/cdc/user-account/login-form/reconsent/default-cdc-reconsent-layout.config.ts:L10-L18
defaultCdcRoutingConfig	RoutingConfig	{ routing: { routes: cdcRoutesConfig, }, }	no	integration-libs/cdc/root/config/default-cdc-routing-config.ts:L37-L41

@spartacus/cds

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
CdsConfig	abstract class; @Injectable; public API	cds	{ baseSite?: string[]; tenant?: string; baseUrl?: string; consentTemplateId?: string; endpoints?: CdsEndpoints; merchandising?: MerchandisingConfig; profileTag…	sì		integration-libs/cds/src/config/cds-config.ts:L13-L27

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends CdsConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/cds/src/config/cds-config.ts:L30-L30
MerchandisingConfig	interface; public API	defaultCarouselViewportThreshold	number	sì		integration-libs/cds/src/config/merchandising.config.ts:L7-L9
ProfileTagConfig	interface; public API	javascriptUrl	string	sì		integration-libs/cds/src/config/profile-tag.config.ts:L7-L14
〃		configUrl	string	sì		
〃		cdsSiteId	string	sì		
〃		allowInsecureCookies	boolean	sì		
〃		gtmId	string	sì		
〃		sciEnabled	boolean	sì		
ProfileTagJsConfig	interface; public API	tenant	string	sì		integration-libs/cds/src/profiletag/model/profile-tag.model.ts:L28-L36
〃		siteId	string	sì		
〃		spa	boolean	no		
〃		javascriptUrl	string	sì		
〃		configUrl	string	sì		
〃		allowInsecureCookies	boolean	sì		
〃		gtmId	string	sì		

@spartacus/checkout

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
CheckoutConfig	abstract class; @Injectable; public API	checkout	{ /** * Set checkout steps as ordered array of pages. */ steps?: Array<CheckoutStep>; /** * Allow for express checkout when default shipping method and payment…	sì	defaultCheckoutConfig = { checkout: { steps: [{ id: 'deliveryAddress', name: 'checkoutProgress.deliveryAddress', routeName: 'checkoutDeliveryA… (feature-libs/checkout/base/root/config/default-checkout-config.ts:L10-L42)	feature-libs/checkout/base/root/config/checkout-config.ts:L18-L51

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends CheckoutConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/checkout/base/root/config/checkout-config.ts:L54-L54

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultB2BCheckoutConfig	CheckoutConfig	{ checkout: { steps: [{ id: 'paymentType', name: 'checkoutB2B.progress.methodOfPayment', routeName: 'checkoutPaymentTy…	sì	feature-libs/checkout/b2b/root/config/default-b2b-checkout-config.ts:L13-L51
defaultCheckoutB2BRoutingConfig	RoutingConfig	{ routing: { routes: { checkoutPaymentType: { paths: ['checkout/payment-type'] }, }, }, }	sì	feature-libs/checkout/b2b/root/config/default-checkout-b2b-routing-config.ts:L9-L15
defaultCheckoutConfig	CheckoutConfig	{ checkout: { steps: [{ id: 'deliveryAddress', name: 'checkoutProgress.deliveryAddress', routeName: 'checkoutDeliveryA…	no	feature-libs/checkout/base/root/config/default-checkout-config.ts:L10-L42
defaultCheckoutRoutingConfig	RoutingConfig	{ routing: { routes: { checkoutLogin: { paths: ['checkout-login'], authFlow: true }, checkout: { paths: ['checkout'] },…	no	feature-libs/checkout/base/root/config/default-checkout-routing-config.ts:L9-L20
defaultOccCheckoutB2BConfig	OccConfig	{ backend: { occ: { endpoints: { ... defaultB2bCheckoutDetailsOccEndpoint, setDeliveryAddress: 'orgUsers/${userId}/carts…	no	feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L16-L29
defaultOccCheckoutConfig	OccConfig	{ backend: { occ: { endpoints: { setDeliveryAddress: DELIVERY_ENDPOINT, cardTypes: 'cardtypes', createDeliveryAddress: …	no	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L12-L35
defaultPlaceOrderSpinnerLayoutConfig	LayoutConfig	{ launch: { PLACE_ORDER_SPINNER: { inline: true, component: SpinnerComponent, dialogType: DIALOG_TYPE.POPOVER_CENTER_BA…	no	feature-libs/checkout/base/components/checkout-place-order/default-place-order-spinner-layout.config.ts:L13-L21

@spartacus/core

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
AnonymousConsentsConfig	abstract class; @Injectable; extends OccConfig; public API	anonymousConsents	{ /** * Specify the consent template ID to be show on the registration page. */ registerConsent?: string; /** * Show the legal description at the top of the an…	sì	defaultAnonymousConsentsConfig = { anonymousConsents: { registerConsent: 'MARKETING_NEWSLETTER', showLegalDescriptionInDialog: true, requiredConsents: […](core-libs/core/src/anonymous-consents/config/default-anonymous-consents-config.ts:L9-L19)	core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L11-L43
AuthConfig	abstract class; @Injectable; public API	authentication	{ /** * OAuth client id. */ client_id?: string; /** * Secret for client required by Hybris OAuth. */ client_secret?: string; /** * Base url for auth server (fo…	sì	defaultAuthConfig = { authentication: { client_id: 'mobile_android_public', tokenEndpoint: '/oauth/token', revokeEndpoint: '/oauth/revoke',… (core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L48-L74)	core-libs/core/src/auth/user-auth/config/auth-config.ts:L32-L143
BackendConfig	interface; extends BackendHttpTimeoutConfig; declare module '../../occ/config/occ-config'; public API	(nessuna chiave propria)				core-libs/core/src/http/http-timeout/http-timeout.config.ts:L40-L40
BackendConfig	interface; public API	occ	{ baseUrl?: string; prefix?: string; /** * Indicates whether or not cross-site Access-Control requests should be made * using credentials such as cookies, auth…	sì		core-libs/core/src/occ/config/occ-config.ts:L13-L40
〃		media	{ /** * Media URLs are typically relative, so that the host can be configured. * Configurable media baseURLs are useful for SEO, multi-site, * switching enviro…	sì		
〃		loadingScopes	LoadingScopes	sì		
BackendHttpTimeoutConfig	abstract class; public API	timeout	HttpTimeoutConfig	sì	defaultBackendHttpTimeoutConfig = { backend: { timeout: { server: 20_000, }, }, } (core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15)	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L35-L37

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Captcha Config	interface; public API	enabled	boolean	no		core-libs/core/src/model/misc.model.ts:L135-L138
〃		publicKey	string	sì		
CmsComponent ChildRoutes Config	interface; public API	parent	Pick<Route, 'data'>	sì		core-libs/core/src/cms/config/cms-config.ts:L48-L58
〃		children	Route[]	sì		
CMSComponent Config	interface; extends StandardCmsComponentConfig, JspIncludeCmsComponentConfig; public API	[component Type: string]	CmsComponentMapping | undefined	-		core-libs/core/src/cms/config/cms-config.ts:L103-L107
CmsConfig	abstract class; @Injectable; extends OccConfig; public API	featureModules	{ [featureName: string]: FeatureModule Config | string }	sì		core-libs/core/src/cms/config/cms-config.ts:L124-L134
〃		cmsComponents	CMSComponentConfig	sì		
〃		components Loading	{ pageSize?: number; }	sì		
CmsPage Config	interface; public API	ignoreBackend	boolean	sì		core-libs/core/src/cms/config/cms-structure.config.ts:L19-L49
〃		pageId	string	sì		
〃		type	string	sì		
〃		title	string	sì		
〃		template	string	sì		
〃		slots	{ [key: string]: Cms PageSlotConfig }	no		
CmsPageSlot Config	interface; public API	componentIds	string[]	sì		core-libs/core/src/cms/config/cms-structure.config.ts:L63-L66
〃		properties	any	sì		
CmsPageSlots Config	interface; public API	[key: string]	CmsPageSlotConfig	-		core-libs/core/src/cms/config/cms-structure.config.ts:L54-L56
CmsStructure Config	abstract class; @Injectable; extends CmsConfig; public API	cmsStructure	{ components?: { [key: string]: ContentSlot ComponentData | any }; pages?: CmsPage Config[]; slots?: Cms PageSlotsConfig; }	sì		core-libs/core/src/cms/config/cms-structure.config.ts:L75-L85

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends AnonymousConsentsConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L46-L46
Config	interface; extends AuthConfig; declare module '../../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/auth/user-auth/config/auth-config.ts:L146-L146
Config	interface; extends CmsConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/cms/config/cms-config.ts:L137-L137
Config	interface; extends CmsStructureConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/cms/config/cms-structure.config.ts:L88-L88
Config	interface; extends PageMetaConfig; declare module '../../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/cms/page/config/page-meta.config.ts:L19-L19
Config	abstract class; @Injectable; public API	(nessuna chiave propria)				core-libs/core/src/config/config-tokens.ts:L17-L21
Config	interface; extends FeaturesConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/features-config/config/features-config.ts:L121-L121
Config	interface; extends FederatedLoginConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/federated-login/config/federated-login-config.ts:L50-L50
Config	interface; extends GlobalMessageConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/global-message/config/global-message-config.ts:L30-L30
Config	interface; extends I18nConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/i18n/config/i18n-config.ts:L95-L95
Config	interface; extends OccConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/occ/config/occ-config.ts:L51-L51

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends RoutingConfig; declare module '../../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L48-L48
Config	interface; extends SiteContextConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/site-context/config/site-context-config.ts:L22-L22
Config	interface; extends SiteThemeConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/site-theme/config/site-theme-config.ts:L22-L22
Config	interface; extends StateConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/state/config/state-config.ts:L38-L38
Config	interface; extends HierarchicalAddressConfig; declare module '../../config/config-tokens'; public API	(nessuna chiave propria)				core-libs/core/src/user/config/hierarchical-address-config.ts:L26-L26
ExternalRoutesConfig	abstract class; @Injectable; public API	routing	ExternalRoutesConfigDefinition	sì		core-libs/core/src/routing/external-routes/external-routes-config.ts:L14-L20
FeatureModuleConfig	interface; public API	module	() => Promise<any>	sì		core-libs/core/src/cms/config/cms-config.ts:L109-L122
〃		dependencies	((() => Promise<any>) | string)[]	sì		
〃		cmsComponents	string[]	sì		
FeaturesConfig	abstract class; @Injectable; public API	features	{ /** * Configure feature level. * Value corresponds to a specific release version number: '1.0', '1.1', etc. * Each subsequent level contains all of the featu…	sì		core-libs/core/src/features-config/config/features-config.ts:L27-L118
FederatedLoginConfig	abstract class; @Injectable; public API	federatedLogin	{ enabled?: boolean; /** URL parameter name to use when passing context from an originating domain to a login domain. */ contextParameterName?: string; /** * L…	sì		core-libs/core/src/federated-login/config/federated-login-config.ts:L10-L47

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
GlobalMessageConfig	abstract class; @Injectable; public API	globalMessages	{ [GlobalMessageType.MSG_TYPE_CONFIRMATION]?: GlobalMessageTypeConfig; [GlobalMessageType.MSG_TYPE_INFO]?: GlobalMessageTypeConfig; [GlobalMessageType.MSG_TYPE…	sì	defaultGlobalMessageConfig = { globalMessages: { [GlobalMessageType.MSG_TYPE_CONFIRMATION]: { timeout: 3000, }, [GlobalMessageType.MSG_TYPE_INFO]: {… (core-libs/core/src/global-message/config/default-global-message-config.ts:L10-L28)	core-libs/core/src/global-message/config/global-message-config.ts:L15-L27
HierarchicalAddressConfig	abstract class; @Injectable; public API	hierarchicalAddress	{ /** * Country isocodes whose addresses use a hierarchical (multi-level) * format: country → region → city → district. For these countries the * address form …	sì	defaultHierarchicalAddressConfig = { hierarchicalAddress: { countriesUsingHierarchicalAddressFormat: ['CN'], }, } (core-libs/core/src/user/config/default-hierarchical-address-config.ts:L9-L13)	core-libs/core/src/user/config/hierarchical-address-config.ts:L10-L23
HttpTimeoutConfig	interface; public API	server	number	sì		core-libs/core/src/http/http-timeout/http-timeout.config.ts:L23-L33
〃		browser	number	sì		
I18nConfig	abstract class; @Injectable; public API	i18n	{ /** * When there are missing translation resources for the active language, the fallback language will be used. */ fallbackLang?: string | false; /** * Confi…	sì	defaultI18nConfig = { i18n: { fallbackLang: undefined, debug: false, }, } (core-libs/core/src/i18n/config/default-i18n-config.ts:L9-L14)	core-libs/core/src/i18n/config/i18n-config.ts:L14-L92
JspIncludeCmsComponentConfig	interface; public API	AccountAddressBookComponent	CmsComponentMapping	sì		core-libs/core/src/cms/config/cms-config.ts:L33-L40
〃		ForgotPasswordComponent	CmsComponentMapping	sì		
〃		ResetPasswordComponent	CmsComponentMapping	sì		
〃		ProductDetailsTabComponent	CmsComponentMapping	sì		
〃		ProductSpecsTabComponent	CmsComponentMapping	sì		
〃		ProductReviewsTabComponent	CmsComponentMapping	sì		
LoadingScopeConfig	interface	include	string[]	sì		core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L23
〃		maxAge	number	sì		
〃		reloadOn	Type<CxEvent>[]	sì		

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
LoadingScopesConfig	interface	[scope: string]	LoadingScopeConfig | undefined	-		core-libs/core/src/occ/config/loading-scopes-config.ts:L25-L27
OccConfig	abstract class; @Injectable; extends SiteContextConfig; public API	backend	BackendConfig	sì	defaultOccConfig = { backend: { occ: { prefix: '/occ/v2/', }, }, } (core-libs/core/src/occ/config/default-occ-config.ts:L9-L15)	core-libs/core/src/occ/config/occ-config.ts:L42-L48
PageMetaConfig	abstract class; @Injectable; public API	pageMeta	PageMetaResolversConfig	sì	defaultPageMetaConfig = { pageMeta: { resolvers: [{ property: 'title', method: 'resolveTitle', }, { property: 'heading', method: 'resolveHeadi… (core-libs/core/src/cms/page/config/default-page-meta.config.ts:L9-L48)	core-libs/core/src/cms/page/config/page-meta.config.ts:L10-L16
PageMetaResolverConfig	interface; public API	property	string	no		core-libs/core/src/cms/page/config/page-meta.config.ts:L31-L41
〃		method	string	no		
PageMetaResolversConfig	interface; public API	resolvers	PageMetaResolverConfig[]	sì		core-libs/core/src/cms/page/config/page-meta.config.ts:L22-L29
〃		canonicalUrl	CanonicalUrlOptions	sì		
ProductLoadingScopeConfig	interface; extends LoadingScopeConfig	include	(ProductScope | string)[]	sì		core-libs/core/src/occ/adapters/product/product-occ-config.ts:L28-L30
ProductScopesConfig	interface; extends LoadingScopesConfig	list	ProductLoadingScopeConfig	sì		core-libs/core/src/occ/adapters/product/product-occ-config.ts:L20-L26
〃		details	ProductLoadingScopeConfig	sì		
〃		attributes	ProductLoadingScopeConfig	sì		
〃		variants	ProductLoadingScopeConfig	sì		
〃		[scope: string]	ProductLoadingScopeConfig | undefined	-		

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
RouteBreadcrumbConfig	interface; public API	raw	string	sì		core-libs/core/src/cms/page/routing/route-page-meta.model.ts:L45-L55
〃		i18n	string	sì		
RouteConfig	interface; public API	paths	string[]	sì		core-libs/core/src/routing/configurable-routes/routes-config.ts:L15-L57
〃		matchers	(UrlMatcher | InjectionToken<UrlMatcherFactory>)[]	sì		
〃		paramsMapping	ParamsMapping	sì		
〃		disabled	boolean	sì		
〃		protected	boolean	sì		
〃		authFlow	boolean	sì		
RoutePageMetaConfig	interface; public API	breadcrumb	string | RouteBreadcrumbConfig	sì		core-libs/core/src/cms/page/routing/route-page-meta.model.ts:L30-L40
〃		resolver	Type<any>	sì		
RoutesConfig	interface; public API	[routeName: string]	RouteConfig	-		core-libs/core/src/routing/configurable-routes/routes-config.ts:L11-L13
RoutingConfig	abstract class; @Injectable; public API	routing	RoutingConfigDefinition	sì		core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L22-L28
SearchConfig	interface; public API	pageSize	number	sì		core-libs/core/src/product/model/search-config.ts:L8-L24
〃		currentPage	number	sì		
〃		sort	string	sì		
〃		filters	string	sì		
〃		query	string	sì		
SiteContextConfig	abstract class; @Injectable; public API	context	{ urlParameters?: string[]; [contextName: string]: string[] | undefined; }	sì		core-libs/core/src/site-context/config/site-context-config.ts:L10-L19

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
SiteTheme Config	abstract class; @Injectable; public API	siteTheme	{ optionalThemes?: Array<SiteTheme>; }	sì	defaultSiteTheme Config = { siteTheme: { optionalThemes: [{ i18nNameKey: 'site ThemeSwitcher.themes. highContrastDark', className: 'cx-theme-high-… (core-libs/ core/src/site-theme/ config/default-site-theme-config.ts:L9-L22)	core-libs/core/ src/site-theme/ config/site-theme-config. ts:L11-L19
StandardCms Component Config	interface; public API	CMSSiteContext Component	CmsComponentMapping	sì		core-libs/core/ src/cms/config/ cms-config.ts: L13-L31
〃		CMSLink Component	CmsComponentMapping	sì		
〃		Simple Responsive Banner Component	CmsComponentMapping	sì		
〃		SimpleBanner Component	CmsComponentMapping	sì		
〃		Banner Component	CmsComponentMapping	sì		
〃		CMSParagraph Component	CmsComponentMapping	sì		
〃		Breadcrumb Component	CmsComponentMapping	sì		
〃		Navigation Component	CmsComponentMapping	sì		
〃		Footer Navigation Component	CmsComponentMapping	sì		
〃		Category Navigation Component	CmsComponentMapping	sì		
〃		ProductAddTo CartComponent	CmsComponentMapping	sì		
〃		MiniCart Component	CmsComponentMapping	sì		
〃		Product Carousel Component	CmsComponentMapping	sì		
〃		SearchBox Component	CmsComponentMapping	sì		
〃		Product References Component	CmsComponentMapping	sì		
〃		CMSTab Paragraph Component	CmsComponentMapping	sì		
〃		LoginComponent	CmsComponentMapping	sì		

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
StateConfig	abstract class; @Injectable; public API	state	{ ssrTransfer?: { keys?: { /** * A set of state keys that should be transferred from server. */ [key: string]: StateTransferType | undefined; }; }	sì		core-libs/core/src/state/config/state-config.ts:L20-L35
TestConfig	interface; extends Config; public API	features	Config['features'] & FeatureTogglesInterface	sì		core-libs/core/src/config/test-config.module.ts:L27-L29
TranslationChunksConfig	interface; public API	[chunk: string]	string[]	-		core-libs/core/src/i18n/translation-resources.ts:L16-L18

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultAnonymousConsentsConfig	AnonymousConsentsConfig	{ anonymousConsents: { registerConsent: 'MARKETING_NEWSLETTER', showLegalDescriptionInDialog: true, requiredConsents: […]	sì	core-libs/core/src/anonymous-consents/config/default-anonymous-consents-config.ts:L9-L19
defaultAuthConfig	AuthConfig	{ authentication: { client_id: 'mobile_android_public', tokenEndpoint: '/oauth/token', revokeEndpoint: '/oauth/revoke',…	no	core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L48-L74
defaultBackendHttpTimeoutConfig	OccConfig	{ backend: { timeout: { server: 20_000, }, }, }	sì	core-libs/core/src/http/http-timeout/default-http-timeout.config.ts:L9-L15
defaultCmsModuleConfig	CmsConfig	{ backend: { occ: { endpoints: { component: 'users/${userId}/cms/components/${id}', components: 'users/${userId}/cms/co…	sì	core-libs/core/src/cms/config/default-cms-config.ts:L9-L24
defaultGlobalMessageConfig	GlobalMessageConfig	{ globalMessages: { [GlobalMessageType.MSG_TYPE_CONFIRMATION]: { timeout: 3000, }, [GlobalMessageType.MSG_TYPE_INFO]: {…	no	core-libs/core/src/global-message/config/default-global-message-config.ts:L10-L28
defaultHierarchicalAddressConfig	HierarchicalAddressConfig	{ hierarchicalAddress: { countriesUsingHierarchicalAddressFormat: ['CN'], }, }	sì	core-libs/core/src/user/config/default-hierarchical-address-config.ts:L9-L13
defaultI18nConfig	I18nConfig	{ i18n: { fallbackLang: undefined, debug: false, }, }	no	core-libs/core/src/i18n/config/default-i18n-config.ts:L9-L14
defaultOccConfig	OccConfig	{ backend: { occ: { prefix: '/occ/v2/', }, }, }	sì	core-libs/core/src/occ/config/default-occ-config.ts:L9-L15
defaultOccCostCentersConfig	OccConfig	{ backend: { occ: { endpoints: { getActiveCostCenters: '/costcenters?fields=DEFAULT,unit(BASIC,addresses(DEFAULT))', },…	no	core-libs/core/src/occ/adapters/cost-center/default-occ-cost-centers-config.ts:L9-L18
defaultOccProductConfig	OccConfig	{ backend: { occ: { endpoints: { product: { default: 'products/${productCode}?fields=DEFAULT,averageRating,images(FULL)…	no	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L10-L60

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccSiteContextConfig	OccConfig	{ backend: { occ: { endpoints: { languages: 'languages', currencies: 'currencies', countries: 'countries', regions: 'co…	no	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L9-L25
defaultOccUserConfig	OccConfig	{ backend: { occ: { endpoints: { paymentDetailsAll: 'users/${userId}/paymentdetails', paymentDetail: 'users/${userId}/p…	no	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L9-L34
defaultPageMetaConfig	PageMetaConfig	{ pageMeta: { resolvers: [{ property: 'title', method: 'resolveTitle', }, { property: 'heading', method: 'resolveHeadi…	sì	core-libs/core/src/cms/page/config/default-page-meta.config.ts:L9-L48
defaultSiteThemeConfig	SiteThemeConfig	{ siteTheme: { optionalThemes: [{ i18nNameKey: 'siteThemeSwitcher.themes.highContrastDark', className: 'cx-theme-high-…	sì	core-libs/core/src/site-theme/config/default-site-theme-config.ts:L9-L22

@spartacus/customer-ticketing

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends CustomerTicketingConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/customer-ticketing/root/config/customer-ticketing-config.ts:L30-L30
CustomerTicketingConfig	abstract class; @Injectable; public API	customerTicketing	{ agentSessionTimer?: { startingDelayInSeconds?: number; }; attachmentRestrictions?: { maxSize?: number; allowedTypes?: string[]; }; inputCharactersLimit?: num…	sì	defaultCustomerTicketingConfig = { customerTicketing: { attachmentRestrictions: { maxSize: MAX_SIZE_FOR_ATTACHMENT, allowedTypes: ['.pdf', '.txt', '.pn… (feature-libs/customer-ticketing/root/config/default-customer-ticketing-config.ts:L15-L37)	feature-libs/customer-ticketing/root/config/customer-ticketing-config.ts:L10-L27
TicketSearchConfig	interface; public API	pageSize	number	sì		feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts:L82-L86
〃		currentPage	number	sì		
〃		sort	string	sì		

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultCustomerTicketingConfig	CustomerTicketingConfig	{ customerTicketing: { attachmentRestrictions: { maxSize: MAX_SIZE_FOR_ATTACHMENT, allowedTypes: ['.pdf', '.txt', '.pn…	sì	feature-libs/customer-ticketing/root/config/default-customer-ticketing-config.ts:L15-L37
defaultCustomerTicketingFormLayoutConfig	LayoutConfig	{ launch: { CUSTOMER_TICKETING_REOPEN: { inline: true, component: CustomerTicketingReopenDialogComponent, dialogType: D…	sì	feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/default-customer-ticketing-form-layout.config.ts:L12-L32
defaultCustomerTicketingRoutingConfig	RoutingConfig	{ routing: { routes: { supportTickets: { paths: ['my-account/support-tickets'], }, supportTicketDetails: { paths: ['my-…	no	feature-libs/customer-ticketing/root/config/default-customer-ticketing-routing-config.ts:L9-L21
defaultOccCustomerTicketingConfig	OccConfig	{ backend: { occ: { endpoints: { getTicket: 'users/${customerId}/tickets/${ticketId}', getTickets: 'users/${customerId}…	no	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L9-L27

@spartacus/digital-payments

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends DigitalPaymentsConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L50-L50
DigitalPaymentsConfig	abstract class; @Injectable	digitalPayments	{ occQueryParams?: { sessionId?: string; signature?: string; billingAddress?: string; country?: string; firstName?: string; lastName?: string; line1?: string; …	sì	defaultDigitalPaymentsConfig = { digitalPayments: { occQueryParams: { sessionId: 'sid', signature: 'sign', billingAddress: 'billingAddress', country: … (integration-libs/digital-payments/src/checkout/adapters/config/default-digital-payments-endpoint.config.ts:L9-L25)	integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L27-L47

Costanti default*Config

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultDigitalPaymentsConfig	DigitalPaymentsConfig	{ digitalPayments: { occQueryParams: { sessionId: 'sid', signature: 'sign', billingAddress: 'billingAddress', country: …	no	integration-libs/digital-payments/src/checkout/adapters/config/default-digital-payments-endpoint.config.ts:L9-L25
defaultDpConfirmationDialogConfig	LayoutConfig	{ launch: { DP_SHOW_CONFIRMATION_DIALOG: { inlineRoot: true, component: DpConfirmationDialogComponent, dialogType: DIAL…	no	integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-confirmation-dialog/default-dp-confirmation-dialog.config.ts:L10-L18

@spartacus/epd-visualization

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
`Config`	interface; extends EpdVisualizationConfig; declare module '@spartacus/core'	`(nessuna chiave propria)`				integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L110-L110
`EpdVisualizationConfig`	abstract class; @Injectable; implements Config; public API	`epdVisualization`	`EpdVisualizationInnerConfig`	sì		integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L15-L24
`EpdVisualizationInnerConfig`	interface; public API	`ui5`	`Ui5Config`	sì		integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L29-L46
〃		`apis`	`VisualizationApiConfig`	sì		
〃		`usageIds`	`UsageIdConfig`	sì		
〃		`visualPicking`	`VisualPickingConfig`	sì		
`Ui5Config`	interface; public API	`bootstrapUrl`	`string`	no		integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L51-L60
`UsageIdConfig`	interface; public API	`folderUsageId`	`UsageId`	no		integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L77-L94
〃		`productUsageId`	`UsageIdDefinition`	no		
`VisualizationApiConfig`	interface; public API	`baseUrl`	`string`	no		integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L65-L72

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
VisualPickingConfig	interface; public API	productReferenceType	string	no		integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L99-L104

@spartacus/estimated-delivery-date

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccCartWithEddConfig	OccConfig	{ backend: { occ: { endpoints: { /* eslint-disable max-len */ ... defaultEddOccEndpoints, /* eslint-enable */ }, }, }, }	no	feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L18-L28

@spartacus/omf

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends OmfConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/omf/order/config/omf-config.ts:L20-L20
OmfConfig	abstract class; @Injectable	omf	OmfConfiguration	sì	defaultOmfConfig = { omf: { guidHttpHeaderName: 'guid', }, } (integration-libs/omf/order/config/default-omf-config.ts:L9-L13)	integration-libs/omf/order/config/omf-config.ts:L11-L17

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOmfConfig	OmfConfig	{ omf: { guidHttpHeaderName: 'guid', }, }	no	integration-libs/omf/order/config/default-omf-config.ts:L9-L13

@spartacus/opf

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
ApplePaySessionConfig	interface; public API	request	any	no		integration-libs/opf/quick-buy/root/model/opf-apple-pay.model.ts:L45-L52
〃		onValidateMerchant	(event: any) => Observable<any>	no		
〃		onShippingContactSelected	(event: any) => Observable<any>	no		
〃		onPaymentMethodSelected	(event: any) => Observable<any>	no		
〃		onShippingMethodSelected	(event: any) => Observable<any>	no		
〃		onPaymentAuthorized	(event: any) => Observable<any>	no		
BackendConfig	interface; extends OpfApiBackendConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/opf/base/root/config/opf-api-config.ts:L20-L20
Config	interface; extends OpfConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/opf/base/root/config/opf-config.ts:L93-L93
Config	interface; extends OpfQuickBuyConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/opf/quick-buy/root/config/opf-quick-buy-config.ts:L27-L27
OpfApiBackendConfig	interface; public API	opfApi	{ endpoints?: OpfApiEndpoints; }	sì		integration-libs/opf/base/root/model/opf-api-backend-config.model.ts:L9-L13
OpfApiConfig	abstract class; @Injectable; extends Config; public API	backend	OpfApiBackendConfig	sì		integration-libs/opf/base/root/config/opf-api-config.ts:L11-L17
OpfCheckoutReviewCardEditConfig	interface; public API	route	string	sì		integration-libs/opf/checkout/components/opf-checkout-review-card/opf-checkout-review-card.model.ts:L7-L10
〃		ariaLabelKey	string	sì		

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
OpfConfig	abstract class; @Injectable; public API	opf	{ opfBaseUrl?: string; commerceCloudPublic Key?: string; payment Option?: { /** * The map of info message translation keys for specific payment configuration IDs…	sì	defaultOpfConfig = { opf: { opfBaseUrl: '', payment Option: { enableInfo Message: true, }, }, } (integration-libs/opf/base/root/config/default-opf-config.ts:L9-L16)	integration-libs/opf/base/root/config/opf-config.ts:L10-L90
OpfPayment Config	interface; public API	configuration Id	string	sì		integration-libs/opf/payment/root/model/opf-payment.model.ts:L215-L223
〃		cartId	string	sì		
〃		resultURL	string	sì		
〃		cancelURL	string	sì		
〃		channel	string	sì		
〃		browserInfo	OpfPaymentBrowserInfo	sì		
OpfPayment Initiation Config	interface; public API	otpKey	string	sì		integration-libs/opf/payment/root/model/opf-payment.model.ts:L195-L198
〃		config	OpfPaymentConfig	sì		
OpfPayment Update Config	interface; public API	payment SessionId	string	no		integration-libs/opf/payment/root/model/opf-payment.model.ts:L200-L204
〃		otpKey	string	sì		
〃		config	OpfPaymentUpdatePayload	sì		
OpfQuickBuy Config	abstract class; @Injectable; public API	providers	{ googlePay: OpfQuick BuyGooglePayProvider; applePay: OpfQuickBuy ApplePayProvider; [key: string]: unknown; }	sì	defaultOpfQuickBuyConfig = { providers: { googlePay: { resourceUrl: 'https://pay.google.com/gp/p/js/pay.js', environment: 'TEST', paymentRequest: … (integration-libs/opf/quick-buy/root/config/default-opf-quick-buy-config.ts:L10-L63)	integration-libs/opf/quick-buy/root/config/opf-quick-buy-config.ts:L14-L24

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultGiftCardOccEndpointsConfig	OccConfig	{ backend: { occ: { endpoints: { applyGiftCard: 'users/${userId}/carts/${cartId}/giftCards', removeGiftCard: 'users/${u…	sì	integration-libs/opf/gift-card/occ/config/default-occ-opf-gift-card-config.ts:L9-L19
defaultGlobalSpinnerLayoutConfig	LayoutConfig	{ launch: { PLACE_ORDER_SPINNER: { inlineRoot: true, component: SpinnerComponent, dialogType: DIALOG_TYPE.POPOVER_CENTE…	no	integration-libs/opf/global-functions/core/config/default-opf-global-layout.config.ts:L14-L27

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccOpfCartConfig	OccConfig	{ backend: { occ: { endpoints: { ...opfCartEndpoints, }, }, }, }	sì	integration-libs/opf/global-functions/root/config/default-occ-opf-cart-config.ts:L14-L22
defaultOccOpfCheckoutConfig	OccConfig	{ backend: { occ: { endpoints: { cartUserEmail: 'users/${userId}/carts/${cartId}?fields=sapCustomerEmail', }, }, }, }	sì	integration-libs/opf/checkout/root/config/default-occ-opf-checkout-config.ts:L9-L18
defaultOccOpfGiftCardCartEndpointsConfig	OccConfig	{ backend: { occ: { endpoints: { ...giftCardCartEndpoints, }, }, }, }	sì	integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-cart-config.ts:L16-L24
defaultOccOpfGiftCardOrderEndpointsConfig	OccConfig	{ backend: { occ: { endpoints: { ...giftCardOrderEndpoints, }, }, }, }	sì	integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-order-config.ts:L17-L25
defaultOccOpfPaymentConfig	OccConfig	{ backend: { occ: { endpoints: { setCartPaymentOption: 'users/${userId}/carts/${cartId}/paymentOption', }, }, }, }	no	integration-libs/opf/payment/occ/config/default-occ-opf-payment-config.ts:L9-L17
defaultOccOpfQuickBuyCartConfig	OccConfig	{ backend: { occ: { endpoints: { quickBuyCreateDeliveryAddress: 'users/${userId}/carts/${cartId}/addresses/delivery', q…	sì	integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts:L9-L24
defaultOpfApiBaseConfig	OpfApiConfig	{ backend: { opfApi: { endpoints: { getActiveConfigurations: 'active-configurations', }, }, }, }	no	integration-libs/opf/base/opf-api/config/default-opf-api-base-config.ts:L9-L17
defaultOpfApiCtaConfig	OpfApiConfig	{ backend: { opfApi: { endpoints: { getCtaScripts: 'payments/cta-scripts-rendering', }, }, }, }	no	integration-libs/opf/cta/opf-api/config/default-opf-api-cta-config.ts:L9-L17
defaultOpfApiPaymentConfig	OpfApiConfig	{ backend: { opfApi: { endpoints: { verifyPayment: 'payments/${paymentSessionId}/verify', submitPayment: 'payments/${pa…	no	integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L9-L23
defaultOpfApiQuickBuyConfig	OpfApiConfig	{ backend: { opfApi: { endpoints: { getApplePayWebSession: 'payments/apple-pay-web-sessions', }, }, }, }	no	integration-libs/opf/quick-buy/opf-api/config/default-opf-api-quick-buy-config.ts:L9-L17

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOpfB2bCheckoutConfig	CheckoutConfig	{ checkout: { flows: { OPF: { steps: opfB2bCheckoutSteps, guest: false, }, }, }, }	sì	integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-config.ts:L46-L55
defaultOpfB2bCheckoutOccEndpointsConfig	OccConfig	{ backend: { occ: { endpoints: { ...orderEndpoints, }, }, }, }	sì	integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-occ-endpoints-config.ts:L14-L22
defaultOpfCheckoutConfig	CheckoutConfig	{ checkout: { flows: { OPF: { steps: opfCheckoutSteps, guest: false, }, }, }, }	sì	integration-libs/opf/checkout/root/config/default-opf-checkout-config.ts:L36-L45
defaultOpfCheckoutRoutingConfig	RoutingConfig	{ routing: { routes: { opfCheckoutPaymentType: { paths: ['checkout/opf-payment-type'], }, opfCheckoutDeliveryAddress: {…	no	integration-libs/opf/checkout/root/config/default-opf-checkout-routing-config.ts:L9-L29
defaultOpfConfig	OpfConfig	{ opf: { opfBaseUrl: '', paymentOption: { enableInfoMessage: true, }, }, }	sì	integration-libs/opf/base/root/config/default-opf-config.ts:L9-L16
defaultOpfErrorModalLayoutConfig	LayoutConfig	{ launch: { OPF_ERROR: { inline: true, component: OpfErrorModalComponent, dialogType: DIALOG_TYPE.POPOVER_CENTER_BACKDR…	sì	integration-libs/opf/base/components/opf-error-modal/default-opf-error-modal.layout.config.ts:L10-L18
defaultOpfGiftCardCartConfig	CartConfig	{ cart: { validation: { enabled: true, }, selectiveCart: { enabled: false, }, showRealTimeStockInPDP: { enabled: false,…	sì	integration-libs/opf/gift-card/root/config/default-gift-card-cart-config.ts:L9-L21
defaultOpfPaymentRoutingConfig	RoutingConfig	{ routing: { routes: { paymentVerificationResult: { paths: ['opf/payment-verification-redirect/result'], }, paymentVeri…	sì	integration-libs/opf/payment/root/config/default-opf-payment-routing-config.ts:L9-L20
defaultOpfQuickBuyConfig	OpfQuickBuyConfig	{ providers: { googlePay: { resourceUrl: 'https://pay.google.com/gp/p/js/pay.js', environment: 'TEST', paymentRequest: …	sì	integration-libs/opf/quick-buy/root/config/default-opf-quick-buy-config.ts:L10-L63

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultPlaceOrderSpinnerLayoutConfig	LayoutConfig	{ launch: { PLACE_ORDER_SPINNER: { inline: true, component: SpinnerComponent, dialogType: DIALOG_TYPE.POPOVER_CENTER_BA…	no	integration-libs/opf/gift-card/root/components/opf-gift-card-checkout/opf-gift-card-checkout.module.ts:L17-L25

@spartacus/opps

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends OppsConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/opps/root/config/opps-config.ts:L28-L28
OppsConfig	abstract class; @Injectable; public API	opps	{ couponcodes?: { httpHeaderName?: string; urlParameter?: string; localStorageKey?: string; }; loginRequired?: { urlParameter?: string; }; }	sì	defaultOppsConfig = { opps: { couponcodes: { httpHeaderName: 'couponcodes', urlParameter: 'couponcodes', localStorageKey: 'opps-couponcodes… (integration-libs/opps/root/config/default-opps-config.ts:L9-L20)	integration-libs/opps/root/config/opps-config.ts:L10-L25

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOppsConfig	OppsConfig	{ opps: { couponcodes: { httpHeaderName: 'couponcodes', urlParameter: 'couponcodes', localStorageKey: 'opps-couponcodes…	sì	integration-libs/opps/root/config/default-opps-config.ts:L9-L20

@spartacus/order

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends OrderConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/order/root/config/order-config.ts:L22-L22
OrderConfig	abstract class; @Injectable; public API	orderAttachments	{ previewMimeTypes: string[]; }	sì	defaultOrderConfig = { orderAttachments: { previewMimeTypes: ['text/plain', 'application/json', 'application/xml', 'application/xhtml+xml',… (feature-libs/order/root/config/default-order-config.ts:L9-L28)	feature-libs/order/root/config/order-config.ts:L10-L19

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
″		showOrderQuoteLink	boolean	sì		

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultConsignmentTrackingLayoutConfig	LayoutConfig	{ launch: { CONSIGNMENT_TRACKING: { inlineRoot: true, component: TrackingEventsComponent, dialogType: DIALOG_TYPE.DIALO…	no	feature-libs/order/components/order-details/order-detail-items/default-consignment-tracking-layout.config.ts:L10-L18
defaultMyAccountV2DownloadInvoicesLayoutConfig	LayoutConfig	{ launch: { DOWNLOAD_ORDER_INVOICES: { inlineRoot: true, component: MyAccountV2DownloadInvoicesComponent, dialogType: D…	sì	feature-libs/order/components/order-details/my-account-v2/download-invoices/default-my-account-v2-download-invoices-layout.config.ts:L10-L18
defaultOccOrderConfig	OccConfig	{ backend: { occ: { endpoints: { orderHistory: 'users/${userId}/orders', orderDetail: 'users/${userId}/orders/${orderId…	no	feature-libs/order/occ/config/default-occ-order-config.ts:L9-L50
defaultOrderAttachmentsDialogLayoutConfig	LayoutConfig	{ launch: { ORDER_ATTACHMENTS: { inline: true, component: OrderAttachmentsDialogComponent, dialogType: DIALOG_TYPE.DIAL…	no	feature-libs/order/components/order-details/order-attachments/default-order-attachments-dialog-layout.config.ts:L10-L18
defaultOrderConfig	OrderConfig	{ orderAttachments: { previewMimeTypes: ['text/plain', 'application/json', 'application/xml', 'application/xhtml+xml',…	no	feature-libs/order/root/config/default-order-config.ts:L9-L28
defaultOrderDocumentFlowDialogLayoutConfig	LayoutConfig	{ launch: { ORDER_DOCUMENT_FLOW: { inlineRoot: true, component: OrderDocumentFlowDialogComponent, dialogType: DIALOG_TY…	no	feature-libs/order/document-flow/components/default-order-document-flow-dialog-layout.config.ts:L10-L18
defaultOrderRoutingConfig	RoutingConfig	{ routing: { routes: { orders: { paths: ['my-account/orders'], }, orderDetails: { paths: ['my-account/order/:orderCode'…	no	feature-libs/order/root/config/default-order-routing-config.ts:L9-L54
defaultReorderLayoutConfig	LayoutConfig	{ launch: { REORDER: { inline: true, component: ReorderDialogComponent, dialogType: DIALOG_TYPE.DIALOG, }, }, }	no	feature-libs/order/components/order-details/reoder-layout.config.ts:L10-L18
defaultReplenishmentOrderCancellationLayoutConfig	LayoutConfig	{ launch: { REPLENISHMENT_ORDER: { inline: true, component: ReplenishmentOrderCancellationDialogComponent, dialogType: …	sì	feature-libs/order/components/replenishment-order-details/default-replenishment-order-cancellation-layout.config.ts:L10-L18

@spartacus/organization

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends OrganizationUIConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/organization/administration/root/config/organization-ui.config.ts:L33-L33
OrganizationUIConfig	abstract class; @Injectable; public API	organizationUI	{ /** * Configuration for list search functionality. */ listSearch?: { /** * Minimum number of characters required to trigger a search. * Default: 3 */ minChar…	sì	defaultOrganizationUIConfig = { organizationUI: { listSearch: { minCharacters: 3, }, }, } (feature-libs/organization/administration/root/config/default-organization-ui.config.ts:L9-L15)	feature-libs/organization/administration/root/config/organization-ui.config.ts:L13-L30

Costanti default*Config

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultAccountSummaryRoutingConfig	RoutingConfig	{ routing: { routes: { orgAccountSummary: { paths: [`${listPath}`], }, orgAccountSummaryDetails: { paths: [`${listPath}…	sì	feature-libs/organization/account-summary/root/config/default-account-summary-routing.config.ts:L11-L23
defaultBudgetRoutingConfig	RoutingConfig	{ routing: { routes: { orgBudget: { paths: ['organization/budgets'], }, orgBudgetCreate: { paths: ['organization/budget…	sì	feature-libs/organization/administration/root/config/default-budget-routing.config.ts:L15-L38
defaultCostCenterRoutingConfig	RoutingConfig	{ routing: { routes: { orgCostCenter: { paths: ['organization/cost-centers'], }, orgCostCenterCreate: { paths: ['organi…	sì	feature-libs/organization/administration/root/config/default-cost-center-routing.config.ts:L15-L42

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccAccountSummaryConfig	OccConfig	{ backend: { occ: { endpoints: { ...accountSummaryHeaderOccEndpoints, }, }, }, }	no	feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L17-L25
defaultOccOrderApprovalConfig	OccConfig	{ backend: { occ: { endpoints: { orderApprovals: '/users/${userId}/orderapprovals', orderApproval: '/users/${userId}/or…	no	feature-libs/organization/order-approval/occ/config/default-occ-organization-config.ts:L9-L21
defaultOccOrganizationConfig	OccConfig	{ backend: { occ: { endpoints: { budgets: '/users/${userId}/budgets', budget: '/users/${userId}/budgets/${budgetCode}',…	no	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L9-L75
defaultOccOrganizationUserRegistrationConfig	OccConfig	{ backend: { occ: { endpoints: { organizationUserRegistration: '/orgUsers', }, }, }, }	no	feature-libs/organization/user-registration/occ/config/default-occ-organization-config.ts:L9-L17
defaultOccUnitOrderConfig	OccConfig	{ backend: { occ: { endpoints: { unitLevelOrderHistory: `/orgUsers/\${userId}/orgUnits/orders`, unitLevelOrderDetail: `…	no	feature-libs/organization/unit-order/occ/config/default-occ-organization-config.ts:L9-L18
defaultOrganizationLayoutConfig		{ layoutSlots: { CompanyPageTemplate: { slots: ['BodyContent'], }, }, } as LayoutConfig	sì	feature-libs/organization/administration/root/config/default-organization-layout.config.ts:L9-L15
defaultOrganizationUIConfig	OrganizationUIConfig	{ organizationUI: { listSearch: { minCharacters: 3, }, }, }	sì	feature-libs/organization/administration/root/config/default-organization-ui.config.ts:L9-L15

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultPermissionRoutingConfig	RoutingConfig	{ routing: { routes: { orgPurchaseLimit: { paths: ['organization/purchase-limits'], }, orgPurchaseLimitCreate: { paths:…	sì	feature-libs/organization/administration/root/config/default-permission-routing.config.ts:L15-L34
defaultUnitsRoutingConfig	RoutingConfig	{ routing: { routes: { orgUnits: { paths: ['organization/units'], }, orgUnitCreate: { paths: ['organization/units/creat…	sì	feature-libs/organization/administration/root/config/default-units-routing.config.ts:L17-L88
defaultUserGroupRoutingConfig	RoutingConfig	{ routing: { routes: { orgUserGroup: { paths: ['organization/user-groups'], }, orgUserGroupCreate: { paths: ['organizat…	sì	feature-libs/organization/administration/root/config/default-user-group-routing.config.ts:L16-L51
defaultUserRoutingConfig	RoutingConfig	{ routing: { routes: { orgUser: { paths: ['organization/users'], }, orgUserCreate: { paths: ['organization/users/create…	sì	feature-libs/organization/administration/root/config/default-user-routing.config.ts:L15-L62

@spartacus/pdf-invoices

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccPDFInvoicesConfig	OccConfig	{ backend: { occ: { endpoints: { pdfInvoicesListInvoices: 'users/${userId}/orders/${orderId}/invoices', pdfInvoicesDown…	no	feature-libs/pdf-invoices/occ/config/default-occ-pdf-invoices-config.ts:L9-L19

@spartacus/pickup-in-store

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
`Config`	interface; extends PickupInStoreConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/pickup-in-store/core/config/pickup-in-store-config.ts:L21-L21
`PickupInStoreConfig`	abstract class; @Injectable; public API	`pickupInStore`	`{ consentTemplateId?: string; }`	sì	`defaultPickupInStoreConfig = { pickupInStore: { consentTemplateId: 'STORE_USER_INFORMATION', }, }` (feature-libs/pickup-in-store/core/config/default-pickup-in-store-config.ts:L9-L13)	feature-libs/pickup-in-store/core/config/pickup-in-store-config.ts:L10-L18

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
`defaultOccPickupLocationConfig`	`OccConfig`	`{ backend: { occ: { endpoints: { storeDetails: 'stores/${storeName}', }, }, }, }`	sì	feature-libs/pickup-in-store/occ/adapters/default-occ-pickup-location-config.ts:L12-L20
`defaultOccStockConfig`	`OccConfig`	`{ backend: { occ: { endpoints: { stock: 'products/${productCode}/stock', stockAtStore: 'products/${productCode}/stock/$…`	sì	feature-libs/pickup-in-store/occ/adapters/default-occ-stock-config.ts:L12-L21
`defaultPickupInStoreConfig`	`PickupInStoreConfig`	`{ pickupInStore: { consentTemplateId: 'STORE_USER_INFORMATION', }, }`	sì	feature-libs/pickup-in-store/core/config/default-pickup-in-store-config.ts:L9-L13
`defaultPickupOptionsDialogLayoutConfig`	`LayoutConfig`	`{ launch: { PICKUP_IN_STORE: { inlineRoot: true, component: PickupOptionDialogComponent, dialogType: DIALOG_TYPE.DIALOG…`	sì	feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog-layout.config.ts:L10-L18

@spartacus/product

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccBulkPricingConfig	OccConfig	{ backend: { occ: { endpoints: { product: { bulkPrices: 'orgProducts/${productCode}?fields=price(DEFAULT),volumePrices(…	no	feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L9-L20
defaultOccFutureStockConfig	OccConfig	{ backend: { occ: { endpoints: { ...futureStockEndpoints, }, }, }, }	no	feature-libs/product/future-stock/occ/config/default-occ-future-stock.config.ts:L15-L23
defaultOccProductVariantsConfig	OccConfig	{ backend: { occ: { endpoints: { product: { variants: 'products/${productCode}?fields=name,purchasable,baseOptions(DEFA…	no	feature-libs/product/variants/occ/config/default-occ-product-variants-config.ts:L9-L20
defaultProductImageZoomLayoutConfig	LayoutConfig	{ launch: { PRODUCT_IMAGE_ZOOM: { inline: true, component: ProductImageZoomDialogComponent, dialogType: DIALOG_TYPE.DIA…	sì	feature-libs/product/image-zoom/components/product-image-zoom/default-product-image-zoom-layout.config.ts:L10-L18

@spartacus/product-configurator

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
CommonConfiguratorUISettingsConfig	abstract class; @Injectable; public API	productConfigurator	CommonProductConfiguratorUISettingsConfig	sì	defaultCommonConfiguratorUISettingsConfig = { productConfigurator: { cartEntryBundleLineItemsThreshold: 10, }, } (feature-libs/product-configurator/common/components/config/default-common-configurator-ui-settings.config.ts:L9-L14)	feature-libs/product-configurator/common/components/config/common-configurator-ui-settings.config.ts:L19-L25

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
CommonProductConfiguratorUISettingsConfig	interface	cartEntryBundleLineItemsThreshold	number	sì		feature-libs/product-configurator/common/components/config/common-configurator-ui-settings.config.ts:L10-L17
Config	interface; declare module '@spartacus/core'	productConfigurator	ProductConfiguratorConfig	sì		feature-libs/product-configurator/common/core/model/product-configurator.config.ts:L12-L14
ConfiguratorAttributeCompositionConfig	abstract class; @Injectable; public API	productConfigurator	ConfiguratorAttributeComposition	sì		feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.config.ts:L18-L24
ConfiguratorCoreConfig	abstract class; @Injectable; public API	productConfigurator	ProductConfiguratorCoreConfig	sì	default ConfiguratorCoreConfig = { productConfigurator: { enableVariantSearch: false, }, } (feature-libs/product-configurator/rulebased/core/config/default-configurator-core.config.ts:L9-L13)	feature-libs/product-configurator/rulebased/core/config/configurator-core.config.ts:L14-L20
ConfiguratorMessageConfig	abstract class; public API	productConfigurator	ProductConfiguratorMessageConfig	sì	default ConfiguratorMessageConfig = { productConfigurator: { updateConfigurationMessage: { waitingTime: 1000, }, }, } (feature-libs/product-configurator/rulebased/components/config/default-configurator-message.config.ts:L9-L15)	feature-libs/product-configurator/rulebased/components/config/configurator-message.config.ts:L13-L15

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
ConfiguratorUISettingsConfig	abstract class; @Injectable; public API	productConfigurator	ProductConfiguratorUISettingsConfig	sì	default ConfiguratorUISettingsConfig = { productConfigurator: { updateDebounceTime: { quantity: 750, input: 500, date: 1500, }, addRetractOption: false, descr… (feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L10-L33)	feature-libs/product-configurator/rulebased/components/config/configurator-ui-settings.config.ts:L51-L57
ProductConfiguratorConfig	interface; extends CommonProductConfiguratorUISettingsConfig; declare module '../../core/model/product-configurator.config'; public API	(nessuna chiave propria)				feature-libs/product-configurator/common/components/config/common-configurator-ui-settings.config.ts:L28-L29
ProductConfiguratorConfig	interface; public API	(nessuna chiave propria)				feature-libs/product-configurator/common/core/model/product-configurator.config.ts:L9-L9
ProductConfiguratorConfig	interface; extends ProductConfiguratorUISettingsConfig, ProductConfiguratorMessageConfig, ConfiguratorAttributeComposition, ProductConfiguratorCoreConfig; declare module '@spartacus/product-configurator/common'	(nessuna chiave propria)				feature-libs/product-configurator/rulebased/components/config/augmented-config.model.ts:L15-L19
ProductConfiguratorCoreConfig	interface	enableVariantSearch	boolean	sì		feature-libs/product-configurator/rulebased/core/config/configurator-core.config.ts:L10-L12
ProductConfiguratorMessageConfig	interface	updateConfigurationMessage	{ waitingTime?: number; }	sì		feature-libs/product-configurator/rulebased/components/config/configurator-message.config.ts:L7-L11

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
ProductConfiguratorUISettingsConfig	interface	updateDebounceTime	{ quantity?: number; input?: number; date?: number; }	sì		feature-libs/product-configurator/rulebased/components/config/configurator-ui-settings.config.ts:L10-L49
〃		addRetractOption	boolean	sì		
〃		descriptions	{ attributeDescriptionLength?: number; valueDescriptionLength?: number; }	sì		
〃		cpqContainerDropDownListThreshold	number	sì		
〃		overviewHeaderSelectors	string[]	sì		
〃		overviewMenuFilterTabVisible	Record<string, boolean>	sì		
〃		cpqHideBasePriceAndSelectedOptions	boolean	sì		

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultCommonConfiguratorUISettingsConfig	CommonConfiguratorUISettingsConfig	{ productConfigurator: { cartEntryBundleLineItemsThreshold: 10, }, }	no	feature-libs/product-configurator/common/components/config/default-common-configurator-ui-settings.config.ts:L9-L14
defaultConfiguratorConflictSolverLayoutConfig	LayoutConfig	{ launch: { CONFLICT_SOLVER: { inlineRoot: true, component: ConfiguratorConflictSolverDialogComponent, dialogType: DIAL…	no	feature-libs/product-configurator/rulebased/components/conflict-solver-dialog/default-configurator-conflict-solver-layout.config.ts:L10-L18
defaultConfiguratorCoreConfig	ConfiguratorCoreConfig	{ productConfigurator: { enableVariantSearch: false, }, }	no	feature-libs/product-configurator/rulebased/core/config/default-configurator-core.config.ts:L9-L13
defaultConfiguratorMessageConfig	ConfiguratorMessageConfig	{ productConfigurator: { updateConfigurationMessage: { waitingTime: 1000, }, }, }	no	feature-libs/product-configurator/rulebased/components/config/default-configurator-message.config.ts:L9-L15
defaultConfiguratorOverviewFilterDialogLayoutConfig	LayoutConfig	{ launch: { CONFIGURATOR_OV_FILTER: { inlineRoot: true, component: ConfiguratorOverviewFilterDialogComponent, dialogTyp…	no	feature-libs/product-configurator/rulebased/components/overview-filter-dialog/default-configurator-overview-filer-dialog-layout.config.ts:L10-L19

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultConfiguratorRestartDialogLayoutConfig	LayoutConfig	{ launch: { CONFIGURATOR_RESTART_DIALOG: { inlineRoot: true, component: ConfiguratorRestartDialogComponent, dialogType:…	no	feature-libs/product-configurator/rulebased/components/restart-dialog/default-configurator-restart-dialog-layout.config.ts:L10-L18
defaultConfiguratorUISettingsConfig	ConfiguratorUISettingsConfig	{ productConfigurator: { updateDebounceTime: { quantity: 750, input: 500, date: 1500, }, addRetractOption: false, descr…	no	feature-libs/product-configurator/rulebased/components/config/default-configurator-ui-settings.config.ts:L10-L33
defaultCpqInteractiveRoutingConfig	RoutingConfig	{ routing: { routes: { configureCLOUDCPQCONFIGURATOR: { paths: ['configure/cpq/:ownerType/entityKey/:entityKey'], }, },…	no	feature-libs/product-configurator/rulebased/root/cpq/default-cpq-interactive-routing-config.ts:L9-L17
defaultCpqOverviewRoutingConfig	RoutingConfig	{ routing: { routes: { configureOverviewCLOUDCPQCONFIGURATOR: { paths: ['configure-overview/cpq/:ownerType/entityKey/:…	no	feature-libs/product-configurator/rulebased/root/cpq/default-cpq-overview-routing-config.ts:L9-L20
defaultOccConfiguratorProductConfig	OccConfig	{ backend: { occ: { endpoints: { product: { configurator: 'products/${productCode}?fields=code, configurable,configurato…	no	feature-libs/product-configurator/common/occ/default-occ-configurator-product-config.ts:L10-L30
defaultRulebasedRoutingConfig	RoutingConfig	{ routing: { routes: { configureCPQCONFIGURATOR: { paths: ['configure/vc/:ownerType/entityKey/:entityKey'], }, configur…	no	feature-libs/product-configurator/rulebased/root/default-rulebased-routing-config.ts:L9-L28
defaultTextfieldRoutingConfig	RoutingConfig	{ routing: { routes: { configureTEXTFIELD: { paths: ['configure/textfield/:ownerType/entityKey/:entityKey'], }, configu…	no	feature-libs/product-configurator/textfield/root/default-textfield-routing-config.ts:L9-L23

@spartacus/product-multi-dimensional

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends ProductMultiDimensionalConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/product-multi-dimensional/selector/root/config/product-multi-dimensional-config.ts:L21-L21
ProductMultiDimensionalConfig	class; @Injectable; public API	multiDimensional	{ imageFormat?: VariantQualifier; }	sì	defaultProductMultiDimensionalConfig = { multiDimensional: { imageFormat: VariantQualifier.STYLE_SWATCH, }, } (feature-libs/product-multi-dimensional/selector/root/config/default-product-multi-dimensional-config.ts:L10-L15)	feature-libs/product-multi-dimensional/selector/root/config/product-multi-dimensional-config.ts:L10-L18

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccProductMultiDimensionalSelectorConfig	OccConfig	{ backend: { occ: { endpoints: { product: { multi_dimensional: 'products/${productCode}?fields=multidimensional,categor…	no	feature-libs/product-multi-dimensional/selector/occ/config/default-occ-product-multi-dimensional-selector-config.ts:L12-L26
defaultProductMultiDimensionalConfig	ProductMultiDimensionalConfig	{ multiDimensional: { imageFormat: VariantQualifier.STYLE_SWATCH, }, }	sì	feature-libs/product-multi-dimensional/selector/root/config/default-product-multi-dimensional-config.ts:L10-L15

@spartacus/punchout

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends PunchoutNavigationGuardConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/punchout/root/config/punchout-guard-config.ts:L37-L37
PunchoutNavigationGuardConfig	abstract class; @Injectable; public API	punchoutNavigation	{ [PunchOutOperation.EDIT]: { allowedUrls?: string[]; allowedCxRoutes?: string[]; redirectPage: string | LaunchRoute; }; [PunchOutOperation.CREATE]: { allowedU…	sì	defaultPunchoutNavigationGuardConfig = { punchoutNavigation: { [PunchOutOperation.INSPECT]: { allowedCxRoutes: [...allowedCxRoutesForAll, 'punchoutInspect'], … (integration-libs/punchout/root/config/default-punchout-guard-config.ts:L22-L46)	integration-libs/punchout/root/config/punchout-guard-config.ts:L12-L34

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccPunchoutConfig	OccConfig	{ backend: { occ: { endpoints: { ...punchoutOccEndpoints, }, }, }, }	no	integration-libs/punchout/occ/config/default-occ-punchout-config.ts:L15-L23
defaultPunchoutCmsComponentsConfig	CmsConfig	{ featureModules: { [PUNCHOUT_FEATURE]: { cmsComponents: ['PunchoutSessionComponent', 'PunchoutErrorComponent', 'Punch…	sì	integration-libs/punchout/root/config/default-punchout-cms-component-config.ts:L10-L23

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultPunchoutNavigationGuardConfig	PunchoutNavigationGuardConfig	{ punchoutNavigation: { [PunchOutOperation.INSPECT]: { allowedCxRoutes: [... allowedCxRoutesForAll, 'punchoutInspect'], …	sì	integration-libs/punchout/root/config/default-punchout-guard-config.ts:L22-L46
defaultPunchoutRoutingConfig	RoutingConfig	{ routing: { routes: { punchoutSession: { paths: ['punchout/cxml/session'], protected: false, authFlow: true, }, puncho…	sì	integration-libs/punchout/root/config/default-punchout-routing-config.ts:L9-L30

@spartacus/qualtrics

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends QualtricsConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/qualtrics/components/qualtrics-loader/config/qualtrics-config.ts:L33-L33
QualtricsConfig	abstract class; @Injectable; public API	qualtrics	{ /** * Deployment script, loaded from a resource, to integrate the deployment of the qualtrics project. * You would typically store the file in the local asse…	sì	defaultQualtricsConfig = { qualtrics: {}, } (feature-libs/qualtrics/components/qualtrics-loader/config/default-qualtrics-config.ts:L9-L11)	feature-libs/qualtrics/components/qualtrics-loader/config/qualtrics-config.ts:L14-L30

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultQualtricsConfig	QualtricsConfig	{ qualtrics: {}, }	no	feature-libs/qualtrics/components/qualtrics-loader/config/default-qualtrics-config.ts:L9-L11

@spartacus/quote

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
`Config`	interface; extends QuoteUIConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/quote/components/config/quote-ui.config.ts:L56-L56
`Config`	interface; extends QuoteCoreConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/quote/core/config/quote-core.config.ts:L32-L32
`Config`	interface; declare module '@spartacus/core'	`quote`	`QuoteConfig`	sì		feature-libs/quote/root/config/quote.config.ts:L12-L14
`QuoteActionsConfig`	interface; public API	`primaryActions`	`QuoteActionType[]`	sì		feature-libs/quote/core/config/quote-core.config.ts:L11-L16
〃		`actionsOrderByState`	`Partial<QuoteActionsByState>`	sì		
`QuoteConfig`	interface; public API	(nessuna chiave propria)				feature-libs/quote/root/config/quote.config.ts:L9-L9
`QuoteCoreConfig`	abstract class; @Injectable; public API	`quote`	`{ actions?: QuoteActionsConfig; }`	sì	defaultQuoteCoreConfig = { quote: { actions: { primaryActions: [QuoteActionType.APPROVE, QuoteActionType.CHECKOUT, QuoteActionType.SUBMIT, Quot… (feature-libs/quote/core/config/default-quote.core.config.ts:L10-L37)	feature-libs/quote/core/config/quote-core.config.ts:L18-L29
`QuoteUIConfig`	abstract class; @Injectable; public API	`quote`	`QuoteUIConfigFragment`	sì	defaultQuoteUIConfig = { quote: { maxCharsForComments: 1000, truncateCardTileContentAfterNumChars: 100, updateDebounceTime: { expiryDate: 500,… (feature-libs/quote/components/config/default-quote-ui.config.ts:L104-L115)	feature-libs/quote/components/config/quote-ui.config.ts:L47-L53

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
`defaultConfirmActionDialogConfig`		{ showWarningNote: false, showExpirationDate: false, showSuccessMessage: true, showOnlyWhenCartIsNotEmpty: false, }	no	feature-libs/quote/components/config/default-quote-ui.config.ts:L15-L20
`defaultOccQuoteConfig`	`OccConfig`	{ backend: { occ: { endpoints: { getQuotes: 'users/${userId}/quotes', createQuote: 'users/${userId}/quotes', getQuote: …	no	feature-libs/quote/occ/config/default-occ-quote-config.ts:L12-L34
`defaultQuoteActionDialogConfig`	`LayoutConfig`	{ launch: { QUOTE_ACTION_CONFIRMATION: { inline: true, component: QuoteConfirmDialogComponent, dialogType: DIALOG_TYPE.…	no	feature-libs/quote/components/confirm-dialog/default-quote-confirm-dialog.config.ts:L10-L18

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultQuoteConfigLayoutConfig	LayoutConfig	{ layoutSlots: { QuoteDetailsPageTemplate: { slots: ['BodyContent', 'CenterRightContent'], }, }, }	sì	feature-libs/quote/root/quote-root.module.ts:L80-L86
defaultQuoteCoreConfig	QuoteCoreConfig	{ quote: { actions: { primaryActions: [QuoteActionType.APPROVE, QuoteActionType.CHECKOUT, QuoteActionType.SUBMIT, Quot…	no	feature-libs/quote/core/config/default-quote.core.config.ts:L10-L37
defaultQuoteRoutingConfig	RoutingConfig	{ routing: { routes: { quotes: { paths: ['my-account/quotes'], }, quoteDetails: { paths: ['my-account/quote/:quoteId'],…	sì	feature-libs/quote/root/quote-root.module.ts:L66-L78
defaultQuoteUIConfig	QuoteUIConfig	{ quote: { maxCharsForComments: 1000, truncateCardTileContentAfterNumChars: 100, updateDebounceTime: { expiryDate: 500,…	no	feature-libs/quote/components/config/default-quote-ui.config.ts:L104-L115

@spartacus/requested-delivery-date

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccRequestedDeliveryDateConfig	OccConfig	{ backend: { occ: { endpoints: { requestedDeliveryDate: 'users/${userId}/carts/${cartId}/requestedretrievaldate', }, },	no	feature-libs/requested-delivery-date/occ/config/default-occ-requested-delivery-date-config.ts:L9-L18

@spartacus/s4-service

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends S4ServiceDeliveryModeConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/s4-service/root/model/augmented-types.model.ts:L63-L63
S4ServiceDeliveryModeConfig	abstract class; @Injectable; public API	s4ServiceDeliveryMode	DeliveryMode	sì		integration-libs/s4-service/root/model/augmented-types.model.ts:L54-L60
ServiceOrderConfig	interface; public API	leadDays	number	sì		integration-libs/s4-service/root/model/checkout-service-details.model.ts:L13-L16

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
″		serviceScheduleTimes	string[]	sì		

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccCheckoutServiceOrderConfig	OccConfig	{ backend: { occ: { endpoints: { ...defaultServiceOrderCheckoutDetailsOccEndpoint, setServiceScheduleSlot: 'users/${use…	no	integration-libs/s4-service/checkout/occ/config/default-occ-checkout-s4-service-config.ts:L15-L25
defaultOccServiceOrderConfig	OccConfig	{ backend: { occ: { endpoints: { ...s4ServiceOccEndpoints, }, }, }, }	no	integration-libs/s4-service/order/occ/config/default-occ-s4-service-config.ts:L17-L25
defaultServiceDeliveryModeConfig	S4ServiceDeliveryModeConfig	{ s4ServiceDeliveryMode: { code: 'service-delivery', deliveryCost: { currencyIso: 'USD', formattedValue: 'USD0.00', pri…	no	integration-libs/s4-service/root/config/default-service-delivery-mode-config.ts:L10-L22
defaultServiceDetailsCheckoutConfig	CheckoutConfig	{ checkout: { steps: [{ id: 'paymentType', name: 'checkoutB2B.progress.methodOfPayment', routeName: 'checkoutPaymentTy…	sì	integration-libs/s4-service/root/config/default-service-details-checkout-config.ts:L13-L57
defaultServiceOrdersRoutingConfig	RoutingConfig	{ routing: { routes: { checkoutServiceDetails: { paths: ['checkout/service-details'] }, cancelServiceDetails: { paths: …	sì	integration-libs/s4-service/root/config/default-service-orders-routing-config.ts:L9-L23

@spartacus/segment-refs

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends SegmentRefsConfig; declare module '@spartacus/core'	(nessuna chiave propria)				integration-libs/segment-refs/root/config/segment-refs-config.ts:L21-L21
SegmentRefsConfig	abstract class; @Injectable	segmentRefs	{ httpHeaderName: string; }	sì	defaultSegmentRefsConfig = { segmentRefs: { httpHeaderName: 'Segmentrefs', }, } (integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13)	integration-libs/segment-refs/root/config/segment-refs-config.ts:L10-L18

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultSegmentRefsConfig	SegmentRefsConfig	{ segmentRefs: { httpHeaderName: 'Segmentrefs', }, }	no	integration-libs/segment-refs/root/config/default-segment-refs-config.ts:L9-L13

@spartacus/setup

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultB2bOccConfig	OccConfig	{ backend: { occ: { endpoints: { ... defaultB2bUserAccountOccEndpoints, ... defaultB2bUserProfileOccEndpoints, ...default…	sì	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L38-L49

@spartacus/smartedit

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends SmartEditConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/smartedit/root/config/smart-edit-config.ts:L22-L22
SmartEditConfig	abstract class; @Injectable; public API	smartEdit	{ storefrontPreviewRoute?: string; allowOrigin?: string; }	sì	defaultSmartEditConfig = { smartEdit: { storefrontPreviewRoute: 'cx-preview', allowOrigin: 'localhost:9002', }, } (feature-libs/smartedit/root/config/default-smart-edit-config.ts:L9-L14)	feature-libs/smartedit/root/config/smart-edit-config.ts:L10-L19

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultSmartEditConfig	SmartEditConfig	{ smartEdit: { storefrontPreviewRoute: 'cx-preview', allowOrigin: 'localhost:9002', }, }	no	feature-libs/smartedit/root/config/default-smart-edit-config.ts:L9-L14

@spartacus/storefinder

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends StoreFinderConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/storefinder/core/config/store-finder-config.ts:L26-L26

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
StoreFinderConfig	abstract class; @Injectable; public API	googleMaps	{ apiUrl?: string; apiKey?: string; scale?: number; selectedMarkerScale?: number; radius?: number; mapId?: string; }	sì	defaultStoreFinderConfig = { googleMaps: { apiUrl: 'https://maps.googleapis.com/maps/api/js', apiKey: '', scale: 5, selectedMarkerScale: 17, radiu… (feature-libs/storefinder/core/config/default-store-finder-config.ts:L9-L21)	feature-libs/storefinder/core/config/store-finder-config.ts:L10-L23

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccStoreFinderConfig	OccConfig	{ backend: { occ: { endpoints: { store: 'stores/${storeId}?fields=FULL', stores: 'stores?fields=stores(name,displayName…	no	feature-libs/storefinder/occ/adapters/default-occ-store-finder-config.ts:L9-L20
defaultStoreFinderConfig	StoreFinderConfig	{ googleMaps: { apiUrl: 'https://maps.googleapis.com/maps/api/js', apiKey: '', scale: 5, selectedMarkerScale: 17, radiu…	no	feature-libs/storefinder/core/config/default-store-finder-config.ts:L9-L21
defaultStoreFinderLayoutConfig	LayoutConfig	{ layoutSlots: { StoreFinderPageTemplate: { slots: ['MiddleContent', 'SideContent'], }, }, }	sì	feature-libs/storefinder/root/config/default-store-finder-layout-config.ts:L9-L15
defaultStoreFinderRoutingConfig	RoutingConfig	{ routing: { routes: { storeFinder: { paths: ['store-finder'], }, storeFinderFind: { paths: ['store-finder/find'], }, s…	sì	feature-libs/storefinder/root/config/default-store-finder-routing-config.ts:L9-L35

@spartacus/storefront

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE DEFAULT (STESSA RADICE)	RIFERIMENTO
AutoFocusConfig	interface; extends EscapeFocusConfig	autofocus	boolean | string	sì	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L72-L96
〃		refreshFocus	unknown	sì	
BaseFocusConfig	interface	(nessuna chiave propria)			core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L17-L17

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE DEFAULT (STESSA RADICE)	RIFERIMENTO
BlockFocusConfig	interface; extends VisibleFocusConfig	block	boolean	sì	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L27-L29
CaptchaApiConfig	abstract class; @Injectable; public API	captchaRenderer	Type<CaptchaRenderer>	sì	core-libs/storefront/shared/components/captcha/captcha-api-config.ts:L11-L29
〃		apiUrl	string	sì	
〃		fields	{ [key: string]: string }	sì	
Config	interface; extends IconConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/cms-components/misc/icon/icon.model.ts:L88-L88
Config	interface; extends LogoutConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/cms-components/user/logout/config/logout-config.ts:L28-L28
Config	interface; extends LcpCmsComponentsConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/cms-structure/cms-lcp-context/config/lcp-cms-components.config.ts:L51-L51
Config	interface; extends PWAModuleConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L29-L29
Config	interface; extends SeoConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/cms-structure/seo/config/seo.config.ts:L27-L27
Config	interface; extends SkipLinkConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/layout/a11y/skip-link/config/skip-link.config.ts:L31-L31
Config	interface; extends LayoutConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/layout/config/layout-config.ts:L128-L128
Config	interface; extends DirectionConfig; declare module '@spartacus/core'	(nessuna chiave propria)			core-libs/storefront/layout/direction/config/direction.config.ts:L34-L34

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends OnNavigateConfig; declare module '@spartacus/core'	(nessuna chiave propria)				core-libs/storefront/router/config/on-navigate-config.ts:L30-L30
Config	interface; extends CaptchaApiConfig; declare module '@spartacus/core'	(nessuna chiave propria)				core-libs/storefront/shared/components/captcha/captcha-api-config.ts:L32-L32
Config	interface; extends PaginationConfig; declare module '@spartacus/core'	(nessuna chiave propria)				core-libs/storefront/shared/components/list-navigation/pagination/config/pagination.config.ts:L20-L20
Config	interface; extends MediaConfig; declare module '@spartacus/core'	(nessuna chiave propria)				core-libs/storefront/shared/components/media/media.config.ts:L80-L80
Config	interface; extends TableConfig; declare module '@spartacus/core'	(nessuna chiave propria)				core-libs/storefront/shared/components/table/config/table.config.ts:L63-L63
Config	interface; extends FormConfig; declare module '@spartacus/core'	(nessuna chiave propria)				core-libs/storefront/shared/config/form-config.ts:L21-L21
Config	interface; extends ViewConfig; declare module '@spartacus/core'	(nessuna chiave propria)				core-libs/storefront/shared/config/view-config.ts:L29-L29
DirectionConfig	abstract class; @Injectable; public API	direction	Direction	sì	defaultDirectionConfig = { direction: { detect: true, default: DirectionMode.LTR, // we're not polluting the system with all defaults for ltr, b… (core-libs/storefront/layout/direction/config/default-direction.config.ts:L10-L19)	core-libs/storefront/layout/direction/config/direction.config.ts:L25-L31
EscapeFocusConfig	interface; extends PersistFocusConfig	focusOnEscape	boolean	sì		core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L64-L70
〃		focusOnDoubleEscape	boolean	sì		

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
FocusConfig	interface; extends LockFocusConfig; public API	(nessuna chiave propria)				core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L149-L149
FormConfig	abstract class; @Injectable; public API	form	{ passwordVisibilityToggle: boolean; }	sì	defaultFormConfig = { form: { passwordVisibilityToggle: true, }, } (core-libs/storefront/shared/config/default-form-config.ts:L9-L13)	core-libs/storefront/shared/config/form-config.ts:L10-L18
IconConfig	abstract class; @Injectable; public API	icon	IconOptions	sì	defaultIconConfig = { icon: { flipDirection: { CARET_RIGHT: DirectionMode.RTL, CARET_LEFT: DirectionMode.RTL, }, }, } (core-libs/storefront/cms-components/misc/icon/default-icon.config.ts:L10-L17)	core-libs/storefront/cms-components/misc/icon/icon.model.ts:L79-L85
LaunchConfig	interface; public API	[key: string]	LaunchOptions	-		core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L9-L11
LayoutConfig	abstract class; @Injectable; public API	breakpoints	LayoutBreakPoints	sì	defaultLayoutConfig = { breakpoints: { xs: 576, sm: 768, md: 992, lg: 1200, xl: { min: 1200, }, }, } (core-libs/storefront/layout/config/default-layout.config.ts:L9-L19)	core-libs/storefront/layout/config/layout-config.ts:L90-L125
〃		layoutSlots	LayoutSlotConfig	sì		
〃		deferredLoading	{ /** * The global strategy will be used as a fallback strategy for all DOM creation, * but can be overridden by local configuration, i.e. for cms components. …	sì		
〃		launch	LaunchConfig	sì		
LcpCmsComponentsConfig	abstract class; @Injectable; public API	lcpCmsComponents	{ /** * If a CMS component ID includes this substring, Spartacus treats it as containing * the LCP (Largest Contentful Paint) element. You can append this subs…	sì	defaultLcpCmsComponentsConfig = { lcpCmsComponents: { idMarker: '__cxLCP__', ids: [], }, } (core-libs/storefront/cms-structure/cms-lcp-context/config/default-lcp-cms-components.config.ts:L9-L14)	core-libs/storefront/cms-structure/cms-lcp-context/config/lcp-cms-components.config.ts:L14-L48
LockFocusConfig	interface; extends TrapFocusConfig	lock	boolean	sì		core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L142-L147

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
LogoutConfig	abstract class; @Injectable; public API	logout	{ /** * Route to redirect to after a successful logout. * Accepts a semantic route name registered in the routing config (e.g. `'home'`, `'login'`), * or a raw…	sì	defaultLogoutConfig = { logout: { redirectRoute: 'home', }, } (core-libs/storefront/cms-components/user/logout/config/default-logout-config.ts:L9-L13)	core-libs/storefront/cms-components/user/logout/config/logout-config.ts:L10-L25
MediaConfig	abstract class; @Injectable; public API	media	{ /** * Picture element configuration holds the media queries assigned to * a format. * The order of formats matters. * <source> elements in <picture> will be …	sì		core-libs/storefront/shared/components/media/media.config.ts:L19-L77
〃		mediaFormats	{ /** * Represents the media format code, that is the key to distinguish different * media in a container. */ [format: string]: MediaFormatSize; }	sì		
〃		imageLoadingStrategy	ImageLoadingStrategy	sì		
OnNavigateConfig	abstract class; @Injectable; public API	enableResetViewOnNavigate	{ active?: boolean; ignoreQueryString?: boolean; ignoreRoutes?: string[]; /** * When set, finds the element with the tag name matching this string * to return …	sì	defaultOnNavigateConfig = { enableResetViewOnNavigate: { active: true, ignoreQueryString: false, ignoreRoutes: [], selectedHostElement: 'body', }… (core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16)	core-libs/storefront/router/config/on-navigate-config.ts:L10-L27
PaginationConfig	abstract class; @Injectable; public API	pagination	PaginationOptions	sì	defaultPaginationConfig = { pagination: { addStart: true, addEnd: true, }, } (core-libs/storefront/shared/components/list-navigation/pagination/config/default-pagination.config.ts:L9-L14)	core-libs/storefront/shared/components/list-navigation/pagination/config/pagination.config.ts:L11-L17
PersistFocusConfig	interface; extends BlockFocusConfig	key	string	sì		core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L31-L62
〃		group	string	sì		
〃		focusTargetSelector	string	sì		
〃		clearOnRestore	boolean	sì		
PWAModuleConfig	abstract class; @Injectable; public API	pwa	{ enabled?: boolean; addToHomeScreen?: boolean; }	sì	defaultPWAModuleConfig = { pwa: { enabled: false, addToHomeScreen: false, }, } (core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L21-L26)	core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L10-L19

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
SearchBoxConfig	interface; public API	displaySuggestions	boolean | string	sì		core-libs/storefront/cms-components/navigation/search-box/search-box.model.ts:L7-L18
〃		displayProducts	boolean | string	sì		
〃		displayProductImages	boolean | string	sì		
〃		maxProducts	number	sì		
〃		maxSuggestions	number	sì		
〃		minCharactersBeforeRequest	number	sì		
〃		recentSearches	boolean | string	sì		
〃		maxRecentSearches	number	sì		
〃		trendingSearches	boolean | string	sì		
〃		maxTrendingSearches	number	sì		
SeoConfig	abstract class; @Injectable; public API	seo	SeoOptions	sì	defaultSeoConfig = { seo: { structuredData: { disableInDevMode: true, }, }, } (core-libs/storefront/cms-structure/seo/config/default-seo.config.ts:L9-L15)	core-libs/storefront/cms-structure/seo/config/seo.config.ts:L10-L16
SkipFocusConfig	interface; public API	isEnabled	boolean	no		core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L17-L24
〃		activeElementSelectors	string[]	sì		
SkipLinkConfig	abstract class; @Injectable; public API	skipLinks	SkipLink[]	sì	defaultSkipLinkConfig = { skipLinks: [{ key: 'cx-header', i18nKey: 'skipLink.labels.header', }, { key: 'cx-main', i18nKey: 'skipLink.labels.ma… (core-libs/storefront/layout/a11y/skip-link/config/default-skip-link.config.ts:L9-L21)	core-libs/storefront/layout/a11y/skip-link/config/skip-link.config.ts:L10-L16

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
TabConfig	interface; public API	label	Translatable | string	sì	defaultTabConfig = { openTabs: [0], breakpoint: BREAKPOINT.md, } (core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L39-L42)	core-libs/storefront/cms-components/content/tab/tab.model.ts:L39-L63
〃		mode	TAB_MODE	sì		
〃		breakpoint	BREAKPOINT	sì		
〃		openTabs	number[]	sì		
〃		restrictDirectionKeys	boolean	sì		
TabFocusConfig	interface; extends AutoFocusConfig	tab	boolean | 'scroll' | string	sì		core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L98-L100
TableConfig	abstract class; @Injectable; public API	table	{ [tableType: string]: ResponsiveTableConfiguration; }	sì	defaultTableConfig = { tableOptions: { headerComponent: TableHeaderCellComponent, dataComponent: TableDataCellComponent, }, } (core-libs/storefront/shared/components/table/config/default-table.config.ts:L11-L16)	core-libs/storefront/shared/components/table/config/table.config.ts:L20-L43
〃		tableOptions	{ /** * Global component to render table header _content_ (`<th>...</th>`). A specific component * can be configured alternatively per table or table field. */…	sì		
TrapFocusConfig	interface; extends TabFocusConfig	trap	TrapFocusType	sì		core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L135-L140
〃		trapTabOnly	TrapFocusType	sì		
ViewConfig	abstract class; @Injectable; public API	view	{ /** * Configurations related to the view of the application */ defaultPageSize?: number; infiniteScroll?: { active?: boolean; productLimit?: number; showMore…	sì	defaultViewConfig = { view: { defaultPageSize: 12, infiniteScroll: { active: false, productLimit: 0, showMoreButton: false, }, }, } (core-libs/storefront/cms-components/product/config/default-view-config.ts:L9-L18)	core-libs/storefront/shared/config/view-config.ts:L10-L26
VisibleFocusConfig	interface	disableMouseFocus	boolean	sì		core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L19-L25

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultAnonymousConsentLayoutConfig	LayoutConfig	{ launch: { ANONYMOUS_CONSENT: { inlineRoot: true, component: AnonymousConsentDialogComponent, dialogType: DIALOG_TYPE.…	no	core-libs/storefront/cms-components/anonymous-consent-management/default-anonymous-consent-layout.config.ts:L11-L19
defaultCouponLayoutConfig	LayoutConfig	{ launch: { COUPON: { inlineRoot: true, component: CouponDialogComponent, dialogType: DIALOG_TYPE.DIALOG, }, CLAIM_DIAL…	no	core-libs/storefront/cms-components/myaccount/my-coupons/default-coupon-card-layout.config.ts:L11-L24
defaultDirectionConfig	DirectionConfig	{ direction: { detect: true, default: DirectionMode.LTR, // we're not polluting the system with all defaults for ltr, b…	no	core-libs/storefront/layout/direction/config/default-direction.config.ts:L10-L19
defaultFormConfig	FormConfig	{ form: { passwordVisibilityToggle: true, }, }	no	core-libs/storefront/shared/config/default-form-config.ts:L9-L13
defaultIconConfig	IconConfig	{ icon: { flipDirection: { CARET_RIGHT: DirectionMode.RTL, CARET_LEFT: DirectionMode.RTL, }, }, }	no	core-libs/storefront/cms-components/misc/icon/default-icon.config.ts:L10-L17
defaultLayoutConfig	LayoutConfig	{ breakpoints: { xs: 576, sm: 768, md: 992, lg: 1200, xl: { min: 1200, }, }, }	sì	core-libs/storefront/layout/config/default-layout.config.ts:L9-L19
defaultLcpCmsComponentsConfig	LcpCmsComponentsConfig	{ lcpCmsComponents: { idMarker: '__cxLCP__', ids: [], }, }	no	core-libs/storefront/cms-structure/cms-lcp-context/config/default-lcp-cms-components.config.ts:L9-L14
defaultLogoutConfig	LogoutConfig	{ logout: { redirectRoute: 'home', }, }	sì	core-libs/storefront/cms-components/user/logout/config/default-logout-config.ts:L9-L13

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOnNavigateConfig	OnNavigateConfig	{ enableResetViewOnNavigate: { active: true, ignoreQueryString: false, ignoreRoutes: [], selectedHostElement: 'body', }…	sì	core-libs/storefront/router/config/default-on-navigate-config.ts:L9-L16
defaultPaginationConfig	PaginationConfig	{ pagination: { addStart: true, addEnd: true, }, }	sì	core-libs/storefront/shared/components/list-navigation/pagination/config/default-pagination.config.ts:L9-L14
defaultPWAModuleConfig	PWAModuleConfig	{ pwa: { enabled: false, addToHomeScreen: false, }, }	sì	core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L21-L26
defaultSeoConfig	SeoConfig	{ seo: { structuredData: { disableInDevMode: true, }, }, }	sì	core-libs/storefront/cms-structure/seo/config/default-seo.config.ts:L9-L15
defaultSkipLinkConfig	SkipLinkConfig	{ skipLinks: [{ key: 'cx-header', i18nKey: 'skipLink.labels.header', }, { key: 'cx-main', i18nKey: 'skipLink.labels.ma…	sì	core-libs/storefront/layout/a11y/skip-link/config/default-skip-link.config.ts:L9-L21
defaultStockNotificationLayoutConfig	LayoutConfig	{ launch: { STOCK_NOTIFICATION: { inlineRoot: true, component: StockNotificationDialogComponent, dialogType: DIALOG_TYP…	no	core-libs/storefront/cms-components/product/stock-notification/stock-notification-dialog/default-stock-notification-layout.config.ts:L11-L19
defaultTabConfig		{ openTabs: [0], breakpoint: BREAKPOINT.md, }	no	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L39-L42

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultTableConfig	TableConfig	{ tableOptions: { headerComponent: TableHeaderCellComponent, dataComponent: TableDataCellComponent, }, }	sì	core-libs/storefront/shared/components/table/config/default-table.config.ts:L11-L16
defaultViewConfig	ViewConfig	{ view: { defaultPageSize: 12, infiniteScroll: { active: false, productLimit: 0, showMoreButton: false, }, }, }	sì	core-libs/storefront/cms-components/product/config/default-view-config.ts:L9-L18

@spartacus/subscription-billing

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends ExtendSubscriptionFrequencyDropdownOptionsConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/subscription-billing/root/config/extend-subscription-dropdown-options-config.ts:L22-L22
ExtendSubscriptionFrequencyDropdownOptionsConfig	abstract class; @Injectable; public API	Months	number	sì		feature-libs/subscription-billing/root/config/extend-subscription-dropdown-options-config.ts:L10-L19
〃		Years	number	sì		
〃		Days	number	sì		
〃		Weeks	number	sì		

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccSubscriptionBillingConfig	OccConfig	{ backend: { occ: { endpoints: { ... subscriptionBillingEndpoints, product: { subscription: 'products/${productCode}?fie…	no	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L31-L43
defaultSubscriptionBillingRoutingConfig	RoutingConfig	{ routing: { routes: { subscriptions: { paths: ['my-account/subscriptions'], }, subscriptionDetails: { paths: ['my-acco…	sì	feature-libs/subscription-billing/root/config/default-subscription-billing-routing-config.ts:L9-L28

@spartacus/tracking

Classi e interfacce `*Config`

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
`AepCollectorConfig`	interface; extends TmsCollectorConfig	`scriptUrl`	`string`	sì		feature-libs/tracking/tms/aep/config/default-aep.config.ts:L10-L12
`Config`	interface; extends PersonalizationConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/tracking/personalization/root/config/personalization-config.ts:L29-L29
`Config`	interface; extends TmsConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/tracking/tms/core/config/tms-config.ts:L50-L50
`GtmCollectorConfig`	interface; extends TmsCollectorConfig	`gtmId`	`string`	sì		feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L10-L12
`PersonalizationConfig`	abstract class; @Injectable; public API	`personalization`	`{ enabled?: boolean; httpHeaderName?: { id: string; timestamp: string; }; context?: { slotPosition: string; componentId: string; }; }`	sì	`defaultPersonalizationConfig = { personalization: { enabled: false, httpHeaderName: { id: 'Occ-Personalization-Id', timestamp: 'Occ-Personalization-Ti…` (feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L9-L21)	feature-libs/tracking/personalization/root/config/personalization-config.ts:L10-L26
`TmsCollectorConfig`	interface; public API	`debug`	`boolean`	sì		feature-libs/tracking/tms/core/config/tms-config.ts:L14-L29
〃		`dataLayerProperty`	`string`	sì		
〃		`events`	`AbstractType<CxEvent>[]`	sì		
〃		`collector`	`Type<TmsCollector>`	sì		
`TmsConfig`	abstract class; @Injectable; public API	`tagManager`	`TmsCollectors`	sì		feature-libs/tracking/tms/core/config/tms-config.ts:L38-L47

Costanti `default*Config`

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
`defaultAdobeExperiencePlatformConfig`	`TmsConfig`	`{ tagManager: { aep: { collector: AepCollectorService, }, }, }`	no	feature-libs/tracking/tms/aep/config/default-aep.config.ts:L20-L26

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultGoogleTagManagerConfig	TmsConfig	{ tagManager: { gtm: { collector: GtmCollectorService, }, }, }	no	feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L20-L26
defaultPersonalizationConfig	PersonalizationConfig	{ personalization: { enabled: false, httpHeaderName: { id: 'Occ-Personalization-Id', timestamp: 'Occ-Personalization-Ti…	no	feature-libs/tracking/personalization/root/config/default-personalization-config.ts:L9-L21

@spartacus/user

Classi e interfacce *Config

CONFIG	TIPO	CHIAVE	TIPO CHIAVE	OPZIONALE	DEFAULT (STESSA RADICE)	RIFERIMENTO
Config	interface; extends UserAccountConfig; declare module '@spartacus/core'	(nessuna chiave propria)				feature-libs/user/account/root/config/user-account-config.ts:L26-L26
UserAccountConfig	abstract class; @Injectable; public API	userAccount	{ /** * When enabled, applies the user's preferred currency (from OCC GET /users/{userId}) * on login, and reverts to the pre-login currency stored in localSto…	sì	defaultUserAccountConfig = { userAccount: { enableUserCurrencySync: false, }, } (feature-libs/user/account/root/config/default-user-account-config.ts:L9-L13)	feature-libs/user/account/root/config/user-account-config.ts:L10-L23

Costanti default*Config

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultCloseDialogModalLayoutConfig	LayoutConfig	{ launch: { CLOSE_ACCOUNT: { inline: true, component: CloseAccountModalComponent, dialogType: DIALOG_TYPE.DIALOG, }, },…	no	feature-libs/user/profile/components/close-account/components/close-account-modal/default-close-account-modal-layout.config.ts:L10-L18
defaultOccUserAccountConfig	OccConfig	{ backend: { occ: { endpoints: { user: 'users/${userId}', createVerificationToken: 'users/anonymous/verificationToken',…	sì	feature-libs/user/account/occ/adapters/config/default-occ-user-account-endpoint.config.ts:L9-L18

COSTANTE	TIPO DICHIARATO	VALORE (TRONCATO 120)	PUBLIC API	RIFERIMENTO
defaultOccUserProfileConfig	OccConfig	{ backend: { occ: { endpoints: { userRegister: 'users', userForgotPassword: 'forgottenpasswordtokens', userRestoreToken…	sì	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L9-L23
defaultOccUserWishlistConfig	OccConfig	{ backend: { occ: { endpoints: { getWishlists: 'users/${userId}/wishlists', getWishlistEntries: 'users/${userId}/wishli…	sì	feature-libs/user/wishlist/occ/adapters/config/default-occ-user-wishlist-endpoint.config.ts:L9-L21
defaultSuggestedAddressesDialogLayoutConfig	LayoutConfig	{ launch: { SUGGESTED_ADDRESSES: { inlineRoot: true, component: SuggestedAddressDialogComponent, dialogType: DIALOG_TYP…	no	feature-libs/user/profile/components/address-book/address-form/suggested-addresses-dialog/default-suggested-addresses-dialog-layout.config.ts:L10-L18
defaultUserAccountConfig	UserAccountConfig	{ userAccount: { enableUserCurrencySync: false, }, }	sì	feature-libs/user/account/root/config/default-user-account-config.ts:L9-L13
defaultVerificationTokenLayoutConfig	LayoutConfig	{ launch: { [LAUNCH_CALLER.ACCOUNT_VERIFICATION_TOKEN]: { inlineRoot: true, component: VerificationTokenDialogComponent…	no	feature-libs/user/account/components/verification-token-form/default-verification-token-layout.config.ts:L14-L22

A03

Componenti CMS (cmsComponents)

Mapping dei type code in `cmsComponents`

IN QUESTO CAPITOLO

Conteggi

Come rigenerare

Tabella 1 — Mapping type code → componente

Tabella 2 — Type code lazy per feature (featureModules.<feature>.cmsComponents)

Generata da `tools/extract2/a03_cms_components.py` allo SHA 5b7a1c2b. Non modificare a mano: rigenerare.

Tabella 1: ogni chiave dentro un object literal `cmsComponents: { ... }` (in `provideDefaultConfig`, `provideConfig`, `<CmsConfig>{...}`, factory, ecc.). Un type code compare più volte se è mappato in più punti (es. override in librerie diverse o varianti). "Contenitore" è la dichiarazione top-level che racchiude il mapping (classe `@NgModule` o funzione/const factory). "Altre chiavi" elenca le proprietà diverse da `component`/`guards`/`providers` (es. `childRoutes`, `deferLoading`, `data`). La colonna "Lazy in feature" incrocia con la tabella 2.

Tabella 2: per ogni `featureModules.<feature>.cmsComponents` nei root module, i type code che, quando presenti in pagina, causano il caricamento lazy della feature. Le chiavi calcolate (`[CART_BASE_FEATURE]`) e le liste referenziate per costante (`CHECKOUT_BASE_CMS_COMPONENTS`) sono risolte tramite la tabella delle costanti stringa/array del codice.

Conteggi

- Occorrenze di mapping `cmsComponents: { ... }` (righe tabella 1): **258**
- Type code distinti mappati: **229**
- File che dichiarano mapping: **201**
- Entry point (librerie) coinvolti: **54**
- Mapping con `guards`: **132**
- Mapping con `providers`: **43**
- Feature lazy con `cmsComponents: [...]` (righe tabella 2): **50**
- Type code dichiarati nelle liste lazy (con ripetizioni / distinti): **279 / 193**
- Type code lazy senza mapping statico trovato in ambito: **7**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract2/a03_cms_components.py
# oppure tutte le appendici: bash tools/extract2/run_all.sh
```

Ambito: `git ls-files core-libs feature-libs integration-libs | grep '\.ts$' | grep -vE '\.spec\.ts$|/testing/|\.mock\.ts$|/schematics/'`. La colonna "Libreria" è l'entry point npm ricavato dal `package.json`/`ng-package.json` più vicino.

Tabella 1 — Mapping type code → componente

#	TYPECODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
1	AccountAddressBookComponent	AddressBookComponent	[AuthGuard]			userProfile	AddressBookModule	@spartacus/user/profile/components	feature-libs/user/profile/components/address-book/address-book.module.ts:L36-L39
2	AccountMyQuotesComponent	QuoteListComponent	[AuthGuard]			quote	QuoteListModule	@spartacus/quote/components	feature-libs/quote/components/list/quote-list.module.ts:L43-L46
3	AccountOrderAttachmentsComponent	OrderAttachmentsComponent	[AuthGuard]			order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L168-L171
4	AccountOrderDetailsActionsComponent	MyAccountV2OrderDetailsActionsComponent				order	myAccountV2CmsMapping	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L82-L85
5	AccountOrderDetailsActionsComponent	OrderDetailActionsComponent	[AuthGuard]			order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L130-L133
6	AccountOrderDetailsActionsComponent	S4ServiceOrderDetailActionsComponent				order	S4ServiceOrderDetailActionsModule	@spartacus/s4-service/order	integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.module.ts:L31-L34
7	AccountOrderDetailsApprovalDetailsComponent	OrderDetailPermissionResultsComponent				organizationOrderApproval	OrderApprovalDetailsModule	@spartacus/organization/order-approval	feature-libs/organization/order-approval/components/details/order-approval-details.module.ts:L67-L69
8	AccountOrderDetailsGroupedItemsComponent	OrderDetailItemsComponent	[AuthGuard]		data	order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L141-L148
9	AccountOrderDetailsItemsComponent	OrderDetailItemsComponent	[AuthGuard]		data	order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L134-L140

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
10	AccountOrderDetailsOverviewComponent	OrderOverviewComponent	[AuthGuard]			order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L153-L156
11	AccountOrderDetailsPDFInvoicesComponent	InvoicesListComponent	[AuthGuard]			pdfInvoices	PDFInvoicesComponentsModule	@spartacus/pdf-invoices/components	feature-libs/pdf-invoices/components/pdf-invoices-components.module.ts:L45-L48
12	AccountOrderDetailsReorderComponent	OrderDetailReorderComponent	[AuthGuard]			order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L164-L167
13	AccountOrderDetailsSimpleOverviewComponent	OrderOverviewComponent	[AuthGuard]		data	order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L157-L163
14	AccountOrderDetailsTotalsComponent	OrderDetailTotalsComponent	[AuthGuard]			opfGiftCard, order	OrderDetailsModule	@spartacus/order/components	feature-libs/order/components/order-details/order-details.module.ts:L149-L152
15	AccountOrderDetailsTotalsComponent	OpfGiftCardOrderDetailTotalsComponent	[AuthGuard]			opfGiftCard, order	OpfGiftCardOrderDetailsModule	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/components/opf-gift-card-order-details/opf-gift-card-order-details.module.ts:L25-L28
16	AccountOrderDocumentFlowComponent	OrderDocumentFlowComponent	[AuthGuard]			orderDocumentFlow	OrderDocumentFlowComponentsModule	@spartacus/order/document-flow/components	feature-libs/order/document-flow/components/order-document-flow-components.module.ts:L51-L54
17	AccountOrderHistoryComponent	MyAccountV2OrderHistoryComponent				order	myAccountV2CmsMapping	@spartacus/order/components	feature-libs/order/components/order-history/order-history.module.ts:L36-L39
18	AccountOrderHistoryComponent	OrderHistoryComponent	[AuthGuard]			order	OrderHistoryModule	@spartacus/order/components	feature-libs/order/components/order-history/order-history.module.ts:L68-L71
19	AccountPaymentDetailsComponent	PaymentMethodsComponent	[AuthGuard]			opfTokenisation	PaymentMethodsModule	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/payment-methods/payment-methods.module.ts:L34-L37

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
20	AccountPaymentDetailsComponent	OpfTokenisationAccountPaymentMethodsComponent	[AuthGuard]			opfTokenisation	OpfTokenisationAccountPaymentMethodsModule	@spartacus/opf/tokenisation/root	integration-libs/opf/tokenisation/root/components/opf-tokenisation-account-payment-methods/opf-tokenisation-account-payment-methods.module.ts:L36-L39
21	AccountReplenishmentHistoryComponent	ReplenishmentOrderHistoryComponent	[AuthGuard]			order	ReplenishmentOrderHistoryModule	@spartacus/order/components	feature-libs/order/components/replenishment-order-history/replenishment-order-history.module.ts:L37-L40
22	AccountSavedCartHistoryComponent	SavedCartListComponent	[AuthGuard]			cartSavedCart	SavedCartListModule	@spartacus/cart/saved-cart/components	feature-libs/cart/saved-cart/components/list/saved-cart-list.module.ts:L35-L38
23	AccountSummaryDocumentComponent	AccountSummaryDocumentComponent	[AuthGuard, AdminGuard]			organizationAccountSummary	accountSummaryDocumentCmsConfig	@spartacus/organization/account-summary/components	feature-libs/organization/account-summary/components/details/document/account-summary-document.module.ts:L27-L30
24	AccountSummaryHeaderComponent	AccountSummaryHeaderComponent	[AuthGuard, AdminGuard]			organizationAccountSummary	accountSummaryHeaderCmsConfig	@spartacus/organization/account-summary/components	feature-libs/organization/account-summary/components/details/header/account-summary-header.module.ts:L21-L24
25	AddToSavedCartsComponent	AddToSavedCartComponent				cartSavedCart	AddToSavedCartModule	@spartacus/cart/saved-cart/components	feature-libs/cart/saved-cart/components/add-to-saved-cart/add-to-saved-cart.module.ts:L31-L33
26	AddToWishListComponent	AddToWishListComponent				addToWishList	AddToWishListModule	@spartacus/cart/wish-list/components/add-to-wishlist	feature-libs/cart/wish-list/components/add-to-wishlist/add-to-wish-list.module.ts:L39-L41
27	AnonymousConsentManagementBannerComponent	AnonymousConsentManagementBannerComponent			deferLoading		AnonymousConsentManagementBannerModule	@spartacus/storefront	core-libs/storefront/cms-components/anonymous-consent-management/anonymous-consent-management.module.ts:L34-L37
28	AnonymousConsentOpenDialogComponent	AnonymousConsentOpenDialogComponent					AnonymousConsentManagementBannerModule	@spartacus/storefront	core-libs/storefront/cms-components/anonymous-consent-management/anonymous-consent-management.module.ts:L38-L40
29	BannerComponent	BannerComponent					BannerModule	@spartacus/storefront	core-libs/storefront/cms-components/content/banner/banner.module.ts:L36-L38

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
30	Breadcrumb Component	Breadcrumb Component					Breadcrumb Module	@spartacus/storefront	core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.module.ts:L31-L33
31	Bulk Pricing Table Component	BulkPricing Table Component				productBulkPricing	BulkPricing TableModule	@spartacus/product/bulk-pricing/components	feature-libs/product/bulk-pricing/components/bulk-pricing-table/bulk-pricing-table.module.ts:L21-L23
32	Cancel Order Component	CancelOrder Component	[AuthGuard]	OrderAmendService → useExisting: Order CancellationService		order	CancelOrder Module	@spartacus/order/components	feature-libs/order/components/amend-order/cancellations/cancel-order/cancel-order.module.ts:L40-L49
33	Cancel Order Confirmation Component	CancelOrder Confirmation Component	[AuthGuard, Order CancellationGuard]	OrderAmendService → useExisting: Order CancellationService		order	CancelOrder Confirmation Module	@spartacus/order/components	feature-libs/order/components/amend-order/cancellations/cancel-order-confirmation/cancel-order-confirmation.module.ts:L29-L38
34	Cancel Service Order	CancelService Order Component	[AuthGuard, Cancel Service OrderGuard]			order	CancelService OrderModule	@spartacus/s4-service/order	integration-libs/s4-service/order/components/cancel-service-order/cancel-service-order.module.ts:L36-L39
35	Cancel Service Order Headline	CancelService OrderHeadline Component	[AuthGuard]			order	CancelService OrderHeadline Module	@spartacus/s4-service/order	integration-libs/s4-service/order/components/cancel-service-order-headline/cancel-service-order-headline.module.ts:L40-L43
36	Cart Apply Coupon Component	CartCoupon Component				cartBase	CartCoupon Module	@spartacus/cart/base/components	feature-libs/cart/base/components/cart-coupon/cart-coupon.module.ts:L43-L45
37	Cart Component	CartDetails Component				cartBase	CartDetails Module	@spartacus/cart/base/components	feature-libs/cart/base/components/cart-details/cart-details.module.ts:L37-L39
38	Cart Proceed To Checkout Component	CartProceedTo Checkout Component				cartBase	CartProceedTo CheckoutModule	@spartacus/cart/base/components	feature-libs/cart/base/components/cart-proceed-to-checkout/cart-proceed-to-checkout.module.ts:L31-L33
39	Cart Quick Order Form Component	CartQuick OrderForm Component				cartQuick Order	CartQuickOrder FormModule	@spartacus/cart/quick-order/components	feature-libs/cart/quick-order/components/cart-quick-order-form/cart-quick-order-form.module.ts:L37-L39
40	Cart Totals Component	CartTotals Component				cartBase	CartTotals Module	@spartacus/cart/base/components	feature-libs/cart/base/components/cart-totals/cart-totals.module.ts:L18-L20

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
41	CategoryNavigationComponent	CategoryNavigationComponent			data		CategoryNavigationModule	@spartacus/storefront	core-libs/storefront/cms-components/navigation/category-navigation/category-navigation.module.ts:L23-L28
42	CheckoutCostCenterComponent	CheckoutCostCenterComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CartValidationGuard]			checkout	CheckoutCostCenterModule	@spartacus/checkout/b2b/components	feature-libs/checkout/b2b/components/checkout-cost-center/checkout-cost-center.module.ts:L23-L26
43	CheckoutDeliveryAddress	B2BCheckoutDeliveryAddressComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CartValidationGuard]			checkout	B2BCheckoutDeliveryAddressModule	@spartacus/checkout/b2b/components	feature-libs/checkout/b2b/components/checkout-delivery-address/checkout-delivery-address.module.ts:L39-L42
44	CheckoutDeliveryAddress	CheckoutDeliveryAddressComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CartValidationGuard]			checkout	CheckoutDeliveryAddressModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.module.ts:L37-L40
45	CheckoutDeliveryMode	CheckoutDeliveryModeComponent	[CheckoutAuthGuard, CartNotEmptyGuard]		data	checkout	CheckoutDeliveryModeModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.module.ts:L39-L47
46	CheckoutDeliveryMode	ServiceCheckoutDeliveryModeComponent				checkout	ServiceCheckoutDeliveryModeModule	@spartacus/s4-service/checkout	integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.module.ts:L37-L39
47	CheckoutOrchestrator	CheckoutOrchestratorComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CheckoutGuard]			checkout	CheckoutOrchestratorModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-orchestrator/checkout-orchestrator.module.ts:L20-L23
48	CheckoutOrderSummary	CheckoutOrderSummaryComponent				checkout, opfGiftCard	CheckoutOrderSummaryModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-order-summary/checkout-order-summary.module.ts:L18-L20
49	CheckoutOrderSummary	OpfGiftCardCheckoutOrderSummaryComponent				checkout, opfGiftCard	OpfGiftCardCheckoutModule	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/components/opf-gift-card-checkout/opf-gift-card-checkout.module.ts:L37-L39
50	CheckoutPaymentDetails	CheckoutPaymentMethodComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutPaymentMethodModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-method.module.ts:L36-L39

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
51	CheckoutPaymentDetails	DpPaymentMethodComponent				checkout	DpPaymentMethodModule	@spartacus/digital-payments	integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L34-L36
52	CheckoutPaymentDetails	OpfTokenisationPaymentMethodComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	OpfTokenisationPaymentMethodModule	@spartacus/opf/tokenisation/root	integration-libs/opf/tokenisation/root/components/opf-tokenisation-payment-method/opf-tokenisation-payment-method.module.ts:L40-L43
53	CheckoutPaymentType	CheckoutPaymentTypeComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CartValidationGuard]			checkout	CheckoutPaymentTypeModule	@spartacus/checkout/b2b/components	feature-libs/checkout/b2b/components/checkout-payment-type/checkout-payment-type.module.ts:L30-L33
54	CheckoutPlaceOrder	CheckoutPlaceOrderComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutPlaceOrderModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-place-order/checkout-place-order.module.ts:L47-L50
55	CheckoutPlaceOrder	CheckoutScheduledReplenishmentPlaceOrderComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutScheduledReplenishmentPlaceOrderModule	@spartacus/checkout/scheduled-replenishment/components	feature-libs/checkout/scheduled-replenishment/components/checkout-place-order/checkout-place-order.module.ts:L37-L40
56	CheckoutProgress	CheckoutProgressComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CheckoutStepsSetGuard]			checkout	CheckoutProgressModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-progress/checkout-progress.module.ts:L35-L38
57	CheckoutProgressMobileBottom	CheckoutProgressMobileBottomComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CheckoutStepsSetGuard]			checkout	CheckoutProgressMobileBottomModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-progress/checkout-progress-mobile-bottom/checkout-progress-mobile-bottom.module.ts:L32-L35
58	CheckoutProgressMobileTop	CheckoutProgressMobileTopComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CheckoutStepsSetGuard]			checkout	CheckoutProgressMobileTopModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-progress/checkout-progress-mobile-top/checkout-progress-mobile-top.module.ts:L32-L35
59	CheckoutReviewOrder	B2BCheckoutReviewSubmitComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	B2BCheckoutReviewSubmitModule	@spartacus/checkout/b2b/components	feature-libs/checkout/b2b/components/checkout-review-submit/checkout-review-submit.module.ts:L43-L46
60	CheckoutReviewOrder	CheckoutReviewSubmitComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutReviewSubmitModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-review-submit/checkout-review-submit.module.ts:L41-L44

#	TYPECODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
61	CheckoutReviewOrder	ServiceCheckoutReviewSubmitComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	ServiceCheckoutReviewSubmitModule	@spartacus/s4-service/checkout	integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.module.ts:L43-L46
62	CheckoutReviewOverview	CheckoutReviewOverviewComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutReviewOverviewModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-review/checkout-review-overview/checkout-review-overview.module.ts:L25-L28
63	CheckoutReviewPayment	CheckoutReviewPaymentComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutReviewPaymentModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-review/checkout-review-payment/checkout-review-payment.module.ts:L35-L38
64	CheckoutReviewPickup	PickUpItemsDetailsComponent			data	pickupInStore	PickUpItemsDetailsModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/pickup-items-details/pickup-items-details.module.ts:L42-L48
65	CheckoutReviewShipping	CheckoutReviewShippingComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutReviewShippingModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-review/checkout-review-shipping/checkout-review-shipping.module.ts:L36-L39
66	CheckoutScheduleReplenishmentOrder	CheckoutScheduleReplenishmentOrderComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutScheduleReplenishmentOrderModule	@spartacus/checkout/scheduled-replenishment/components	feature-libs/checkout/scheduled-replenishment/components/checkout-schedule-replenishment-order/checkout-schedule-replenishment-order.module.ts:L29-L32
67	CheckoutServiceDetails	CheckoutServiceDetailsComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			checkout	CheckoutServiceDetailsModule	@spartacus/s4-service/checkout	integration-libs/s4-service/checkout/components/checkout-service-details/checkout-service-details.module.ts:L35-L38
68	ClaimDialogComponent	ClaimDialogComponent	[AuthGuard]				MyCouponsModule	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/my-coupons/my-coupons.module.ts:L76-L79
69	ClearCartComponent	ClearCartComponent				cartBase	ClearCartModule	@spartacus/cart/base/components	feature-libs/cart/base/components/clear-cart/clear-cart-button/clear-cart.module.ts:L25-L27
70	CloseAccountComponent	CloseAccountComponent	[AuthGuard]			userProfile	CloseAccountModule	@spartacus/user/profile/components	feature-libs/user/profile/components/close-account/close-account.module.ts:L43-L46

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
71	CMSLinkComponent	LinkComponent					LinkModule	@spartacus/storefront	core-libs/storefront/cms-components/content/link/link.module.ts:L19-L19
72	CMSParagraphComponent	ParagraphComponent					CmsParagraphModule	@spartacus/storefront	core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L24-L26
73	CMSProductListComponent	ProductListComponent			data	productMultiDimensionalList	ProductListModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-list/product-list.module.ts:L68-L75
74	CMSSiteContextComponent	SiteContextSelectorComponent		SiteContextComponentService → useClass: SiteContextComponentService			SiteContextSelectorModule	@spartacus/storefront	core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.module.ts:L36-L45
75	CMSTabParagraphComponent	ParagraphComponent					CmsParagraphModule	@spartacus/storefront	core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L27-L29
76	CMSTabParagraphContainer	TabParagraphContainerComponent					TabParagraphContainerModule	@spartacus/storefront	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.module.ts:L33-L35
77	ConfiguratorAddToCartButton	ConfiguratorAddToCartButtonComponent				productConfiguratorRulebased	ConfiguratorAddToCartButtonModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/add-to-cart-button/configurator-add-to-cart-button.module.ts:L24-L26
78	ConfiguratorExitButton	ConfiguratorExitButtonComponent				productConfiguratorRulebased	ConfiguratorExitButtonModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/exit-button/configurator-exit-button.module.ts:L17-L19
79	ConfiguratorForm	ConfiguratorFormComponent				productConfiguratorRulebased	ConfiguratorFormModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/form/configurator-form.module.ts:L25-L27
80	ConfiguratorFormGroup	ConfiguratorGroupComponent				productConfiguratorRulebased	ConfiguratorGroupModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/group/configurator-group.module.ts:L64-L66
81	ConfiguratorGroupTitle	ConfiguratorGroupTitleComponent				productConfiguratorRulebased	ConfiguratorGroupTitleModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/group-title/configurator-group-title.module.ts:L18-L20
82	ConfiguratorMenu	ConfiguratorGroupMenuComponent				productConfiguratorRulebased	ConfiguratorGroupMenuModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/group-menu/configurator-group-menu.module.ts:L24-L26

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
83	ConfiguratorOverviewBanner	ConfiguratorOverviewNotificationBannerComponent				productConfiguratorRulebased	ConfiguratorOverviewNotificationBannerModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/overview-notification-banner/configurator-overview-notification-banner.module.ts:L26-L28
84	ConfiguratorOverviewFilterButton	ConfiguratorOverviewFilterButtonComponent				productConfiguratorRulebased	ConfiguratorOverviewFilterButtonModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/overview-filter-button/configurator-overview-filter-button.module.ts:L23-L25
85	ConfiguratorOverviewForm	ConfiguratorOverviewFormComponent				productConfiguratorRulebased	ConfiguratorOverviewFormModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/overview-form/configurator-overview-form.module.ts:L30-L32
86	ConfiguratorOverviewSidebar	ConfiguratorOverviewSidebarComponent				productConfiguratorRulebased	ConfiguratorOverviewSidebarModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/overview-sidebar/configurator-overview-sidebar.module.ts:L25-L27
87	ConfiguratorPrevNext	ConfiguratorPreviousNextButtonsComponent				productConfiguratorRulebased	ConfiguratorPreviousNextButtonsModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/previous-next-buttons/configurator-previous-next-buttons.module.ts:L23-L25
88	ConfiguratorPriceSummary	ConfiguratorPriceSummaryComponent				productConfiguratorRulebased	ConfiguratorPriceSummaryModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/price-summary/configurator-price-summary.module.ts:L24-L26
89	ConfiguratorProductTitle	ConfiguratorProductTitleComponent				productConfiguratorRulebased	ConfiguratorProductTitleModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/product-title/configurator-product-title.module.ts:L33-L35
90	ConfiguratorTabBar	ConfiguratorTabBarComponent				productConfiguratorRulebased	ConfiguratorTabBarModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/tab-bar/configurator-tab-bar.module.ts:L31-L33
91	ConfiguratorUpdateMessage	ConfiguratorUpdateMessageComponent				productConfiguratorRulebased	ConfiguratorUpdateMessageModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/update-message/configurator-update-message.module.ts:L25-L27
92	ConfiguratorVariantCarousel	ConfiguratorVariantCarouselComponent				productConfiguratorRulebased	ConfiguratorVariantCarouselModule	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/variant-carousel/configurator-variant-carousel.module.ts:L27-L29

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
93	ConfigureProductComponent	Configure Product Component					Configure ProductModule	@spartacus/product-configurator/common	feature-libs/product-configurator/common/components/configure-product/configure-product.module.ts:L26-L28
94	Consent Management Component	MyAccountV2 Consent Management Component					myAccountV2CmsMapping	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/consent-management/consent-management.module.ts:L28-L31
95	Consent Management Component	Consent Management Component	[AuthGuard]				Consent Management Module	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/consent-management/consent-management.module.ts:L51-L54
96	Coupon Claim Component	CouponClaim Component	[AuthGuard]				MyCoupons Module	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/my-coupons/my-coupons.module.ts:L72-L75
97	Cpq Configurator ConflictAndError Messages Component	Configurator ConflictAnd ErrorMessages Component				productConfiguratorRulebased	Configurator ConflictAnd ErrorMessages Module	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/components/configurator-conflict-and-error-messages/configurator-conflict-and-error-messages.module.ts:L26-L28
98	Export Order Entries Component	ExportOrder Entries Component				cartImportExport	ExportOrder EntriesModule	@spartacus/cart/import-export/components	feature-libs/cart/import-export/components/export-entries/export-order-entries.module.ts:L26-L28
99	Footer Navigation Component	Footer Navigation Component					Footer Navigation Module	@spartacus/storefront	core-libs/storefront/cms-components/navigation/footer-navigation/footer-navigation.module.ts:L27-L29
100	Forgot Password Component	Forgot Password Component	[NotAuth Guard]	ForgotPassword ComponentService → use Class: ForgotPassword ComponentService		userProfile	ForgotPassword Module	@spartacus/user/profile/components	feature-libs/user/profile/components/forgot-password/forgot-password.module.ts:L51-L66
101	Forgot Password Component	Forgot Password Component		ForgotPassword ComponentService → use Class: CDCForgot PasswordComponent Service		userProfile	CDCForgot PasswordModule	@spartacus/cdc/user-profile	integration-libs/cdc/user-profile/forgot-password/cdc-forgot-password.module.ts:L37-L45
102	Future Stock Component	FutureStock Accordion Component				productFutureStock	FutureStock Accordion Module	@spartacus/product/future-stock/components	feature-libs/product/future-stock/components/future-stock-accordion/future-stock-accordion.module.ts:L23-L25

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
103	GigyaRaasComponent	GigyaRaasComponent	[GigyaRaasGuard]			cdc	GigyaRaasModule	@spartacus/cdc/components	integration-libs/cdc/components/gigya-raas/gigya-raas.module.ts:L20-L23
104	GuestCheckoutLoginComponent	CheckoutLoginComponent	[NotCheckoutAuthGuard]			checkout	CheckoutLoginModule	@spartacus/checkout/base/components	feature-libs/checkout/base/components/checkout-login/checkout-login.module.ts:L39-L42
105	HamburgerMenuComponent	HamburgerMenuComponent					HamburgerMenuModule	@spartacus/storefront	core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.module.ts:L17-L19
106	ImportExportOrderEntriesComponent	ImportExportOrderEntriesComponent				cartImportExport	ImportExportOrderEntriesModule	@spartacus/cart/import-export/components	feature-libs/cart/import-export/components/import-export/import-export-order-entries.module.ts:L25-L27
107	ImportOrderEntriesComponent	ImportOrderEntriesComponent				cartImportExport	ImportOrderEntriesModule	@spartacus/cart/import-export/components	feature-libs/cart/import-export/components/import-to-cart/import-order-entries.module.ts:L49-L51
108	LanguageCurrencyComponent	LanguageCurrencyComponent					SiteContextSelectorModule	@spartacus/storefront	core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.module.ts:L46-L48
109	LoginComponent	LoginComponent				userAccount	LoginModule	@spartacus/user/account/components	feature-libs/user/account/components/login/login.module.ts:L32-L34
110	ManageAccountSummaryListComponent	AccountSummaryListComponent	[AuthGuard, AdminGuard]	ListService → useExisting: AccountSummaryUnitListService; ItemService → useExisting: AccountSummaryItemService	childRoutes	organizationAccountSummary	accountSummaryListCmsConfig	@spartacus/organization/account-summary/components	feature-libs/organization/account-summary/components/list/account-summary-list.config.ts:L49-L82
111	ManageBudgetsListComponent	ListComponent	[AuthGuard, AdminGuard]	ListService → useExisting: BudgetListService; ItemService → useExisting: BudgetItemService	childRoutes	organizationAdministration	budgetCmsConfig	@spartacus/organization/administration/components	feature-libs/organization/administration/components/budget/budget.config.ts:L31-L79
112	ManageCostCentersListComponent	ListComponent	[AuthGuard, AdminGuard]	ListService → useExisting: CostCenterListService; ItemService → useExisting: CostCenterItemService	childRoutes	organizationAdministration	costCenterCmsConfig	@spartacus/organization/administration/components	feature-libs/organization/administration/components/cost-center/cost-center.config.ts:L32-L92
113	ManagePermissionsListComponent	ListComponent	[AuthGuard, AdminGuard]	ListService → useExisting: PermissionListService; ItemService → useExisting: PermissionItemService	childRoutes	organizationAdministration	permissionCmsConfig	@spartacus/organization/administration/components	feature-libs/organization/administration/components/permission/permission.config.ts:L27-L71

#	TYPECODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
114	Manage Units List Component	UnitList Component	[AuthGuard, AdminGuard]	ListService → use Existing: UnitList Service; ItemService → useExisting: UnitItem Service	childRoutes	organizationAdministration	unitsCmsConfig	@spartacus/organization/administration/components	feature-libs/organization/administration/components/unit/units.config.ts:L53-L193
115	Manage Units List Component			B2BUserService → use Existing: CdcB2BUser Service; OrgUnit Service → useClass: CdcOrgUnitService; unitsCmsConfig.cms Components?.Manage UnitsListComponent?.pro…		organizationAdministration	Cdc Administration Module	@spartacus/cdc/organization/administration	integration-libs/cdc/organization/administration/cdc-administration.module.ts:L46-L56
116	Manage User Groups List Component	ListComponent	[AuthGuard, AdminGuard]	ListService → use Existing: UserGroup ListService; Item Service → useExisting: UserGroupItemService	childRoutes	organizationAdministration	userGroupCmsConfig	@spartacus/organization/administration/components	feature-libs/organization/administration/components/user-group/user-group.config.ts:L34-L110
117	Manage Users List Component	ListComponent	[AuthGuard, AdminGuard]	ListService → use Existing: UserList Service; ItemService → useExisting: UserItem Service	childRoutes	organizationAdministration	userCmsConfig	@spartacus/organization/administration/components	feature-libs/organization/administration/components/user/user.config.ts:L43-L140
118	Manage Users List Component			UserListService → use Existing: CdcUserList Service; B2BUser Service → useClass: CdcB2BUserService; userCmsConfig.cms Components?.Manage UsersListComponent?.prov…		organizationAdministration	Cdc Administration Module	@spartacus/cdc/organization/administration	integration-libs/cdc/organization/administration/cdc-administration.module.ts:L28-L38
119	Merchandising Carousel Component	Merchandising Carousel Component					MerchandisingCarouselCmsModule	@spartacus/cds	integration-libs/cds/src/merchandising/cms-components/merchandising-carousel-cms.module.ts:L28-L30
120	MiniCart Component	MiniCart Component				miniCart	MiniCartModule	@spartacus/cart/base/components/mini-cart	feature-libs/cart/base/components/mini-cart/mini-cart.module.ts:L31-L33
121	My Account Side Navigation Component	MyAccountV2 Navigation Component	[AuthGuard]				MyAccountV2 NavigationModule	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/my-account-v2/my-account-v2-navigation/my-account-v2-navigation.module.ts:L22-L25
122	My Account View Order Component	MyAccountV2 Orders Component	[AuthGuard]			order	MyAccountV2 OrdersModule	@spartacus/order/components	feature-libs/order/components/my-account-v2/my-account-v2-orders.module.ts:L36-L39

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
123	MyAccountViewRequestsComponent	MyAccountV2CustomerTicketingComponent	[AuthGuard]			customerTicketing	MyAccountV2CustomerTicketingModule	@spartacus/customer-ticketing/components	feature-libs/customer-ticketing/components/my-account-v2/my-account-v2-customer-ticketing.module.ts:L25-L28
124	MyAccountViewUserComponent	MyAccountV2UserComponent	[AuthGuard]			userAccount	MyAccountV2UserModule	@spartacus/user/account/components	feature-libs/user/account/components/my-account-v2-user/my-account-v2-user.module.ts:L23-L26
125	MyCouponsComponent	MyCouponsComponent	[AuthGuard]				MyCouponsModule	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/my-coupons/my-coupons.module.ts:L68-L71
126	MyInterestsComponent	MyInterestsComponent	[AuthGuard]				MyInterestsModule	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/my-interests/my-interests.module.ts:L51-L54
127	MyPreferredStoreComponent	MyPreferredStoreComponent				pickupInStore	MyPreferredStoreModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/my-preferred-store/my-preferred-store.module.ts:L22-L24
128	NavigationComponent	NavigationComponent					NavigationModule	@spartacus/storefront	core-libs/storefront/cms-components/navigation/navigation/navigation.module.ts:L35-L37
129	NotificationPreferenceComponent	MyAccountV2NotificationPreferenceComponent					myAccountV2CmsMapping	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/notification-preference/notification-preference.module.ts:L27-L30
130	NotificationPreferenceComponent	NotificationPreferenceComponent	[AuthGuard]				NotificationPreferenceModule	@spartacus/storefront	core-libs/storefront/cms-components/myaccount/notification-preference/notification-preference.module.ts:L53-L56
131	OAuthCallbackComponent	SpinnerComponent	[OAuthCallbackGuard]			userAccount	OAuthCallbackModule	@spartacus/user/account/components	feature-libs/user/account/components/oauth-callback/oauth-callback.module.ts:L31-L34
132	OpfB2bCheckoutPaymentAndReview	OpfB2bCheckoutPaymentAndReviewComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			opfB2bCheckout	OpfB2bCheckoutPaymentAndReviewModule	@spartacus/opf/b2b-checkout/components	integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-payment-and-review/opf-b2b-checkout-payment-and-review.module.ts:L61-L64

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
133	OpfCheckoutDeliveryAddress	OpfB2bCheckoutDeliveryAddressComponent				opfB2bCheckout	OpfB2bCheckoutDeliveryAddressModule	@spartacus/opf/b2b-checkout/components	integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-delivery-address/opf-b2b-checkout-delivery-address.module.ts:L35-L37
134	OpfCheckoutEmailUpdateComponent	OpfCheckoutEmailUpdateComponent				opfCheckout	OpfCheckoutEmailUpdateModule	@spartacus/opf/checkout/components	integration-libs/opf/checkout/components/opf-checkout-email-update/opf-checkout-email-update.module.ts:L26-L28
135	OpfCheckoutPaymentAndReview	OpfCheckoutPaymentAndReviewComponent	[CheckoutAuthGuard, CartNotEmptyGuard]			opfCheckout	OpfCheckoutPaymentAndReviewModule	@spartacus/opf/checkout/components	integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.module.ts:L60-L63
136	OpfCheckoutPaymentType	OpfB2bCheckoutPaymentTypeComponent	[CheckoutAuthGuard, CartNotEmptyGuard, CartValidationGuard]			opfB2bCheckout	OpfB2bCheckoutPaymentTypeModule	@spartacus/opf/b2b-checkout/components	integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-payment-type/opf-b2b-checkout-payment-type.module.ts:L38-L41
137	OpfCheckoutReview	OpfB2bCheckoutReviewComponent				opfB2bCheckout	OpfB2bCheckoutReviewModule	@spartacus/opf/b2b-checkout/components	integration-libs/opf/b2b-checkout/components/opf-b2b-checkout-review/opf-b2b-checkout-review.module.ts:L49-L51
138	OpfCtaQuickBuyButtons	OpfCtaQuickBuyButtonsComponent				opfCta	OpfCtaQuickBuyButtonsModule	@spartacus/opf/cta/components	integration-libs/opf/cta/components/opf-cta-quick-buy-buttons/opf-cta-quick-buy-buttons.module.ts:L25-L27
139	OpfCtaScriptsComponent	OpfCtaScriptsComponent				opfCta	OpfCtaScriptsModule	@spartacus/opf/cta/components	integration-libs/opf/cta/components/opf-cta-scripts/opf-cta-scripts.module.ts:L26-L28
140	OpfQuickBuyButtonsComponent	OpfQuickBuyButtonsComponent				opfQuickBuy	OpfQuickBuyButtonsModule	@spartacus/opf/quick-buy/components	integration-libs/opf/quick-buy/components/opf-quick-buy-buttons/opf-quick-buy-buttons.module.ts:L20-L22
141	OrderApprovalDetailApprovalDetailsComponent	OrderDetailPermissionResultsComponent	[AuthGuard, ApproverGuard]	OrderDetailsService → useExisting: OrderApprovalDetailService		organizationOrderApproval	OrderApprovalDetailsModule	@spartacus/organization/order-approval	feature-libs/organization/order-approval/components/details/order-approval-details.module.ts:L57-L66
142	OrderApprovalDetailFormComponent	OrderApprovalDetailFormComponent	[AuthGuard, ApproverGuard]			organizationOrderApproval	OrderApprovalDetailsModule	@spartacus/organization/order-approval	feature-libs/organization/order-approval/components/details/order-approval-details.module.ts:L90-L93

#	TYPECODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
143	OrderApprovalDetailItemsComponent	OrderDetailItemsComponent	[AuthGuard, ApproverGuard]	OrderDetailsService → useExisting: OrderApprovalDetailService		organizationOrderApproval	OrderApprovalDetailsModule	@spartacus/organization/order-approval	feature-libs/organization/order-approval/components/details/order-approval-details.module.ts:L80-L89
144	OrderApprovalDetailShippingComponent	OrderOverviewComponent	[AuthGuard, ApproverGuard]	OrderDetailsService → useExisting: OrderApprovalDetailService		organizationOrderApproval	OrderApprovalDetailsModule	@spartacus/organization/order-approval	feature-libs/organization/order-approval/components/details/order-approval-details.module.ts:L70-L79
145	OrderApprovalDetailTotalsComponent	OrderDetailTotalsComponent	[AuthGuard, ApproverGuard]	OrderDetailsService → useExisting: OrderApprovalDetailService		organizationOrderApproval	OrderApprovalDetailsModule	@spartacus/organization/order-approval	feature-libs/organization/order-approval/components/details/order-approval-details.module.ts:L47-L56
146	OrderApprovalListComponent	OrderApprovalListComponent	[AuthGuard, ApproverGuard]			organizationOrderApproval	OrderApprovalListModule	@spartacus/organization/order-approval	feature-libs/organization/order-approval/components/list/order-approval-list.module.ts:L27-L30
147	OrderConfirmationBillingComponent	OrderDetailBillingComponent	[OrderConfirmationGuard]	OrderDetailsService → useExisting: OrderFacade		order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L120-L129
148	OrderConfirmationItemsComponent	OrderConfirmationItemsComponent	[OrderConfirmationGuard]			order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L80-L83
149	OrderConfirmationOverviewComponent	OrderOverviewComponent	[OrderConfirmationGuard]	OrderDetailsService → useExisting: OrderFacade		order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L96-L105
150	OrderConfirmationPickUpComponent	PickUpItemsDetailsComponent			data	pickupInStore	PickUpItemsDetailsModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/pickup-items-details/pickup-items-details.module.ts:L35-L41
151	OrderConfirmationShippingComponent	OrderConfirmationShippingComponent	[OrderConfirmationGuard]			order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L116-L119
152	OrderConfirmationThankMessageComponent	OrderConfirmationThankYouMessageComponent	[OrderConfirmationGuard]			order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L72-L75

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
153	OrderConfirmationTotalsComponent	OrderConfirmationTotalsComponent	[OrderConfirmationGuard]			opfGiftCard, order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L88-L91
154	OrderConfirmationTotalsComponent	OpfGiftCardOrderConfirmationTotalsComponent	[OrderConfirmationGuard]			opfGiftCard, order	OpfGiftCardOrderConfirmationModule	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/components/opf-gift-card-order-confirmation/opf-gift-card-order-confirmation.module.ts:L17-L20
155	OrderReturnRequestListComponent	OrderReturnRequestListComponent	[AuthGuard]			order	ReturnRequestListModule	@spartacus/order/components	feature-libs/order/components/return-request-list/order-return-request-list.module.ts:L34-L37
156	OrganizationUserRegistrationComponent	UserRegistrationFormComponent	[NotAuthGuard]			organizationUserRegistration	UserRegistrationFormModule	@spartacus/organization/user-registration/components	feature-libs/organization/user-registration/components/form/user-registration-form.module.ts:L45-L48
157	PageTitleComponent	PageTitleComponent					PageTitleModule	@spartacus/storefront	core-libs/storefront/cms-components/navigation/page-header/page-title.module.ts:L22-L24
158	PDFDocumentComponent	PDFComponent					PDFModule	@spartacus/storefront	core-libs/storefront/cms-components/content/pdf/pdf.module.ts:L18-L20
159	PickupInStoreDeliveryModeComponent	PickUpItemsDetailsComponent			data	pickupInStore	PickUpItemsDetailsModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/pickup-items-details/pickup-items-details.module.ts:L49-L55
160	ProductAddToCartComponent	AddToCartComponent			data	addToCart	AddToCartModule	@spartacus/cart/base/components/add-to-cart	feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts:L37-L42
161	ProductCarouselComponent	ProductCarouselComponent					ProductCarouselModule	@spartacus/storefront	core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.module.ts:L45-L47
162	ProductDetailsTabComponent	ProductDetailsTabComponent					ProductDetailsTabModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-tabs/product-details-tab/product-details-tab.module.ts:L24-L26
163	ProductGridComponent	ProductListComponent			data	productMultiDimensionalList	ProductListModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-list/product-list.module.ts:L76-L83

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
164	ProductImagesComponent	ProductImagesComponent				productImageZoom	ProductImagesModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-images/product-images.module.ts:L32-L34
165	ProductImagesComponent	ProductImageZoomProductImagesComponent				productImageZoom	ProductImageZoomModule	@spartacus/product/image-zoom/components	feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom.module.ts:L55-L57
166	ProductIntroComponent	ProductIntroComponent					ProductIntroModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-intro/product-intro.module.ts:L29-L31
167	ProductMultiDimensionalSelectorComponent	ProductMultiDimensionalSelectorComponent	[ProductMultiDimensionalSelectorGuard]			productMultiDimensionalSelector	ProductMultiDimensionalSelectorComponentModule	@spartacus/product-multi-dimensional/selector/components	feature-libs/product-multi-dimensional/selector/components/selector/product-multi-dimensional-selector-component.module.ts:L43-L46
168	ProductReferencesComponent	ProductReferencesComponent					ProductReferencesModule	@spartacus/storefront	core-libs/storefront/cms-components/product/carousel/product-references/product-references.module.ts:L35-L37
169	ProductRefinementComponent	ProductFacetNavigationComponent					ProductFacetNavigationModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-list/product-facet-navigation/product-facet-navigation.module.ts:L23-L25
170	ProductReviewsTabComponent	ProductReviewsComponent					ProductReviewsModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-tabs/product-reviews/product-reviews.module.ts:L43-L45
171	ProductSpecsTabComponent	ProductAttributesComponent					ProductAttributesModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-tabs/product-attributes/product-attributes.module.ts:L18-L20
172	ProductSummaryComponent	ProductSummaryComponent					ProductSummaryModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-summary/product-summary.module.ts:L23-L25
173	ProductVariantSelectorComponent	ProductVariantsContainerComponent	[ProductVariantsGuard]			productVariants	ProductVariantsComponentsModule	@spartacus/product/variants/components	feature-libs/product/variants/components/product-variants-components.module.ts:L26-L29
174	ProfileTagComponent	ProfileTagComponent			deferLoading		ProfileTagCmsModule	@spartacus/cds	integration-libs/cds/src/profiletag/cms-components/profile-tag-cms.module.ts:L21-L24

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
175	PunchoutButtonsComponent	PunchoutButtonsComponent				punchout	PunchoutComponentsModule	@spartacus/punchout/components	integration-libs/punchout/components/punchout-components.module.ts:L51-L53
176	PunchoutCloseSessionComponent	PunchoutCloseSessionComponent				punchout	PunchoutComponentsModule	@spartacus/punchout/components	integration-libs/punchout/components/punchout-components.module.ts:L57-L59
177	PunchoutErrorComponent	PunchoutErrorComponent				punchout	PunchoutComponentsModule	@spartacus/punchout/components	integration-libs/punchout/components/punchout-components.module.ts:L48-L50
178	PunchoutInspectCartComponent	PunchoutInspectCartComponent				punchout	PunchoutComponentsModule	@spartacus/punchout/components	integration-libs/punchout/components/punchout-components.module.ts:L60-L62
179	PunchoutRequisitionComponent	PunchoutRequisitionComponent				punchout	PunchoutComponentsModule	@spartacus/punchout/components	integration-libs/punchout/components/punchout-components.module.ts:L54-L56
180	PunchoutSessionComponent	PunchoutSessionComponent				punchout	PunchoutComponentsModule	@spartacus/punchout/components	integration-libs/punchout/components/punchout-components.module.ts:L45-L47
181	QualtricsComponent	QualtricsComponent				qualtrics	QualtricsComponentsModule	@spartacus/qualtrics/components	feature-libs/qualtrics/components/qualtrics-components.module.ts:L26-L28
182	QualtricsEmbeddedFeedbackComponent	QualtricsEmbeddedFeedbackComponent					QualtricsComponentsModule	@spartacus/qualtrics/components	feature-libs/qualtrics/components/qualtrics-components.module.ts:L23-L25
183	QuickOrderComponent	QuickOrderComponent			data	cartQuickOrder	QuickOrderListModule	@spartacus/cart/quick-order/components	feature-libs/cart/quick-order/components/quick-order/quick-order-list.module.ts:L55-L60
184	QuoteCartGuardComponent	QuoteCartGuardComponent	[QuoteCartGuard]			quote_cart_guard	QuoteCartGuardComponentModule	@spartacus/quote/components/cart-guard	feature-libs/quote/components/cart-guard/quote-cart-guard.component.module.ts:L18-L21
185	QuoteCommentsComponent	QuoteCommentsComponent	[AuthGuard]			quote	QuoteCommentsModule	@spartacus/quote/components	feature-libs/quote/components/comments/quote-comments.module.ts:L29-L32
186	QuoteHeaderOverviewComponent	QuoteHeaderOverviewComponent	[AuthGuard]			quote	QuoteHeaderOverviewModule	@spartacus/quote/components	feature-libs/quote/components/header/overview/quote-header-overview.module.ts:L36-L39
187	QuoteItemsComponent	QuoteItemsComponent	[AuthGuard]			quote	QuoteItemsModule	@spartacus/quote/components	feature-libs/quote/components/items/quote-items.module.ts:L31-L34

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
188	QuoteLinksComponent	QuoteLinksComponent	[AuthGuard]			quote	QuoteLinksModule	@spartacus/quote/components	feature-libs/quote/components/links/quote-links.module.ts:L32-L35
189	QuoteRequestButtonComponent	QuoteRequestButtonComponent				quote_request	QuoteRequestButtonModule	@spartacus/quote/components/request-button	feature-libs/quote/components/request-button/quote-request-button.module.ts:L18-L20
190	QuoteSummaryComponent	QuoteSummaryComponent	[AuthGuard]			quote	QuoteSummaryModule	@spartacus/quote/components	feature-libs/quote/components/summary/quote-summary.module.ts:L32-L35
191	RegisterB2BCustomerWithOTPComponent	UserRegistrationOTPFormComponent	[NotAuthGuard]			organizationUserRegistration	UserRegistrationOTPFormModule	@spartacus/organization/user-registration/components	feature-libs/organization/user-registration/components/registration-otp-form/user-registration-otp-form.module.ts:L41-L44
192	RegisterCustomerComponent	RegisterComponent	[NotAuthGuard]	RegisterComponentService → useClass: RegisterComponentService		userProfile	RegisterComponentModule	@spartacus/user/profile/components	feature-libs/user/profile/components/register/register.module.ts:L55-L69
193	RegisterCustomerComponent			RegisterComponentService → useClass: CDCRegisterComponentService		userProfile	CDCRegisterModule	@spartacus/cdc/user-profile	integration-libs/cdc/user-profile/register/cdc-register.module.ts:L41-L48
194	RegisterCustomerWithOTPComponent	OneTimePasswordRegisterComponent	[NotAuthGuard]	RegisterComponentService → useClass: RegisterComponentService		userAccount, userProfile	OneTimePasswordRegisterModule	@spartacus/user/profile/components	feature-libs/user/profile/components/otp-login-register/otp-login-register.module.ts:L54-L64
195	ReplenishmentConfirmationItemsComponent	OrderConfirmationItemsComponent	[OrderConfirmationGuard]			order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L84-L87
196	ReplenishmentConfirmationMessageComponent	OrderConfirmationThankYouMessageComponent	[OrderConfirmationGuard]			order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L76-L79
197	ReplenishmentConfirmationOverviewComponent	OrderOverviewComponent	[OrderConfirmationGuard]	OrderDetailsService → useExisting: OrderFacade		order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L106-L115
198	ReplenishmentConfirmationTotalsComponent	OrderConfirmationTotalsComponent	[OrderConfirmationGuard]			order	OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L92-L95

#	TYPECODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
199	ReplenishmentDetailActionsComponent	ReplenishmentOrderCancellationComponent				order	ReplenishmentOrderDetailsModule	@spartacus/order/components	feature-libs/order/components/replenishment-order-details/replenishment-order-details.module.ts:L80-L82
200	ReplenishmentDetailItemsComponent	OrderDetailItemsComponent		OrderDetailsService → useExisting: ReplenishmentOrderDetailsService		order	ReplenishmentOrderDetailsModule	@spartacus/order/components	feature-libs/order/components/replenishment-order-details/replenishment-order-details.module.ts:L53-L61
201	ReplenishmentDetailOrderHistoryComponent	OrderHistoryComponent				order	ReplenishmentOrderDetailsModule	@spartacus/order/components	feature-libs/order/components/replenishment-order-details/replenishment-order-details.module.ts:L83-L85
202	ReplenishmentDetailShippingComponent	OrderOverviewComponent		OrderDetailsService → useExisting: ReplenishmentOrderDetailsService		order	ReplenishmentOrderDetailsModule	@spartacus/order/components	feature-libs/order/components/replenishment-order-details/replenishment-order-details.module.ts:L71-L79
203	ReplenishmentDetailTotalsComponent	OrderDetailTotalsComponent		OrderDetailsService → useExisting: ReplenishmentOrderDetailsService		order	ReplenishmentOrderDetailsModule	@spartacus/order/components	feature-libs/order/components/replenishment-order-details/replenishment-order-details.module.ts:L62-L70
204	RescheduleServiceOrder	RescheduleServiceOrderComponent	[AuthGuard, ServiceOrderGuard]			order	RescheduleServiceOrderModule	@spartacus/s4-service/order	integration-libs/s4-service/order/components/reschedule-service-order/reschedule-service-order.module.ts:L47-L50
205	ResetPasswordComponent	ResetPasswordComponent	[NotAuthGuard]	ResetPasswordComponentService → useClass: ResetPasswordComponentService		userProfile	ResetPasswordModule	@spartacus/user/profile/components	feature-libs/user/profile/components/reset-password/reset-password.module.ts:L49-L59
206	ReturningCustomerLoginComponent	LoginFormComponent	[NotAuthGuard, CustomLoginGuard]	LoginFormComponentService → useClass: LoginFormComponentService		userAccount	LoginFormModule	@spartacus/user/account/components	feature-libs/user/account/components/login-form/login-form.module.ts:L52-L62
207	ReturningCustomerLoginComponent	LoginFormComponent	[NotAuthGuard]	LoginFormComponentService → useClass: CdcLoginFormComponentService		userAccount	CDCLoginFormModule	@spartacus/cdc/user-account	integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L51-L66
208	ReturningCustomerOTPLoginComponent	OneTimePasswordLoginFormComponent	[NotAuthGuard]			userAccount	OneTimePasswordLoginFormModeule	@spartacus/user/account/components	feature-libs/user/account/components/otp-login-form/otp-login-form.module.ts:L47-L50

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
209	ReturningCustomerRegisterComponent	LoginRegisterComponent	[NotAuthGuard, LoginAsGuestGuard]			userAccount	LoginRegisterModule	@spartacus/user/account/components	feature-libs/user/account/components/login-register/login-register.module.ts:L35-L38
210	ReturningOrganizationUserRegisterComponent	LoginRegisterB2BComponent	[NotAuthGuard]			userAccount	LoginRegisterModule	@spartacus/user/account/components	feature-libs/user/account/components/login-register/login-register.module.ts:L43-L46
211	ReturnOrderComponent	ReturnOrderComponent	[AuthGuard]	OrderAmendService → useExisting: OrderReturnService		order	ReturnOrderModule	@spartacus/order/components	feature-libs/order/components/amend-order/returns/return-order/return-order.module.ts:L28-L37
212	ReturnOrderConfirmationComponent	ReturnOrderConfirmationComponent	[AuthGuard, OrderReturnGuard]	OrderAmendService → useExisting: OrderReturnService		order	ReturnOrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/amend-order/returns/return-order-confirmation/return-order-confirmation.module.ts:L35-L44
213	ReturnRequestItemsComponent	ReturnRequestItemsComponent				order	ReturnRequestDetailModule	@spartacus/order/components	feature-libs/order/components/return-request-detail/return-request-detail.module.ts:L42-L44
214	ReturnRequestOverviewComponent	ReturnRequestOverviewComponent				order	ReturnRequestDetailModule	@spartacus/order/components	feature-libs/order/components/return-request-detail/return-request-detail.module.ts:L39-L41
215	ReturnRequestTotalsComponent	ReturnRequestTotalsComponent				order	ReturnRequestDetailModule	@spartacus/order/components	feature-libs/order/components/return-request-detail/return-request-detail.module.ts:L45-L47
216	RotatingImagesComponent	BannerCarouselComponent					BannerCarouselModule	@spartacus/storefront	core-libs/storefront/cms-components/content/banner-carousel/banner-carousel.module.ts:L25-L27
217	SavedCartDetailsActionComponent	SavedCartDetailsActionComponent	[AuthGuard]			cartSavedCart	SavedCartDetailsModule	@spartacus/cart/saved-cart/components	feature-libs/cart/saved-cart/components/details/saved-cart-details.module.ts:L52-L55
218	SavedCartDetailsItemsComponent	SavedCartDetailsItemsComponent	[AuthGuard]			cartSavedCart	SavedCartDetailsModule	@spartacus/cart/saved-cart/components	feature-libs/cart/saved-cart/components/details/saved-cart-details.module.ts:L48-L51

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
219	SavedCartDetailsOverviewComponent	SavedCartDetailsOverviewComponent	[AuthGuard]			cartSavedCart	SavedCartDetailsModule	@spartacus/cart/saved-cart/components	feature-libs/cart/saved-cart/components/details/saved-cart-details.module.ts:L44-L47
220	SaveForLaterComponent	SaveForLaterComponent				cartBase	SaveForLaterModule	@spartacus/cart/base/components	feature-libs/cart/base/components/save-for-later/save-for-later.module.ts:L23-L25
221	ScrollToTopComponent	ScrollToTopComponent					ScrollToTopModule	@spartacus/storefront	core-libs/storefront/cms-components/navigation/scroll-to-top/scroll-to-top.module.ts:L18-L20
222	SearchBoxComponent	SearchBoxComponent					SearchBoxModule	@spartacus/storefront	core-libs/storefront/cms-components/navigation/search-box/search-box.module.ts:L43-L45
223	SearchResultsListComponent	ProductListComponent			data	productMultiDimensionalList	ProductListModule	@spartacus/storefront	core-libs/storefront/cms-components/product/product-list/product-list.module.ts:L84-L91
224	SimpleBannerComponent	BannerComponent					BannerModule	@spartacus/storefront	core-libs/storefront/cms-components/content/banner/banner.module.ts:L39-L41
225	SimpleResponsiveBannerComponent	BannerComponent					BannerModule	@spartacus/storefront	core-libs/storefront/cms-components/content/banner/banner.module.ts:L33-L35
226	SiteThemeSwitcherComponent	SiteThemeSwitcherComponent					SiteThemeSwitcherModule	@spartacus/storefront	core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.module.ts:L31-L33
227	StockNotificationComponent	StockNotificationComponent					StockNotificationModule	@spartacus/storefront	core-libs/storefront/cms-components/product/stock-notification/stock-notification.module.ts:L39-L41
228	StoreFinderComponent	StoreFinderComponent			childRoutes	stores	StoreFinderComponentsModule	@spartacus/storefinder/components	feature-libs/storefinder/components/store-finder-components.module.ts:L69-L97
229	SubscriptionBillDetailsComponent	SubscriptionBillingDetailsComponent	[AuthGuard]			subscriptionBilling	SubscriptionBillingDetailsModule	@spartacus/subscription-billing/components	feature-libs/subscription-billing/components/billing-details/subscription-billing-details.module.ts:L33-L36
230	SubscriptionBillsHistoryComponent	SubscriptionBillingListComponent	[AuthGuard]			subscriptionBilling	SubscriptionBillingListModule	@spartacus/subscription-billing/components	feature-libs/subscription-billing/components/list/billing/subscription-billing-list.module.ts:L47-L50

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
231	SubscriptionDetailsComponent	SubscriptionDetailsComponent	[AuthGuard]			subscriptionBilling	SubscriptionDetailsModule	@spartacus/subscription-billing/components	feature-libs/subscription-billing/components/details/subscription-details.module.ts:L33-L36
232	SubscriptionHistoryComponent	SubscriptionListComponent	[AuthGuard]			subscriptionBilling	SubscriptionListModule	@spartacus/subscription-billing/components	feature-libs/subscription-billing/components/list/subscription-list.module.ts:L34-L37
233	SubscriptionProductPriceComponent	SubscriptionProductPriceComponent				subscriptionBilling	SubscriptionProductPriceModule	@spartacus/subscription-billing/components	feature-libs/subscription-billing/components/product/price/subscription-product-price.module.ts:L24-L26
234	SupportTicketCloseComponent	CustomerTicketingCloseComponent	[AuthGuard]			customerTicketing	CustomerTicketingCloseModule	@spartacus/customer-ticketing/components	feature-libs/customer-ticketing/components/details/customer-ticketing-close/customer-ticketing-close.module.ts:L48-L51
235	SupportTicketDetailsComponent	CustomerTicketingDetailsComponent	[AuthGuard]			customerTicketing	CustomerTicketingDetailsModule	@spartacus/customer-ticketing/components	feature-libs/customer-ticketing/components/details/customer-ticketing-details/customer-ticketing-details.module.ts:L31-L34
236	SupportTicketHistoryComponent	CustomerTicketingListComponent	[AuthGuard]			customerTicketing	CustomerTicketingListModule	@spartacus/customer-ticketing/components	feature-libs/customer-ticketing/components/list/customer-ticketing-list/customer-ticketing-list.module.ts:L42-L45
237	SupportTicketReopenComponent	CustomerTicketingReopenComponent	[AuthGuard]			customerTicketing	CustomerTicketingReopenModule	@spartacus/customer-ticketing/components	feature-libs/customer-ticketing/components/details/customer-ticketing-reopen/customer-ticketing-reopen.module.ts:L44-L47
238	SupportTicketUpdateComponent	CustomerTicketingMessagesComponent	[AuthGuard]			customerTicketing	CustomerTicketingMessagesModule	@spartacus/customer-ticketing/components	feature-libs/customer-ticketing/components/details/customer-ticketing-messages/customer-ticketing-messages.module.ts:L35-L38
239	TextfieldConfigurationForm	ConfiguratorTextfieldFormComponent				productConfiguratorTextfield	TextfieldConfiguratorComponentsModule	@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/components/textfield-configurator-components.module.ts:L35-L37
240	UnitLevelOrderDetailsItemsComponent	OrderDetailItemsComponent	[AuthGuard, UnitLevelOrdersViewerGuard]	OrderDetailsService → useExisting: UnitLevelOrderDetailService		organizationUnitOrder	UnitLevelOrderDetailModule	@spartacus/organization/unit-order/components	feature-libs/organization/unit-order/components/unit-level-order-detail/unit-level-order-detail.module.ts:L31-L40

#	TYPECODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
241	UnitLevelOrderDetailsOverviewComponent	UnitLevelOrderOverviewComponent	[AuthGuard, UnitLevelOrdersViewerGuard]			organizationUnitOrder	UnitLevelOrderDetailModule	@spartacus/organization/unit-order/components	feature-libs/organization/unit-order/components/unit-level-order-detail/unit-level-order-detail.module.ts:L27-L30
242	UnitLevelOrderDetailsTotalsComponent	OrderDetailTotalsComponent	[AuthGuard, UnitLevelOrdersViewerGuard]	OrderDetailsService → useExisting: UnitLevelOrderDetailService		organizationUnitOrder	UnitLevelOrderDetailModule	@spartacus/organization/unit-order/components	feature-libs/organization/unit-order/components/unit-level-order-detail/unit-level-order-detail.module.ts:L41-L50
243	UnitLevelOrderHistoryComponent	UnitLevelOrderHistoryComponent	[AuthGuard, UnitLevelOrdersViewerGuard]			organizationUnitOrder	UnitLevelOrderHistoryModule	@spartacus/organization/unit-order/components	feature-libs/organization/unit-order/components/unit-level-order-history/unit-level-order-history.module.ts:L43-L46
244	UpdateEmailComponent	MyAccountV2EmailComponent				userProfile	myAccountV2EmailMapping	@spartacus/user/profile/components	feature-libs/user/profile/components/update-email/update-email.module.ts:L42-L44
245	UpdateEmailComponent	UpdateEmailComponent	[AuthGuard]	UpdateEmailComponentService → useClass: UpdateEmailComponentService		userProfile	UpdateEmailModule	@spartacus/user/profile/components	feature-libs/user/profile/components/update-email/update-email.module.ts:L71-L87
246	UpdateEmailComponent			UpdateEmailComponentService → useClass: CDCUpdateEmailComponentService		userProfile	CDCUpdateEmailModule	@spartacus/cdc/user-profile	integration-libs/cdc/user-profile/update-email/cdc-update-email.module.ts:L40-L47
247	UpdatePasswordComponent	MyAccountV2PasswordComponent				userProfile	myAccountV2PasswordMapping	@spartacus/user/profile/components	feature-libs/user/profile/components/update-password/update-password.module.ts:L42-L44
248	UpdatePasswordComponent	UpdatePasswordComponent	[AuthGuard]	UpdatePasswordComponentService → useClass: UpdatePasswordComponentService		userProfile	UpdatePasswordModule	@spartacus/user/profile/components	feature-libs/user/profile/components/update-password/update-password.module.ts:L70-L86
249	UpdatePasswordComponent			UpdatePasswordComponentService → useClass: CDCUpdatePasswordComponentService		userProfile	CDCUpdatePasswordModule	@spartacus/cdc/user-profile	integration-libs/cdc/user-profile/update-password/cdc-update-password.module.ts:L40-L47
250	UpdateProfileComponent	MyAccountV2ProfileComponent				userProfile	myAccountV2ProfileMapping	@spartacus/user/profile/components	feature-libs/user/profile/components/update-profile/update-profile.module.ts:L38-L40
251	UpdateProfileComponent	UpdateProfileComponent	[AuthGuard]	UpdateProfileComponentService → useClass: UpdateProfileComponentService		userProfile	UpdateProfileModule	@spartacus/user/profile/components	feature-libs/user/profile/components/update-profile/update-profile.module.ts:L67-L77

#	TYPE CODE CMS	COMPONENTE	GUARDS	PROVIDERS	ALTRE CHIAVI	LAZY IN FEATURE	CONTENITORE	LIBRERIA	RIFERIMENTO
252	UpdateProfileComponent			UpdateProfileComponentService → useClass: CDCUpdateProfileComponentService		userProfile	CDCUpdateProfileModule	@spartacus/cdc/user-profile	integration-libs/cdc/user-profile/update-profile/cdc-update-profile.module.ts:L42-L49
253	VerifyOTPForB2BRegistrationComponent	RegisterVerificationTokenFormComponent	[NotAuthGuard]	RegisterVerificationTokenFormComponentService → useClass: RegisterVerificationTokenFormComponentService		organizationUserRegistration	RegisterVerificationTokenFormModule	@spartacus/organization/user-registration/components	feature-libs/organization/user-registration/components/verification-token-form/verification-token-form.module.ts:L53-L68
254	VerifyOTPForRegistrationComponent	RegistrationVerificationTokenFormComponent	[NotAuthGuard]	RegistrationVerificationTokenFormComponentService → useClass: RegistrationVerificationTokenFormComponentService		userProfile	RegistrationVerificationTokenFormModule	@spartacus/user/profile/components	feature-libs/user/profile/components/registration-verification-token-form/verify-register-verification-token-form.module.ts:L50-L60
255	VerifyOTPTokenComponent	VerificationTokenFormComponent	[NotAuthGuard, CustomLoginGuard]	VerificationTokenFormComponentService → useClass: VerificationTokenFormComponentService		userAccount	VerificationTokenFormModule	@spartacus/user/account/components	feature-libs/user/account/components/verification-token-form/verification-token-form.module.ts:L59-L74
256	VideoComponent	VideoComponent					VideoModule	@spartacus/storefront	core-libs/storefront/cms-components/content/video/video.module.ts:L29-L31
257	VisualPickingTabComponent	VisualPickingTabComponent				epd-visualization	VisualPickingTabModule	@spartacus/epd-visualization/components	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.module.ts:L32-L34
258	WishListComponent	WishListComponent	[AuthGuard]		data	cartWishList	WishListComponentsModule	@spartacus/cart/wish-list/components	feature-libs/cart/wish-list/components/wish-list-components.module.ts:L46-L54

Tabella 2 — Type code lazy per feature (`featureModules.<feature>.cmsComponents`)

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
1	addToCart	[ADD_TO_CART_FEATURE]	1	ProductAddToCartComponent		feature-libs/cart/base/root/cart-base-root.module.ts:L43-L43	@spartacus/cart/base/root	feature-libs/cart/base/root/cart-base-root.module.ts:L42-L44
2	cartBase	[CART_BASE_FEATURE]	6	CartApplyCouponComponent, CartComponent, CartProceedToCheckoutComponent, CartTotalsComponent, SaveForLaterComponent, ClearCartComponent		feature-libs/cart/base/root/cart-base-root.module.ts:L30-L37	@spartacus/cart/base/root	feature-libs/cart/base/root/cart-base-root.module.ts:L29-L38

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
3	miniCart	[MINI_CART_FEATURE]	1	MiniCartComponent		feature-libs/cart/base/root/cart-base-root.module.ts:L40-L40	@spartacus/cart/base/root	feature-libs/cart/base/root/cart-base-root.module.ts:L39-L41
4	cartImportExport	[CART_IMPORT_EXPORT_FEATURE]	3	ImportExportOrderEntriesComponent, ImportOrderEntriesComponent, ExportOrderEntriesComponent		feature-libs/cart/import-export/root/import-export-root.module.ts:L16-L20	@spartacus/cart/import-export/root	feature-libs/cart/import-export/root/import-export-root.module.ts:L15-L21
5	cartQuickOrder	[CART_QUICK_ORDER_FEATURE]	2	QuickOrderComponent, CartQuickOrderFormComponent		feature-libs/cart/quick-order/root/quick-order-root.module.ts:L27-L27	@spartacus/cart/quick-order/root	feature-libs/cart/quick-order/root/quick-order-root.module.ts:L26-L28
6	cartSavedCart	[CART_SAVED_CART_FEATURE]	5	AddToSavedCartsComponent, AccountSavedCartHistoryComponent, SavedCartDetailsOverviewComponent, SavedCartDetailsItemsComponent, SavedCartDetailsActionComponent		feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L35-L41	@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/saved-cart-root.module.ts:L34-L43
7	addToWishList	[ADD_TO_WISHLIST_FEATURE]	1	AddToWishListComponent		feature-libs/cart/wish-list/root/wish-list-root.module.ts:L24-L24	@spartacus/cart/wish-list/root	feature-libs/cart/wish-list/root/wish-list-root.module.ts:L23-L25
8	cartWishList	[CART_WISH_LIST_FEATURE]	1	WishListComponent		feature-libs/cart/wish-list/root/wish-list-root.module.ts:L20-L20	@spartacus/cart/wish-list/root	feature-libs/cart/wish-list/root/wish-list-root.module.ts:L19-L22
9	checkout	[CHECKOUT_FEATURE] / CHECKOUT_B2B_CMS_COMPONENTS	16	CheckoutOrchestrator, CheckoutOrderSummary, CheckoutProgress, CheckoutProgressMobileBottom, CheckoutProgressMobileTop, CheckoutDeliveryMode, CheckoutPaymentDetails, CheckoutPlaceOrder, CheckoutReviewOrder, CheckoutReviewPayment, CheckoutReviewShipping, CheckoutReviewOverview, CheckoutDeliveryAddress, GuestCheckoutLoginComponent, CheckoutCostCenterComponent, CheckoutPaymentType	... CHECKOUT_BASE_CMS_COMPONENTS	feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L23-L23	@spartacus/checkout/b2b/root	feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L36-L38
10	checkout	[CHECKOUT_FEATURE] / CHECKOUT_BASE_CMS_COMPONENTS	14	CheckoutOrchestrator, CheckoutOrderSummary, CheckoutProgress, CheckoutProgressMobileBottom, CheckoutProgressMobileTop, CheckoutDeliveryMode, CheckoutPaymentDetails, CheckoutPlaceOrder, CheckoutReviewOrder, CheckoutReviewPayment, CheckoutReviewShipping, CheckoutReviewOverview, CheckoutDeliveryAddress, GuestCheckoutLoginComponent		feature-libs/checkout/base/root/checkout-root.module.ts:L20-L20	@spartacus/checkout/base/root	feature-libs/checkout/base/root/checkout-root.module.ts:L40-L43

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
11	checkout	[CHECKOUT_FEATURE] / CHECKOUT_SCHEDULED_REPLENISHMENT_CMS_COMPONENTS	17	CheckoutOrchestrator, CheckoutOrderSummary, CheckoutProgress, CheckoutProgressMobileBottom, CheckoutProgressMobileTop, CheckoutDeliveryMode, CheckoutPaymentDetails, CheckoutPlaceOrder, CheckoutReviewOrder, CheckoutReviewPayment, CheckoutReviewShipping, CheckoutReviewOverview, CheckoutDeliveryAddress, GuestCheckoutLoginComponent, CheckoutCostCenterComponent, CheckoutPaymentType, CheckoutScheduleReplenishmentOrder	...CHECKOUT_B2B_CMS_COMPONENTS, ...CHECKOUT_BASE_CMS_COMPONENTS	feature-libs/checkout/scheduled-replenishment/root/checkout-scheduled-replenishment-root.module.ts:L13-L13	@spartacus/checkout/scheduled-replenishment/root	feature-libs/checkout/scheduled-replenishment/root/checkout-scheduled-replenishment-root.module.ts:L25-L27
12	customerTicketing	[CUSTOMER_TICKETING_FEATURE]	6	SupportTicketHistoryComponent, SupportTicketUpdateComponent, SupportTicketDetailsComponent, SupportTicketReopenComponent, SupportTicketCloseComponent, MyAccountViewRequestsComponent		feature-libs/customer-ticketing/root/customer-ticketing-root.module.ts:L28-L35	@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/customer-ticketing-root.module.ts:L27-L36
13	orderDocumentFlow	[ORDER_DOCUMENT_FLOW_FEATURE]	1	AccountOrderDocumentFlowComponent		feature-libs/order/document-flow/root/order-document-flow-root.module.ts:L16-L16	@spartacus/order/document-flow/root	feature-libs/order/document-flow/root/order-document-flow-root.module.ts:L15-L17

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
14	order	[ORDER_FEATURE] / ORDER_CMS_ COMPONENTS	36	CancelOrderComponent, Cancel OrderConfirmationComponent, ReturnOrderComponent, Return OrderConfirmationComponent, AccountOrderDetailsActions Component, AccountOrder DetailsItemsComponent, AccountOrderDetailsTotals Component, AccountOrder DetailsOverviewComponent, AccountOrderDetailsBilling Component, AccountOrder DetailsGroupedItems Component, AccountOrder DetailsSimpleOverview Component, AccountOrder HistoryComponent, ReplenishmentDetailItems Component, AccountOrder DetailsReorderComponent, ReplenishmentDetailTotals Component, Replenishment DetailShippingComponent, ReplenishmentDetailActions Component, Replenishment DetailOrderHistory Component, Account ReplenishmentHistory Component, ReturnRequest OverviewComponent, Return RequestItemsComponent, ReturnRequestTotals Component, OrderReturn RequestListComponent, Order ConfirmationThankMessage Component, OrderConfirmation ItemsComponent, Order ConfirmationTotals Component, OrderConfirmation OverviewComponent, Order ConfirmationShipping Component, OrderConfirmation BillingComponent, Order ConfirmationContinueButton Component, Replenishment ConfirmationMessage Component, Replenishment ConfirmationOverview Component, Replenishment ConfirmationItemsComponent, ReplenishmentConfirmation TotalsComponent, MyAccount ViewOrderComponent, Account OrderAttachmentsComponent		feature-libs/ order/root/ order-root. module.ts:L28-L28	@spartacus/ order/root	feature-libs/ order/root/ order-root. module.ts:L71-L74
15	organization AccountSummary	[ORGANIZATION_ ACCOUNT_SUMMARY_ FEATURE]	3	ManageAccountSummaryList Component, AccountSummary HeaderComponent, Account SummaryDocumentComponent		feature-libs/ organization/ account-summary/ root/account-summary-root. module.ts:L25-L29	@spartacus/ organization/ account-summary/ root	feature-libs/ organization/ account-summary/ root/account-summary-root. module.ts:L24-L30
16	organization Administration	[ORGANIZATION_ ADMINISTRATION_ FEATURE]	6	ManageBudgetsListComponent, ManageCostCentersList Component, ManagePermissions ListComponent, ManageUnits ListComponent, ManageUsers ListComponent, ManageUser GroupsListComponent		feature-libs/ organization/ administration/ root/ administration-root.module.ts: L29-L36	@spartacus/ organization/ administration/ root	feature-libs/ organization/ administration/ root/ administration-root.module.ts: L28-L37

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
17	organization OrderApproval	[ORGANIZATION_ ORDER_APPROVAL_ FEATURE]	7	OrderApprovalListComponent, OrderApprovalDetailTotals Component, OrderApproval DetailApprovalDetails Component, OrderApproval DetailShippingComponent, OrderApprovalDetailItems Component, OrderApproval DetailFormComponent, Account OrderDetailsApprovalDetails Component		feature-libs/ organization/ order-approval/ root/order- approval-root. module.ts:L25- L33	@spartacus/ organization/ order-approval/ root	feature-libs/ organization/ order-approval/ root/order- approval-root. module.ts:L24- L35
18	organization UnitOrder	[ORGANIZATION_ UNIT_ORDER_ FEATURE]	4	UnitLevelOrderHistory Component, UnitLevelOrder DetailsOverviewComponent, UnitLevelOrderDetailsItems Component, UnitLevelOrder DetailsTotalsComponent		feature-libs/ organization/ unit-order/root/ unit-order-root. module.ts:L27- L32	@spartacus/ organization/ unit-order/root	feature-libs/ organization/ unit-order/root/ unit-order-root. module.ts:L26- L34
19	organization User Registration	[ORGANIZATION_ USER_ REGISTRATION_ FEATURE]	3	OrganizationUser RegistrationComponent, RegisterB2BCustomerWith OTPComponent, VerifyOTPForB2 BRegistrationComponent		feature-libs/ organization/ user- registration/ root/user- registration- root.module.ts: L15-L19	@spartacus/ organization/ user- registration/ root	feature-libs/ organization/ user- registration/ root/user- registration- root.module.ts: L14-L20
20	pdfInvoices	[PDF_INVOICES_ FEATURE]	1	AccountOrderDetails PDFInvoicesComponent		feature-libs/ pdf-invoices/ root/pdf- invoices-root. module.ts:L18- L18	@spartacus/pdf- invoices/root	feature-libs/ pdf-invoices/ root/pdf- invoices-root. module.ts:L17- L19
21	pickupInStore	[PICKUP_IN_STORE_ FEATURE]	4	CheckoutReviewPickup, My PreferredStoreComponent, OrderConfirmationPickUp Component, PickupInStore DeliveryModeComponent		feature-libs/ pickup-in-store/ root/pickup-in- store-root. module.ts:L27- L32	@spartacus/ pickup-in-store/ root	feature-libs/ pickup-in-store/ root/pickup-in- store-root. module.ts:L26- L33
22	product Configurator Rulebased	[PRODUCT_ CONFIGURATOR_ RULEBASED_ FEATURE] / cms Components	18	ConfiguratorForm, ConfiguratorOverviewForm, ConfiguratorOverviewMenu, ConfiguratorUpdateMessage, ConfiguratorAddToCart Button, ConfiguratorMenu, ConfiguratorGroupTitle, ConfiguratorOverviewBanner, ConfiguratorPrevNext, ConfiguratorPriceSummary, ConfiguratorProductTitle, ConfiguratorTabBar, ConfiguratorExitButton, ConfiguratorVariant Carousel, CpqConfigurator ConflictAndErrorMessages Component, Configurator OverviewFilterButton, ConfiguratorOverviewFilter, ConfiguratorOverviewSidebar		feature-libs/ product- configurator/ rulebased/root/ rulebased- configurator- root-feature. module.ts:L11- L11	@spartacus/ product- configurator/ rulebased/root	feature-libs/ product- configurator/ rulebased/root/ rulebased- configurator- root-feature. module.ts:L36- L38

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
23	productConfiguratorTextfield	[PRODUCT_CONFIGURATOR_TEXTFIELD_FEATURE] / cmsComponents	1	TextfieldConfigurationForm		feature-libs/product-configurator/textfield/root/textfield-configurator-root-feature.module.ts:L11-L11	@spartacus/product-configurator/textfield/root	feature-libs/product-configurator/textfield/root/textfield-configurator-root-feature.module.ts:L17-L19
24	productMultiDimensionalList	[PRODUCT_MULTI_DIMENSIONAL_LIST_FEATURE]	3	CMSProductListComponent, ProductGridComponent, SearchResultsListComponent		feature-libs/product-multi-dimensional/list/root/product-multi-dimensional-list-root.module.ts:L23-L27	@spartacus/product-multi-dimensional/list/root	feature-libs/product-multi-dimensional/list/root/product-multi-dimensional-list-root.module.ts:L22-L28
25	productMultiDimensionalSelector	[PRODUCT_MULTI_DIMENSIONAL_SELECTOR_FEATURE]	1	ProductMultiDimensionalSelectorComponent		feature-libs/product-multi-dimensional/selector/root/product-multi-dimensional-selector-root.module.ts:L19-L19	@spartacus/product-multi-dimensional/selector/root	feature-libs/product-multi-dimensional/selector/root/product-multi-dimensional-selector-root.module.ts:L18-L20
26	productBulkPricing	[PRODUCT_BULK_PRICING_FEATURE]	1	BulkPricingTableComponent		feature-libs/product/bulk-pricing/root/bulk-pricing-root.module.ts:L16-L16	@spartacus/product/bulk-pricing/root	feature-libs/product/bulk-pricing/root/bulk-pricing-root.module.ts:L15-L17
27	productFutureStock	[PRODUCT_FUTURE_STOCK_FEATURE]	1	FutureStockComponent		feature-libs/product/future-stock/root/future-stock-root.module.ts:L15-L15	@spartacus/product/future-stock/root	feature-libs/product/future-stock/root/future-stock-root.module.ts:L14-L16
28	productImageZoom	[PRODUCT_IMAGE_ZOOM_FEATURE]	1	ProductImagesComponent		feature-libs/product/image-zoom/root/product-image-zoom-root.module.ts:L15-L15	@spartacus/product/image-zoom/root	feature-libs/product/image-zoom/root/product-image-zoom-root.module.ts:L14-L16
29	productVariants	[PRODUCT_VARIANTS_FEATURE]	1	ProductVariantSelectorComponent		feature-libs/product/variants/root/product-variants-root.module.ts:L23-L23	@spartacus/product/variants/root	feature-libs/product/variants/root/product-variants-root.module.ts:L22-L24
30	qualtrics	[QUALTRICS_FEATURE]	1	QualtricsComponent		feature-libs/qualtrics/root/qualtrics-root.module.ts:L16-L16	@spartacus/qualtrics/root	feature-libs/qualtrics/root/qualtrics-root.module.ts:L15-L17

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
31	quote	[QUOTE_FEATURE]	6	AccountMyQuotesComponent, QuoteLinksComponent, QuoteCommentsComponent, QuoteHeaderOverviewComponent, QuoteItemsComponent, QuoteSummaryComponent		feature-libs/quote/root/quote-root.module.ts:L33-L40	@spartacus/quote/root	feature-libs/quote/root/quote-root.module.ts:L32-L41
32	quote_cart_guard	[QUOTE_CART_GUARD_FEATURE]	1	QuoteCartGuardComponent		feature-libs/quote/root/quote-root.module.ts:L50-L50	@spartacus/quote/root	feature-libs/quote/root/quote-root.module.ts:L49-L51
33	quote_request	[QUOTE_REQUEST_FEATURE]	1	QuoteRequestComponent		feature-libs/quote/root/quote-root.module.ts:L60-L60	@spartacus/quote/root	feature-libs/quote/root/quote-root.module.ts:L59-L61
34	requested DeliveryDate	[REQUESTED_DELIVERY_DATE_FEATURE]	2	DeliveryModeDatePickerComponent, OrderOverviewDeliveryDateComponent		feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L22-L25	@spartacus/requested-delivery-date/root	feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L21-L26
35	stores	[STORE_FINDER_FEATURE]	1	StoreFinderComponent		feature-libs/storefinder/root/store-finder-root.module.ts:L22-L22	@spartacus/storefinder/root	feature-libs/storefinder/root/store-finder-root.module.ts:L21-L23
36	subscription Billing	[SUBSCRIPTION_BILLING_FEATURE]	5	SubscriptionHistoryComponent, SubscriptionProductPriceComponent, SubscriptionDetailsComponent, SubscriptionBillsHistoryComponent, SubscriptionBillDetailsComponent		feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L28-L34	@spartacus/subscription-billing/root	feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L27-L35
37	personalization	[PERSONALIZATION_FEATURE]	1	PersonalizationScriptComponent		feature-libs/tracking/personalization/root/personalization-root.module.ts:L22-L22	@spartacus/tracking/personalization/root	feature-libs/tracking/personalization/root/personalization-root.module.ts:L21-L23
38	userAccount	[USER_ACCOUNT_FEATURE]	9	LoginComponent, OAuthCallbackComponent, ReturningCustomerLoginComponent, VerifyOTPTokenComponent, ReturningCustomerRegisterComponent, MyAccountViewUserComponent, ReturningCustomerOTPLoginComponent, RegisterCustomerWithOTPComponent, ReturningOrganizationUserRegisterComponent		feature-libs/user/account/root/user-account-root.module.ts:L33-L43	@spartacus/user/account/root	feature-libs/user/account/root/user-account-root.module.ts:L32-L44

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
39	userProfile	[USER_PROFILE_FEATURE]	10	RegisterCustomerComponent, UpdateProfileComponent, UpdateEmailComponent, UpdatePasswordComponent, ForgotPasswordComponent, ResetPasswordComponent, CloseAccountComponent, AccountAddressBookComponent, RegisterCustomerWithOTPComponent, VerifyOTPForRegistrationComponent		feature-libs/user/profile/root/user-profile-root.module.ts:L25-L36	@spartacus/user/profile/root	feature-libs/user/profile/root/user-profile-root.module.ts:L24-L37
40	cdc	[CDC_FEATURE]	1	GigyaRaasComponent		integration-libs/cdc/root/cdc-root.module.ts:L44-L44	@spartacus/cdc/root	integration-libs/cdc/root/cdc-root.module.ts:L43-L45
41	epd-visualization	[EPD_VISUALIZATION_FEATURE]	1	VisualPickingTabComponent		integration-libs/epd-visualization/root/epd-visualization-root.module.ts:L21-L21	@spartacus/epd-visualization/root	integration-libs/epd-visualization/root/epd-visualization-root.module.ts:L20-L22
42	opfB2bCheckout	[OPF_B2B_CHECKOUT_FEATURE] / OPF_B2B_CHECKOUT_CMS_COMPONENTS	4	OpfCheckoutPaymentType, OpfCheckoutDeliveryAddress, OpfCheckoutReview, OpfB2bCheckoutPaymentAndReview		integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts:L19-L19	@spartacus/opf/b2b-checkout/root	integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts:L29-L31
43	opfCheckout	[OPF_CHECKOUT_FEATURE] / CHECKOUT_OPF_CMS_COMPONENTS	2	OpfCheckoutPaymentAndReview, OpfCheckoutEmailUpdateComponent		integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L26-L26	@spartacus/opf/checkout/root	integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L34-L36
44	opfCta	[OPF_CTA_FEATURE]	2	OpfCtaScriptsComponent, OpfCtaQuickBuyButtons		integration-libs/opf/cta/root/opf-cta-root.module.ts:L15-L15	@spartacus/opf/cta/root	integration-libs/opf/cta/root/opf-cta-root.module.ts:L14-L16
45	opfGiftCard	[OPF_GIFT_CARD_FEATURE] / OPF_GIFT_CARD_FEATURE_CMS_COMPONENTS	3	CheckoutOrderSummary, OrderConfirmationTotalsComponent, AccountOrderDetailsTotalsComponent		integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L22-L22	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L31-L33
46	opfQuickBuy	[OPF_QUICK_BUY_FEATURE]	1	OpfQuickBuyButtonsComponent		integration-libs/opf/quick-buy/root/opf-quick-buy-root.module.ts:L20-L20	@spartacus/opf/quick-buy/root	integration-libs/opf/quick-buy/root/opf-quick-buy-root.module.ts:L19-L21
47	opfTokenisation	[OPF_TOKENISATION_FEATURE]	1	AccountPaymentDetailsComponent		integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L26-L26	@spartacus/opf/tokenisation/root	integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L25-L27

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
48	punchout	[PUNCHOUT_ FEATURE]	6	PunchoutSessionComponent, PunchoutErrorComponent, PunchoutButtonsComponent, PunchoutRequisition Component, PunchoutClose SessionComponent, Punchout InspectCartComponent		integration-libs/punchout/root/config/default-punchout-cms-component-config.ts:L13-L20	@spartacus/ punchout/root	integration-libs/punchout/root/config/default-punchout-cms-component-config.ts:L12-L21
49	checkout	[CHECKOUT_ FEATURE] / S4_ SERVICE_CMS_ COMPONENTS	17	CheckoutOrchestrator, CheckoutOrderSummary, CheckoutProgress, Checkout ProgressMobileBottom, CheckoutProgressMobileTop, CheckoutDeliveryMode, CheckoutPaymentDetails, CheckoutPlaceOrder, Checkout ReviewOrder, CheckoutReview Payment, CheckoutReview Shipping, CheckoutReview Overview, CheckoutDelivery Address, GuestCheckoutLogin Component, CheckoutCost CenterComponent, Checkout PaymentType, CheckoutService Details	... CHECKOUT_B2 B_CMS_ COMPONENTS, ... CHECKOUT_ BASE_CMS_ COMPONENTS	integration-libs/s4-service/root/s4-service-root.module.ts:L31-L31	@spartacus/s4-service/root	integration-libs/s4-service/root/s4-service-root.module.ts:L43-L45

#	FEATURE	CHIAVE SORGENTE	N.	TYPE CODE	SPREAD ESPANSI	LISTA DEFINITA IN	LIBRERIA	RIFERIMENTO (ENTRY FEATURE)
50	order	[ORDER_FEATURE]	39	CancelOrderComponent, CancelOrderConfirmationComponent, ReturnOrderComponent, ReturnOrderConfirmationComponent, AccountOrderDetailsActionsComponent, AccountOrderDetailsItemsComponent, AccountOrderDetailsTotalsComponent, AccountOrderDetailsOverviewComponent, AccountOrderDetailsBillingComponent, AccountOrderDetailsGroupedItemsComponent, AccountOrderDetailsSimpleOverviewComponent, AccountOrderHistoryComponent, ReplenishmentDetailItemsComponent, AccountOrderDetailsReorderComponent, ReplenishmentDetailTotalsComponent, ReplenishmentDetailShippingComponent, ReplenishmentDetailActionsComponent, ReplenishmentDetailOrderHistoryComponent, AccountReplenishmentHistoryComponent, ReturnRequestOverviewComponent, ReturnRequestItemsComponent, ReturnRequestTotalsComponent, OrderReturnRequestListComponent, OrderConfirmationThankMessageComponent, OrderConfirmationItemsComponent, OrderConfirmationTotalsComponent, OrderConfirmationOverviewComponent, OrderConfirmationShippingComponent, OrderConfirmationBillingComponent, OrderConfirmationContinueButtonComponent, ReplenishmentConfirmationMessageComponent, ReplenishmentConfirmationOverviewComponent, ReplenishmentConfirmationItemsComponent, ReplenishmentConfirmationTotalsComponent, MyAccountViewOrderComponent, AccountOrderAttachmentsComponent, RescheduleServiceOrder, CancelServiceOrderHeadline, CancelServiceOrder	...ORDER_CMS_COMPONENTS, ...S4_SERVICE_ORDER_CMS_COMPONENTS	integration-libs/s4-service/root/s4-service-root.module.ts:L47-L50	@spartacus/s4-service/root	integration-libs/s4-service/root/s4-service-root.module.ts:L46-L51

Type code lazy senza mapping `cmsComponents` statico in ambito

Possono essere mappati in file esclusi dall'ambito, in config costruite dinamicamente, o essere solo dichiarati per attivare il caricamento della feature.

`AccountOrderDetailsBillingComponent`, `ConfiguratorOverviewFilter`, `ConfiguratorOverviewMenu`, `DeliveryModeDatePickerComponent`, `OrderConfirmationContinueButtonComponent`, `OrderOverviewDeliveryDateComponent`, `PersonalizationScriptComponent`

A04

Endpoint OCC

Endpoint di default in `backend.occ.endpoints`

IN QUESTO CAPITOLO

Conteggi

Come rigenerare

Tabella 1 — backend.occ.endpoints

Tabella 2 — Altri blocchi endpoints (non backend.occ)

Tabella 3 — Augmentation di OccEndpoints

Generata da `tools/extract2/a04_occ_endpoints.py` allo SHA 5b7a1c2b. Non modificare a mano: rigenerare.

Tabella 1: ogni chiave dentro un object literal `endpoints: { ... }` il cui percorso di chiavi è `backend.occ.endpoints` (default config OCC, tipicamente `default-occ-*-config.ts`, incluse le factory). Il valore è riportato come scritto nel sorgente (stringa template OCC, oggetto con scope, o espressione). Gli spread (`...orderEndpoints`) sono espansi risolvendo la `const` corrispondente: la riga punta alla definizione della chiave, la colonna "Via spread" al punto di spread. Una chiave ripetuta indica più default (es. override in librerie diverse, merge con priorità di config).

Tabella 2: altri blocchi `endpoints` (es. OPF API) per completezza.

Tabella 3: le interface augmentation `interface OccEndpoints` dentro `declare module '...'`.

Conteggi

- Chiavi endpoint in `backend.occ.endpoints` (righe tabella 1): **269**
- Chiavi OCC distinte: **248**
- di cui con template a oggetto (scope): **8**
- File default config OCC: **51**
- Chiavi in altri `endpoints` non OCC (es. `backend.opfApi.endpoints`) (tabella 2): **9**
- Blocchi `declare module ... { interface OccEndpoints }` (tabella 3): **35**
- Membri dichiarati via augmentation / nell'interfaccia base: **195 / 49**
- Chiavi OCC con default ma non dichiarate in `OccEndpoints` (base+augmentation): **4**
- Membri di `OccEndpoints` senza default in ambito: **0**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract2/a04_occ_endpoints.py
# oppure tutte le appendici: bash tools/extract2/run_all.sh
```

Ambito: `git ls-files core-libs feature-libs integration-libs | grep '\.ts$' | grep -vE '\.spec\.ts$|/testing/|\.mock\.ts$|/schematics/'`. La colonna "Libreria" è l'entry point npm ricavato dal `package.json`/`ng-package.json` più vicino.

Tabella 1 — backend.occ.endpoints

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
1	accountSummary	'users/${userId}/orgUnits/${orgUnitId}/accountSummary'	stringa		defaultOccAccountSummaryConfig	…accountSummaryHeaderOccEndpoints (feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L21-L21)	@spartacus/organization/account-summary/occ	feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L11-L11
2	accountSummaryDocument	'users/${userId}/orgUnits/${orgUnitId}/orgDocuments'	stringa		defaultOccAccountSummaryConfig	…accountSummaryHeaderOccEndpoints (feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L21-L21)	@spartacus/organization/account-summary/occ	feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L12-L12
3	accountSummaryDocumentAttachment	'users/${userId}/orgUnits/${orgUnitId}/orgDocuments/${orgDocumentId}/attachments/${orgDocumentAttachmentId}'	stringa		defaultOccAccountSummaryConfig	…accountSummaryHeaderOccEndpoints (feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L21-L21)	@spartacus/organization/account-summary/occ	feature-libs/organization/account-summary/occ/config/default-occ-account-summary-config.ts:L13-L14
4	addComment	'users/${userId}/quotes/${quoteCode}/comments'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L25-L25
5	addCpqConfigurationToCart	'users/${userId}/carts/${cartId}/entries/cpqconfigurator'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L15-L16
6	addDiscount	'users/${userId}/quotes/${quoteCode}/discounts'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L26-L26

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
7	addEmail	'users/${userId}/carts/${cartId}/email'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L25-L25
8	addEntries	'orgUsers/${userId}/carts/${cartId}/entries?quantity=${quantity}'	stringa		defaultB2bOccConfig	...defaultB2bCartOccEndpoints (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L44-L44)	@spartacus/setup	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L28-L28
9	addEntries	'users/${userId}/carts/${cartId}/entries'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L20-L20
10	addQuoteEntryComment	'users/${userId}/quotes/${quoteCode}/entries/${entryNumber}/comments'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L27-L28
11	addressCities	'regions/${regionId}/cities?fields=cities(name,isocode)'	stringa		defaultOccSiteContextConfig		@spartacus/core	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L18-L18
12	addressDetail	'users/${userId}/addresses/${addressId}'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L20-L20
13	addressDistricts	'cities/${cityId}/districts?fields=districts(name,isocode)'	stringa		defaultOccSiteContextConfig		@spartacus/core	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L19-L20
14	addresses	'users/${userId}/addresses'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L19-L19
15	addressVerification	'users/${userId}/addresses/verification'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L21-L21

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
16	addTextfieldConfigurationToCart	'users/${userId}/carts/${cartId}/entries/configurator/textfield'	stringa		defaultOccConfiguratorTextfieldConfigFactory		@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L17-L18
17	addVariantConfigurationToCart	'users/${userId}/carts/${cartId}/entries/ccpconfigurator'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L21-L22
18	addWishlistEntry	'users/${userId}/wishlists/${wishlistId}/entries'	stringa		defaultOccUserWishlistConfig		@spartacus/user/wishlist/occ	feature-libs/user/wishlist/occ/adapters/config/default-occ-user-wishlist-endpoint.config.ts:L15-L15
19	anonymousConsentTemplates	'users/anonymous/consenttemplates'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L15-L15
20	applyGiftCard	'users/${userId}/carts/${cartId}/giftCards'	stringa		defaultGiftCardOccEndpointsConfig		@spartacus/opf/gift-card/occ	integration-libs/opf/gift-card/occ/config/default-occ-opf-gift-card-config.ts:L13-L13
21	asmBindCart	'/assistedservicewebservices/bind-cart'	stringa		defaultOccAsmConfig		@spartacus/asm/occ	feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L15-L15
22	asmCreateCustomer	'/assistedservicewebservices/customers'	stringa		defaultOccAsmConfig		@spartacus/asm/occ	feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L16-L16
23	asmCustomer360	'/assistedservicewebservices/${baseSiteId}/users/${userId}/customer360'	stringa		defaultOccAsmCustomer360Config		@spartacus/asm/customer-360/occ	feature-libs/asm/customer-360/occ/adapters/default-occ-asm-customer-360-config.ts:L13-L14

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
24	asmCustomerLists	'/assistedservicewebservices/customerlists'	stringa		defaultOccAsmConfig		@spartacus/asm/occ	feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L14-L14
25	asmCustomerSearch	'/assistedservicewebservices/customers/search'	stringa		defaultOccAsmConfig		@spartacus/asm/occ	feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L13-L13
26	asmSessionEvent	'assistedservicewebservices/${baseSiteId}/users/${userId}/asmSessionEvents'	stringa		defaultOccAsmConfig		@spartacus/asm/occ	feature-libs/asm/occ/adapters/default-occ-asm-config.ts:L17-L18
27	b2bUser	'/users/${userId}/orgCustomers/${orgCustomerId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L59-L59
28	b2bUserApprover	'/users/${userId}/orgCustomers/${orgCustomerId}/approvers/${approverId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L62-L63
29	b2bUserApprovers	'/users/${userId}/orgCustomers/${orgCustomerId}/approvers'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L60-L61
30	b2bUserPermission	'/users/${userId}/orgCustomers/${orgCustomerId}/permissions/${premissionId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L70-L71
31	b2bUserPermissions	'/users/${userId}/orgCustomers/${orgCustomerId}/permissions'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L68-L69

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
32	b2bUsers	'/users/${userId}/orgCustomers'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L58-L58
33	b2bUserUserGroup	'/users/${userId}/orgCustomers/${orgCustomerId}/orgUserGroups/${userGroupId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L66-L67
34	b2bUserUserGroups	'/users/${userId}/orgCustomers/${orgCustomerId}/orgUserGroups'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L64-L65
35	baseSites	'basesites?fields=FULL'	stringa		defaultOccSiteContextConfig		@spartacus/core	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L21-L21
36	budget	'/users/${userId}/budgets/${budgetCode}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L14-L14
37	budgets	'/users/${userId}/budgets'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L13-L13
38	cancelOrder	'users/${userId}/orders/${orderId}/cancellation'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L18-L18
39	cancelReplenishmentOrder	'users/${userId}/replenishmentOrders/${replenishmentOrderCode}?fields=FULL,costCenter(FULL),purchaseOrderNumber,paymentType,user'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L30-L31

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
40	cancelReturn	'users/${userId}/orderReturns/${returnRequestCode}'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L24-L24
41	cancelServiceOrder	'users/${userId}/orders/${code}/serviceOrder/cancellation'	stringa		defaultOccServiceOrderConfig	...s4ServiceOccEndpoints (integration-libs/s4-service/order/occ/config/default-occ-s4-service-config.ts:L21-L21)	@spartacus/s4-service/order	integration-libs/s4-service/order/occ/config/default-occ-s4-service-config.ts:L11-L12
42	cancelSubscription	'users/${userId}/subscriptions/${subscriptionCode}/cancellation'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L20-L21
43	cardTypes	'cardtypes'	stringa		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L17-L17
44	cart	'users/${userId}/carts/${cartId}?fields=DEFAULT,potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrderPromotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),basePrice(formattedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),totalDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,deliveryCost(formattedValue),totalTax(formattedValue, value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),user,saveTime,name,description'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L17-L17
45	cart	'users/${userId}/carts/${cartId}?fields=DEFAULT,potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrderPromotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),arrivalSlots,basePrice(formattedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),totalDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,deliveryCost(formattedValue),totalTax(formattedValue, value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),user,saveTime,name,description'	stringa		defaultOccCartWithEddConfig	...defaultEddOccEndpoints (feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L23-L23)	@spartacus/estimated-delivery-date/show-estimated-delivery-date	feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L13-L13

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
46	cart	'users/${userId}/carts/${cartId}?fields=DEFAULT,sapGiftCards(FULL,balance(value,formattedValue),remainingBalance(value,formattedValue), appliedAmount(value,formattedValue)),sapGiftCardSummary(FULL,totalAppliedAmount(value,formattedValue),totalRemainingBalance(value,formattedValue),totalBalance(value,formattedValue)),potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrderPromotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),basePrice(formattedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),totalDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,deliveryCost(formattedValue),totalTax(formattedValue,value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),user,saveTime,name,description,_availableOperations'	stringa		defaultOccOpfGiftCardCartEndpointsConfig	...giftCardCartEndpoints (integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-cart-config.ts:L20-L20)	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-cart-config.ts:L13-L13
47	cart	'users/${userId}/carts/${cartId}?fields=DEFAULT,potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrderPromotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),basePrice(formattedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),totalDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,deliveryCost(formattedValue),totalTax(formattedValue, value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),user,saveTime,name,description,deliveryAddress(FULL),deliveryMode(FULL),paymentInfo(FULL),sapBillingAddress(FULL)'	stringa		defaultOccOpfCartConfig	...opfCartEndpoints (integration-libs/opf/global-functions/root/config/default-occ-opf-cart-config.ts:L18-L18)	@spartacus/opf/global-functions/root	integration-libs/opf/global-functions/root/config/default-occ-opf-cart-config.ts:L11-L11
48	cart Access Code	'users/${userId}/carts/${cartId}/accessCode'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L32-L32
49	cart Apply Voucher	'users/${userId}/carts/${cartId}/applyVoucher'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L28-L28
50	cart Guest User	'users/${userId}/carts/${cartId}/guestuser'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L33-L33
51	cart Remove Voucher	'users/${userId}/carts/${cartId}/removeVoucher'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L29-L29

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
52	carts	'users/${userId}/carts?fields=carts(DEFAULT, potentialProductPromotions,appliedProductPromotions, potentialOrderPromotions,appliedOrderPromotions, entries(totalPrice(formattedValue), product(images(FULL),stock(FULL)),base Price(formattedValue,value),updateable),total Price(formattedValue),totalItems,totalPriceWith Tax(formattedValue),totalDiscounts(value,formatted Value),subTotal(formattedValue),totalUnitCount, deliveryItemsQuantity,deliveryCost(formattedValue), totalTax(formattedValue, value),pickupItemsQuantity, net,appliedVouchers,productDiscounts(formatted Value),user,saveTime,name,description)'	stringa		default OccCart Config Factory		@spartacus /cart/ base/occ	feature-libs/ cart/base/occ/ config/default- occ-cart-config- factory.ts:L15- L16
53	carts	'users/${userId}/carts?fields=carts(DEFAULT, potentialProductPromotions,appliedProductPromotions, potentialOrderPromotions,appliedOrderPromotions, entries(totalPrice(formattedValue), product(images(FULL),stock(FULL)),arrivalSlots,base Price(formattedValue,value),updateable),total Price(formattedValue),totalItems,totalPriceWith Tax(formattedValue),totalDiscounts(value,formatted Value),subTotal(formattedValue),totalUnitCount, deliveryItemsQuantity,deliveryCost(formattedValue), totalTax(formattedValue, value),pickupItemsQuantity, net,appliedVouchers,productDiscounts(formatted Value),user,saveTime,name,description)'	stringa		default OccCart WithEdd Config	...defaultEdd OccEndpoint s (feature- libs/estimate d-delivery- date/show- estimated- delivery- date/config/d efault-occ- cart-with- edd.config.ts: L23-L23)	@spartacus / estimated- delivery- date/show- estimated- delivery- date	feature-libs/ estimated- delivery-date/ show-estimated- delivery-date/ config/default- occ-cart-with- edd.config.ts: L11-L12
54	carts	'users/${userId}/carts?fields=carts(DEFAULT,sapGift Cards(FULL,balance(value,formattedValue),remaining Balance(value,formattedValue), appliedAmount(value, formattedValue)),sapGiftCardSummary(FULL,total AppliedAmount(value,formattedValue),totalRemaining Balance(value,formattedValue), totalBalance(value, formattedValue)),potentialProductPromotions,applied ProductPromotions,potentialOrderPromotions,applied OrderPromotions,entries(totalPrice(formattedValue), product(images(FULL),stock(FULL)),base Price(formattedValue,value),updateable),total Price(formattedValue),totalItems,totalPriceWith Tax(formattedValue),totalDiscounts(value,formatted Value),subTotal(formattedValue),totalUnitCount, deliveryItemsQuantity,deliveryCost(formattedValue), totalTax(formattedValue, value),pickupItemsQuantity, net,appliedVouchers,productDiscounts(formatted Value),user,saveTime,name,description,_available Operations)'	stringa		default OccOpf GiftCard Cart Endpoint sConfig	...giftCardCar tEndpoints (integration- libs/opf/gift- card/root/con fig/default- occ-opf-gift- card-cart- config.ts:L20 -L20)	@spartacus /opf/gift- card/root	integration- libs/opf/gift- card/root/ config/default- occ-opf-gift- card-cart- config.ts:L11- L12
55	cartUser Email	'users/${userId}/carts/${cartId}?fields=sapCustomer Email'	stringa		default OccOpf Checkout Config		@spartacus /opf/ checkout/ root	integration- libs/opf/ checkout/root/ config/default- occ-opf- checkout-config. ts:L13-L14
56	cart Voucher	'users/${userId}/carts/${cartId}/vouchers'	stringa		default OccCart Config Factory		@spartacus /cart/ base/occ	feature-libs/ cart/base/occ/ config/default- occ-cart-config- factory.ts:L27- L27
57	claim Coupon	'users/${userId}/customercoupons/${couponCode}/ claim'	stringa		default OccUser Config		@spartacus /core	core-libs/core/ src/occ/ adapters/user/ default-occ- user-config.ts: L23-L23

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
58	claimCustomerCoupon	'users/${userId}/customercoupons/claim'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L24-L24
59	clearDeliveryMode	DELIVERY_MODE_ENDPOINT = 'users/${userId}/carts/${cartId}/deliverymode'	costante (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L10-L10)		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L22-L22
60	cloneSavedCart	'/users/${userId}/carts/${cartId}/clonesavedcart'	stringa		defaultOccSavedCartConfigFactory		@spartacus/cart/saved-cart/occ	feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L18-L18
61	component	'users/${userId}/cms/components/${id}'	stringa		defaultCmsModuleConfig		@spartacus/core	core-libs/core/src/cms/config/default-cms-config.ts:L13-L13
62	components	'users/${userId}/cms/components'	stringa		defaultCmsModuleConfig		@spartacus/core	core-libs/core/src/cms/config/default-cms-config.ts:L14-L14
63	consentDetail	'users/${userId}/consents/${consentId}'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L18-L18
64	consents	'users/${userId}/consents'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L17-L17
65	consentTemplates	'users/${userId}/consenttemplates'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L16-L16
66	consignmentTracking	'users/${userId}/orders/${orderCode}/consignments/${consignmentCode}/tracking'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L16-L17

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
67	copyCpqContainerRow	'cpqconfigurator/${configurationId}/rows/${rowId}/copyRow'	stringa		default OccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L34-L35
68	costCenter	'/costcenters/${costCenterCode}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L49-L49
69	costCenterBudget	'/costcenters/${costCenterCode}/budgets/${budgetCode}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L52-L53
70	costCenterBudgets	'/costcenters/${costCenterCode}/budgets'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L51-L51
71	costCenters	'/costcenters'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L48-L48
72	costCentersAll	'/costcentersall'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L50-L50
73	countries	'countries'	stringa		defaultOccSiteContextConfig		@spartacus/core	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L15-L15
74	couponNotification	'users/${userId}/customercoupons/${couponCode}/notification'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L25-L26

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
75	createCart	'users/${userId}/carts?fields=DEFAULT,potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrderPromotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),basePrice(formattedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),totalDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,deliveryCost(formattedValue),totalTax(formattedValue,value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),user'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L18-L19
76	createCart	'users/${userId}/carts?fields=DEFAULT,potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrderPromotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),arrivalSlots,basePrice(formattedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),totalDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,deliveryCost(formattedValue),totalTax(formattedValue,value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),user'	stringa		defaultOccCartWithEddConfig	...defaultEddOccEndpoints (feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L23-L23)	@spartacus/estimated-delivery-date/show-estimated-delivery-date	feature-libs/estimated-delivery-date/show-estimated-delivery-date/config/default-occ-cart-with-edd.config.ts:L14-L15
77	createCpqConfiguration	'products/${productCode}/configurators/cpqconfigurator'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L23-L24
78	createCpqContainerRow	'cpqconfigurator/${configurationId}/rows'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L33-L33
79	createDeliveryAddress	DELIVERY_ENDPOINT = 'users/${userId}/carts/${cartId}/addresses/delivery'	costante (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L9-L9)		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L18-L18
80	createPaymentDetails	'users/${userId}/carts/${cartId}/payment/sop/response'	stringa		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L27-L28

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
81	createQuote	'users/${userId}/quotes'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L17-L17
82	createTextfieldConfiguration	'products/${productCode}/configurator/textfield'	stringa		defaultOccConfiguratorTextfieldConfigFactory		@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L14-L15
83	createTicket	'users/${customerId}/tickets'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L19-L19
84	createTicketEvent	'users/${customerId}/tickets/${ticketId}/events'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L15-L15
85	createVariantConfiguration	'products/${productCode}/configurators/ccpconfigurator'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L14-L15
86	createVerificationToken	'users/anonymous/verificationToken'	stringa		defaultOccUserAccountConfig		@spartacus/user/account/occ	feature-libs/user/account/occ/adapters/config/default-occ-user-account-endpoint.config.ts:L14-L14
87	currencies	'currencies'	stringa		defaultOccSiteContextConfig		@spartacus/core	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L14-L14
88	customerCoupons	'users/${userId}/customercoupons'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L22-L22

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
89	deleteCart	'users/${userId}/carts/${cartId}'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L26-L26
90	deliveryMode	DELIVERY_MODE_ENDPOINT = 'users/${userId}/carts/${cartId}/deliverymode'	costante (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L10-L10)		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L20-L20
91	deliveryModes	'${DELIVERY_MODE_ENDPOINT}s'	stringa		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L23-L23
92	downloadAttachment	'/users/${customerId}/tickets/${ticketId}/events/${eventCode}/attachments/${attachmentId}'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L22-L23
93	downloadAttachment	'users/${userId}/quotes/${quoteCode}/attachments/${attachmentId}'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L29-L30
94	downloadOrderAttachment	'users/${userId}/orders/${orderId}/attachments/${attachmentId}/download'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L44-L45
95	editQuote	'users/${userId}/quotes/${quoteCode}'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L23-L23
96	extendSubscription	'users/${userId}/subscriptions/${subscriptionCode}/extension'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L16-L17

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
97	extensionEffectiveDate	'users/${userId}/subscriptions/${subscriptionCode}/extensionEffectiveAt'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L14-L15
98	futureStock	'users/${userId}/futureStocks/${productCode}'	stringa		defaultOccFutureStockConfig	...futureStockEndpoints (feature-libs/product/future-stock/occ/config/default-occ-future-stock.config.ts:L19-L19)	@spartacus/product/future-stock/occ	feature-libs/product/future-stock/occ/config/default-occ-future-stock.config.ts:L11-L11
99	futureStocks	'users/${userId}/futureStocks'	stringa		defaultOccFutureStockConfig	...futureStockEndpoints (feature-libs/product/future-stock/occ/config/default-occ-future-stock.config.ts:L19-L19)	@spartacus/product/future-stock/occ	feature-libs/product/future-stock/occ/config/default-occ-future-stock.config.ts:L12-L12
100	getActiveCostCenters	'/costcenters?fields=DEFAULT,unit(BASIC,addresses(DEFAULT))'	stringa		defaultOccCostCentersConfig		@spartacus/core	core-libs/core/src/occ/adapters/cost-center/default-occ-cost-centers-config.ts:L13-L14
101	getCheckoutDetails	'users/${userId}/carts/${cartId}?fields=deliveryAddress(FULL),deliveryMode(FULL),paymentInfo(FULL),costCenter(FULL),purchaseOrderNumber,paymentType(FULL)'	stringa		defaultOccCheckoutB2BConfig	...defaultB2bCheckoutDetailsOccEndpoint (feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L20-L20)	@spartacus/checkout/b2b/occ	feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L12-L13
102	getCheckoutDetails	'users/${userId}/carts/${cartId}?fields=deliveryAddress(FULL),deliveryMode(FULL),paymentInfo(FULL)'	stringa		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L29-L30

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
103	getCheckoutDetails	'users/${userId}/carts/${cartId}?fields=deliveryAddress(FULL),deliveryMode(FULL),paymentInfo(FULL),costCenter(FULL),purchaseOrderNumber,paymentType(FULL),servicedAt'	stringa		defaultOccCheckoutServiceOrderConfig	...defaultServiceOrderCheckoutDetailsOccEndpoint (integration-libs/s4-service/checkout/occ/config/default-occ-checkout-s4-service-config.ts:L19-L19)	@spartacus/s4-service/checkout	integration-libs/s4-service/checkout/occ/config/default-occ-checkout-s4-service-config.ts:L11-L12
104	getCpqAccessData	'users/${userId}/access/cpqconfigurator'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L14-L14
105	getEffectiveCancellationDate	'users/${userId}/subscriptions/${subscriptionCode}/cancellationEffectiveAt'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L18-L19
106	getOrderCode	'users/${userId}/quotes/${quoteCode}?fields=sapOrderCode'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L22-L22
107	getProductInterests	'users/${userId}/productinterests?fields=sorts,pagination,results(productInterestEntry,product(code))'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L29-L30
108	getQuote	'users/${userId}/quotes/${quoteCode}?fields=FULL,expirationTime' + PRICE_HEADER_FIELDS + ',entries(FULL)'	espressione		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L18-L21
109	getQuotes	'users/${userId}/quotes'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L16-L16

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
110	getTicket	'users/${customerId}/tickets/${ticketId}'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L13-L13
111	getTicketAssociatedObjects	'users/${customerId}/ticketAssociatedObjects'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L17-L18
112	getTicketCategories	'/ticketCategories'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L16-L16
113	getTickets	'users/${customerId}/tickets'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L14-L14
114	getVariantConfigurationOverview	'ccpconfigurator/${configId}/configurationOverview'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L42-L43
115	getWishlistEntries	'users/${userId}/wishlists/${wishlistId}/entries'	stringa		defaultOccUserWishlistConfig		@spartacus/user/wishlist/occ	feature-libs/user/wishlist/occ/adapters/config/default-occ-user-wishlist-endpoint.config.ts:L14-L14
116	getWishlists	'users/${userId}/wishlists'	stringa		defaultOccUserWishlistConfig		@spartacus/user/wishlist/occ	feature-libs/user/wishlist/occ/adapters/config/default-occ-user-wishlist-endpoint.config.ts:L13-L13
117	languages	'languages'	stringa		defaultOccSiteContextConfig		@spartacus/core	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L13-L13

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
118	notificationPreference	'users/${userId}/notificationpreferences'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L27-L27
119	orderApproval	'/users/${userId}/orderapprovals/${orderApprovalCode}?fields=FULL'	stringa		defaultOccOrderApprovalConfig		@spartacus/organization/order-approval	feature-libs/organization/order-approval/occ/config/default-occ-organization-config.ts:L14-L15
120	orderApprovalDecision	'/users/${userId}/orderapprovals/${orderApprovalCode}/decision'	stringa		defaultOccOrderApprovalConfig		@spartacus/organization/order-approval	feature-libs/organization/order-approval/occ/config/default-occ-organization-config.ts:L16-L17
121	orderApprovalPermissionTypes	'/orderApprovalPermissionTypes'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L57-L57
122	orderApprovals	'/users/${userId}/orderapprovals'	stringa		defaultOccOrderApprovalConfig		@spartacus/organization/order-approval	feature-libs/organization/order-approval/occ/config/default-occ-organization-config.ts:L13-L13
123	orderAttachments	'users/${userId}/orders/${orderId}/attachments'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L43-L43
124	orderDetail	'users/${userId}/orders/${orderId}?fields=FULL'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L14-L14
125	orderDetail	'users/${userId}/orders/${orderId}?fields=FULL,sapGiftCardSummary(FULL,totalAppliedAmount(value,formattedValue),totalRemainingBalance(value,formattedValue),totalBalance(value,formattedValue))'	stringa		defaultOccOpfGiftCardOrderEndpointsConfig	...giftCardOrderEndpoints (integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-order-config.ts:L21-L21)	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-order-config.ts:L11-L12

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
126	orderHistory	'users/${userId}/orders'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L13-L13
127	orderReturnDetail	'users/${userId}/orderReturns/${returnRequestCode}?fields=BASIC,returnEntries(BASIC,refundAmount(formattedValue),orderEntry(basePrice(formattedValue),product(name,code,baseOptions,images(DEFAULT,galleryIndex)))),deliveryCost(formattedValue),totalPrice(formattedValue),subTotal(formattedValue)'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L22-L23
128	orderReturns	'users/${userId}/orderReturns?fields=BASIC'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L21-L21
129	organizationUserRegistration	'/orgUsers'	stringa		defaultOccOrganizationUserRegistrationConfig		@spartacus/organization/user-registration/occ	feature-libs/organization/user-registration/occ/config/default-occ-organization-config.ts:L13-L13
130	orgUnit	'/users/${userId}/orgUnits/${orgUnitId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L20-L20
131	orgUnitApprover	'/users/${userId}/orgUnits/${orgUnitId}/orgCustomers/${orgCustomerId}/roles/${roleId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L25-L26
132	orgUnitApprovers	'/users/${userId}/orgUnits/${orgUnitId}/orgCustomers/${orgCustomerId}/roles'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L23-L24
133	orgUnits	'/users/${userId}/orgUnits'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L19-L19

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
134	orgUnitAddress	'/users/${userId}/orgUnits/${orgUnitId}/addresses/${addressId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L32-L33
135	orgUnitAddresses	'/users/${userId}/orgUnits/${orgUnitId}/addresses'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L31-L31
136	orgUnitApprovalProcesses	'/users/${userId}/orgUnitsAvailableApprovalProcesses'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L17-L18
137	orgUnitAvailable	'/users/${userId}/availableOrgUnitNodes'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L15-L15
138	orgUnitTree	'/users/${userId}/orgUnitsRootNodeTree'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L16-L16
139	orgUnitUserRole	'/users/${userId}/orgCustomers/${orgCustomerId}/roles/${roleId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L29-L30
140	orgUnitUserRoles	'/users/${userId}/orgCustomers/${orgCustomerId}/roles'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L27-L28
141	orgUnitUsers	'/users/${userId}/orgUnits/${orgUnitId}/availableUsers/${roleId}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L21-L22

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
142	page	'users/${userId}/cms/pages/${id}'	stringa		defaultCmsModuleConfig		@spartacus/core	core-libs/core/src/cms/config/default-cms-config.ts:L16-L16
143	pages	'users/${userId}/cms/pages'	stringa		defaultCmsModuleConfig		@spartacus/core	core-libs/core/src/cms/config/default-cms-config.ts:L15-L15
144	paymentDetail	'users/${userId}/paymentdetails/${paymentDetailId}'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L14-L14
145	paymentDetails	'users/${userId}/carts/${cartId}/payment/digitalPayments/response'	stringa		occDigitalPaymentsConfig	...occDigitalPaymentsEndpoints (integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L21-L21)	@spartacus/digital-payments	integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L14-L15
146	paymentDetailsAll	'users/${userId}/paymentdetails'	stringa		defaultOccUserConfig		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L13-L13
147	paymentProviderSubInfo	'users/${userId}/carts/${cartId}/payment/sop/request?responseUrl=sampleUrl'	stringa		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L25-L26
148	paymentRequest	'users/${userId}/carts/${cartId}/payment/digitalPayments/request'	stringa		occDigitalPaymentsConfig	...occDigitalPaymentsEndpoints (integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L21-L21)	@spartacus/digital-payments	integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.config.ts:L12-L13
149	paymentTypes	'paymenttypes'	stringa		defaultOccCheckoutB2BConfig		@spartacus/checkout/b2b/occ	feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L23-L23

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
150	pdfInvoicesDownloadInvoicePDF	'users/${userId}/orders/${orderId}/invoices/${invoiceId}/download'	stringa		defaultOccPDFInvoicesConfig		@spartacus/pdf-invoices/occ	feature-libs/pdf-invoices/occ/config/default-occ-pdf-invoices-config.ts:L14-L15
151	pdfInvoicesListInvoices	'users/${userId}/orders/${orderId}/invoices'	stringa		defaultOccPDFInvoicesConfig		@spartacus/pdf-invoices/occ	feature-libs/pdf-invoices/occ/config/default-occ-pdf-invoices-config.ts:L13-L13
152	performQuoteAction	'users/${userId}/quotes/${quoteCode}/action'	stringa		defaultOccQuoteConfig		@spartacus/quote/occ	feature-libs/quote/occ/config/default-occ-quote-config.ts:L24-L24
153	permission	'/users/${userId}/orderApprovalPermissions/${orderApprovalPermissionCode}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L55-L56
154	permissions	'/users/${userId}/orderApprovalPermissions'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L54-L54
155	placeOrder	'orgUsers/${userId}/orders?fields=FULL'	stringa		defaultB2bOccConfig	...defaultB2bOrderOccEndpoints (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L45-L45)	@spartacus/setup	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L32-L32
156	placeOrder	'users/${userId}/orders?fields=FULL'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L37-L37
157	placePaymentAuthorizedOrder	'users/${userId}/orders/paymentAuthorizedOrderPlacement?fields=FULL'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L38-L39

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
158	placePaymentAuthorizedOrder	'orgUsers/${userId}/orders?fields=FULL'	stringa		defaultOpfB2bCheckoutOccEndpointsConfig	...orderEndpoints (integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-occ-endpoints-config.ts:L18-L18)	@spartacus/opf/b2b-checkout/root	integration-libs/opf/b2b-checkout/root/config/default-opf-b2b-checkout-occ-endpoints-config.ts:L11-L11
159	placePaymentAuthorizedOrder	'users/${userId}/orders/paymentAuthorizedOrderPlacement?fields=FULL,sapGiftCardSummary(FULL,totalAppliedAmount(value,formattedValue),totalRemainingBalance(value,formattedValue),totalBalance(value,formattedValue))'	stringa		defaultOccOpfGiftCardOrderEndpointsConfig	...giftCardOrderEndpoints (integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-order-config.ts:L21-L21)	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/config/default-occ-opf-gift-card-order-config.ts:L13-L14
160	product	{ default: 'products/${productCode}?fields=DEFAULT,averageRating,images(FULL),classifications,manufacturer,numberOfReviews,categories(FULL),baseOptions,baseProduct,variantOptions,variantType', list: 'products/${productCode}?fields=code,purchasable,name,summary,price(formattedValue),images(DEFAULT,galleryIndex),baseProduct', details: 'products/${productCode}?fields=averageRating,stock(DEFAULT),description,availableForPickup,code,url,price(DEFAULT),numberOfReviews,manufacturer,categories(FULL),priceRange,multidimensional,tags,images(FULL)', promotions: 'products/${productCode}?fields=potentialPromotions(description)', attributes: 'products/${productCode}?fields=classifications', price: 'products/${productCode}?fields=price(formattedValue)', stock: 'products/${productCode}?fields=stock(DEFAULT)', unit: 'products/${productCode}?fields=sapUnit', list_item: 'products/${productCode}?fields=code,name,price(formattedValue),images(DEFAULT),baseProduct' }	oggetto (scope)	default, list, details, promotions, attributes, price, stock, unit, list_item	defaultOccProductConfig		@spartacus/core	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L14-L28
161	product	{ configurator: 'products/${productCode}?fields=code,configurable,configuratorType', configuratorProductCard: 'products/${productCode}?fields=code,name,description,images(DEFAULT)' }	oggetto (scope)	configurator, configuratorProductCard	defaultOccConfiguratorProductConfig		@spartacus/product-configurator/common	feature-libs/product-configurator/common/occ/default-occ-configurator-product-config.ts:L14-L19
162	product	{ multi_dimensional: 'products/${productCode}?fields=multidimensional,categories,' + variantMatrix, multi_dimensional_availability: 'products/${productCode}?fields=variantOptions(stockLevel),code),purchasable' }	oggetto (scope)	multi_dimensional, multi_dimensional_availability	defaultOccProductMultiDimensionalSelectorConfig		@spartacus/product-multi-dimensional/selector/occ	feature-libs/product-multi-dimensional/selector/occ/config/default-occ-product-multi-dimensional-selector-config.ts:L16-L22
163	product	{ bulkPrices: 'orgProducts/${productCode}?fields=price(DEFAULT),volumePrices(FULL)' }	oggetto (scope)	bulkPrices	defaultOccBulkPricingConfig		@spartacus/product/bulk-pricing/occ	feature-libs/product/bulk-pricing/occ/config/default-occ-bulk-pricing-config.ts:L13-L16

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
164	product	{ variants: 'products/${productCode}?fields=name, purchasable,baseOptions(DEFAULT),baseProduct,variantOptions(DEFAULT),variantType' }	oggetto (scope)	variants	default Occ Product Variants Config		@spartacus/product/variants/occ	feature-libs/product/variants/occ/config/default-occ-product-variants-config.ts:L13-L16
165	product	{ subscription: 'products/${productCode}?fields=sapPricePlan,sapSubscriptionTerm' }	oggetto (scope)	subscription	default Occ Subscription Billing Config		@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L36-L39
166	product Availabilities	'productAvailabilities?filters=${productCode}:${unitSapCode}'	stringa		default Occ Product Config		@spartacus/core	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L29-L30
167	product Interests	'users/${userId}/productinterests'	stringa		default OccUser Config		@spartacus/core	core-libs/core/src/occ/adapters/user/default-occ-user-config.ts:L28-L28
168	product References	'products/${productCode}/references?fields=DEFAULT,references(target(images(FULL)))'	stringa		default Occ Product Config		@spartacus/core	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L35-L36
169	product Reviews	'products/${productCode}/reviews'	stringa		default Occ Product Config		@spartacus/core	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L31-L31
170	product Search	{ default: 'products/search?fields=products(code,name,summary,configurable,configuratorType,multidimensional,price(FULL),images(DEFAULT),stock(FULL),averageRating,variantOptions,baseProduct,priceRange(maxPrice(formattedValue),minPrice(formattedValue))),facets,breadcrumbs,pagination(DEFAULT),sorts(DEFAULT),freeTextSearch,currentQuery,keywordRedirectUrl', carousel: 'products/search?fields=products(code,purchasable,name,summary,price(formattedValue),stock(DEFAULT),images(DEFAULT,galleryIndex),baseProduct)', carouselMinimal: 'products/search?fields=products(code,name,price(formattedValue),images(DEFAULT),baseProduct)' }	oggetto (scope)	default, carousel, carouselMinimal	default Occ Product Config		@spartacus/core	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L37-L44
171	product SearchByCategory	{ default: 'categories/${categoryCode}/products?fields=DEFAULT', code: 'categories/${categoryCode}/products?fields=products(code)' }	oggetto (scope)	default, code	default Occ Product Config		@spartacus/core	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L45-L48

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
172	productSuggestions	'products/suggestions'	stringa		defaultOccProductConfig		@spartacus/core	core-libs/core/src/occ/adapters/product/default-occ-product-config.ts:L49-L49
173	punchoutSession	'punchout/sessions/${sessionId}'	stringa		defaultOccPunchoutConfig	...punchoutOccEndpoints (integration-libs/punchout/occ/config/default-occ-punchout-config.ts:L19-L19)	@spartacus/punchout/occ	integration-libs/punchout/occ/config/default-occ-punchout-config.ts:L11-L11
174	punchoutSessionRequisition	'punchout/sessions/${sessionId}/requisition'	stringa		defaultOccPunchoutConfig	...punchoutOccEndpoints (integration-libs/punchout/occ/config/default-occ-punchout-config.ts:L19-L19)	@spartacus/punchout/occ	integration-libs/punchout/occ/config/default-occ-punchout-config.ts:L12-L12
175	quickBuyCreateDeliveryAddress	'users/${userId}/carts/${cartId}/addresses/delivery'	stringa		defaultOccOpfQuickBuyCartConfig		@spartacus/opf/quick-buy/occ	integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts:L13-L14
176	quickBuyDeliveryModes	'users/${userId}/carts/${cartId}/deliverymodes'	stringa		defaultOccOpfQuickBuyCartConfig		@spartacus/opf/quick-buy/occ	integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts:L17-L17
177	quickBuySelectedDeliveryMode	'users/${userId}/carts/${cartId}?fields=deliveryMode(FULL)'	stringa		defaultOccOpfQuickBuyCartConfig		@spartacus/opf/quick-buy/occ	integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts:L19-L20
178	quickBuySetBillingAddress	'users/${userId}/carts/${cartId}/addresses/billing'	stringa		defaultOccOpfQuickBuyCartConfig		@spartacus/opf/quick-buy/occ	integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts:L15-L16
179	quickBuySetDeliveryMode	'users/${userId}/carts/${cartId}/deliverymode'	stringa		defaultOccOpfQuickBuyCartConfig		@spartacus/opf/quick-buy/occ	integration-libs/opf/quick-buy/occ/config/default-occ-opf-quick-buy-cart-config.ts:L18-L18

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
180	quoteCode	'users/${userId}/orders/${orderId}?fields=sapQuoteCode'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L15-L15
181	readCpqConfiguration	'cpqconfigurator/${configurationId}/configuration?tabId=${tabId}'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L25-L26
182	readCpqConfigurationForCartEntry	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/cpqconfigurator'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L17-L18
183	readCpqConfigurationForCartEntryFull	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/cpqconfigurator/configuration'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L38-L39
184	readCpqConfigurationForOrderEntry	'users/${userId}/orders/${orderId}/entries/${orderEntryNumber}/cpqconfigurator'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L19-L20
185	readCpqConfigurationForOrderEntryFull	'users/${userId}/orders/${orderId}/entries/${orderEntryNumber}/cpqconfigurator/configuration'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L40-L41
186	readCpqConfigurationForQuoteEntryFull	'users/${userId}/quotes/${quoteId}/entries/${quoteEntryNumber}/cpqconfigurator/configuration'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L42-L43
187	readCpqConfigurationForSavedCartEntryFull	'users/${userId}/savedCarts/${savedCartId}/entries/${entryNumber}/cpqconfigurator/configuration'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L44-L45

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
188	readCpqConfigurationOverview	'cpqconfigurator/${configurationId}/configurationOverview'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L27-L28
189	readTextfieldConfigurationForCartEntry	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/configurator/textfield'	stringa		defaultOccConfiguratorTextfieldConfigFactory		@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L20-L21
190	readTextfieldConfigurationForOrderEntry	'users/${userId}/orders/${orderId}/entries/${orderEntryNumber}/configurator/textfield'	stringa		defaultOccConfiguratorTextfieldConfigFactory		@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L22-L23
191	readTextfieldConfigurationForQuoteEntry	'users/${userId}/quotes/${quoteId}/entries/${quoteEntryNumber}/configurator/textfield'	stringa		defaultOccConfiguratorTextfieldConfigFactory		@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L24-L25
192	readVariantConfiguration	'ccpconfigurator/${configId}'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L17-L17
193	readVariantConfigurationForCartEntry	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/ccpconfigurator'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L24-L25
194	readVariantConfigurationOverviewForOrderEntry	'users/${userId}/orders/${orderId}/entries/${orderEntryNumber}/ccpconfigurator/configurationOverview'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L30-L31

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
195	readVariantConfigurationOverviewForQuoteEntry	'users/${userId}/quotes/${quoteId}/entries/${quoteEntryNumber}/ccpconfigurator/configurationOverview'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L36-L37
196	readVariantConfigurationOverviewForSavedCartEntry	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/ccpconfigurator/configurationOverview'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L33-L34
197	readVariantConfigurationPriceSummary	'ccpconfigurator/${configId}/pricing'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L39-L40
198	regions	'countries/${isoCode}/regions?fields=regions(name,isocode,isocodeShort)'	stringa		defaultOccSiteContextConfig		@spartacus/core	core-libs/core/src/occ/adapters/site-context/default-occ-site-context-config.ts:L16-L17
199	removeCpqContainerRow	'cpqconfigurator/${configurationId}/rows/${rowId}'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L36-L37
200	removeDeliveryAddress	DELIVERY_ENDPOINT = 'users/${userId}/carts/${cartId}/addresses/delivery'	costante (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L9-L9)		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L19-L19
201	removeEntries	'users/${userId}/carts/${cartId}/entries/${entryNumber}'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L23-L24

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
2002	removeGiftCard	'users/${userId}/carts/${cartId}/giftCards/${giftCardId}'	stringa		defaultGiftCardOccEndpointsConfig		@spartacus/opf/gift-card/occ	integration-libs/opf/gift-card/occ/config/default-occ-opf-gift-card-config.ts:L14-L15
2003	removeWishlistEntry	'users/${userId}/wishlists/${wishlistId}/entries/${entryId}'	stringa		defaultOccUserWishlistConfig		@spartacus/user/wishlist/occ	feature-libs/user/wishlist/occ/adapters/config/default-occ-user-wishlist-endpoint.config.ts:L16-L17
2004	reorder	'orgUsers/${userId}/cartFromOrder?orderCode=${orderCode}'	stringa		defaultB2bOccConfig	...defaultB2bOrderOccEndpoints (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L45-L45)	@spartacus/setup	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L35-L35
2005	replenishmentOrderDetails	'users/${userId}/replenishmentOrders/${replenishmentOrderCode}?fields=FULL,costCenter(FULL),purchaseOrderNumber,paymentType,user'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L26-L27
2006	replenishmentOrderDetailsHistory	'users/${userId}/replenishmentOrders/${replenishmentOrderCode}/orders'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L28-L29
2007	replenishmentOrderHistory	'users/${userId}/replenishmentOrders?fields=FULL, replenishmentOrders(FULL, purchaseOrderNumber)'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L32-L33
2008	requestedDeliveryDate	'users/${userId}/carts/${cartId}/requestedretrievaldate'	stringa		defaultOccRequestedDeliveryDateConfig		@spartacus/requested-delivery-date/occ	feature-libs/requested-delivery-date/occ/config/default-occ-requested-delivery-date-config.ts:L13-L14
2009	rescheduleService	'users/${userId}/orders/${code}/serviceOrder/serviceScheduleSlot'	stringa		defaultOccServiceOrderConfig	...s4ServiceOccEndpoints (integration-libs/s4-service/order/occ/config/default-occ-s4-service-config.ts:L21-L21)	@spartacus/s4-service/order	integration-libs/s4-service/order/occ/config/default-occ-s4-service-config.ts:L13-L14

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
210	restoreSavedCart	'/users/${userId}/carts/${cartId}/restoresavedcart'	stringa		defaultOccSavedCartConfigFactory		@spartacus/cart/saved-cart/occ	feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L17-L17
211	returnOrder	'users/${userId}/orderReturns?fields=BASIC,returnEntries(BASIC,refundAmount(formattedValue),orderEntry(basePrice(formattedValue),product(name,code,baseOptions,images(DEFAULT,galleryIndex)))),deliveryCost(formattedValue),totalPrice(formattedValue),subTotal(formattedValue)'	stringa		defaultOccOrderConfig		@spartacus/order/occ	feature-libs/order/occ/config/default-occ-order-config.ts:L19-L20
212	reverseCancellation	'users/${userId}/subscriptions/${subscriptionCode}/cancellationReversal'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L24-L25
213	saveCart	'/users/${userId}/carts/${cartId}/save'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L30-L30
214	savedCart	'/users/${userId}/carts/${cartId}/savedcart'	stringa		defaultOccSavedCartConfigFactory		@spartacus/cart/saved-cart/occ	feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L16-L16
215	savedCarts	'/users/${userId}/carts?savedCartsOnly=true&fields=carts(DEFAULT,potentialProductPromotions,appliedProductPromotions,potentialOrderPromotions,appliedOrderPromotions,entries(totalPrice(formattedValue),product(images(FULL),stock(FULL)),basePrice(formattedValue,value),updateable),totalPrice(formattedValue),totalItems,totalPriceWithTax(formattedValue),totalDiscounts(value,formattedValue),subTotal(formattedValue),totalUnitCount,deliveryItemsQuantity,deliveryCost(formattedValue),totalTax(formattedValue, value),pickupItemsQuantity,net,appliedVouchers,productDiscounts(formattedValue),saveTime,user,name,description)'	stringa		defaultOccSavedCartConfigFactory		@spartacus/cart/saved-cart/occ	feature-libs/cart/saved-cart/occ/config/default-occ-saved-cart-config-factory.ts:L14-L15
216	scheduleReplenishmentOrder	'orgUsers/${userId}/replenishmentOrders?fields=FULL,costCenter(FULL),purchaseOrderNumber,paymentType'	stringa		defaultB2bOccConfig	...defaultB2bOrderOccEndpoints (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L45-L45)	@spartacus/setup	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L33-L34

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
217	searchConfigurator Variants	'ccpconfigurator/${configId}/variants'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L45-L45
218	setBillingAddress	'users/${userId}/carts/${cartId}/addresses/billing'	stringa		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L31-L31
219	setCartCostCenter	'users/${userId}/carts/${cartId}/costcenter'	stringa		defaultOccCheckoutB2BConfig		@spartacus/checkout/b2b/occ	feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L24-L24
220	setCartPaymentDetails	'users/${userId}/carts/${cartId}/paymentdetails'	stringa		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L24-L24
221	setCartPaymentOption	'users/${userId}/carts/${cartId}/paymentOption'	stringa		defaultOccOpfPaymentConfig		@spartacus/opf/payment/occ	integration-libs/opf/payment/occ/config/default-occ-opf-payment-config.ts:L13-L13
222	setCartPaymentType	'users/${userId}/carts/${cartId}/paymenttype'	stringa		defaultOccCheckoutB2BConfig		@spartacus/checkout/b2b/occ	feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L25-L25
223	setDeliveryAddress	'orgUsers/${userId}/carts/${cartId}/addresses/delivery'	stringa		defaultOccCheckoutB2BConfig		@spartacus/checkout/b2b/occ	feature-libs/checkout/b2b/occ/config/default-occ-checkout-b2b-config.ts:L21-L22
224	setDeliveryAddress	DELIVERY_ENDPOINT = 'users/${userId}/carts/${cartId}/addresses/delivery'	costante (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L9-L9)		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L16-L16

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
225	setDeliveryMode	DELIVERY_MODE_ENDPOINT = 'users/${userId}/carts/${cartId}/deliverymode'	costante (feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L10-L10)		defaultOccCheckoutConfig		@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/config/default-occ-checkout-config.ts:L21-L21
226	setServiceScheduleSlot	'users/${userId}/carts/${cartId}/serviceOrder/serviceScheduleSlot'	stringa		defaultOccCheckoutServiceOrderConfig		@spartacus/s4-service/checkout	integration-libs/s4-service/checkout/occ/config/default-occ-checkout-s4-service-config.ts:L20-L21
227	stock	'products/${productCode}/stock'	stringa		defaultOccStockConfig		@spartacus/pickup-in-store/occ	feature-libs/pickup-in-store/occ/adapters/default-occ-stock-config.ts:L16-L16
228	stockAtStore	'products/${productCode}/stock/${storeName}'	stringa		defaultOccStockConfig		@spartacus/pickup-in-store/occ	feature-libs/pickup-in-store/occ/adapters/default-occ-stock-config.ts:L17-L17
229	store	'stores/${storeId}?fields=FULL'	stringa		defaultOccStoreFinderConfig		@spartacus/storefinder/occ	feature-libs/storefinder/occ/adapters/default-occ-store-finder-config.ts:L13-L13
230	storeDetails	'stores/${storeName}'	stringa		defaultOccPickupLocationConfig		@spartacus/pickup-in-store/occ	feature-libs/pickup-in-store/occ/adapters/default-occ-pickup-location-config.ts:L16-L16
231	stores	'stores?fields=stores(name,displayName,formattedDistance,openingHours(weekDayOpeningList(FULL),specialDayOpeningList(FULL)),geoPoint(latitude,longitude),address(line1,line2,town,region(FULL),postalCode,phone,country,email), features),pagination(DEFAULT),sorts(DEFAULT)'	stringa		defaultOccStoreFinderConfig		@spartacus/storefinder/occ	feature-libs/storefinder/occ/adapters/default-occ-store-finder-config.ts:L14-L15
232	storescounts	'stores/storescounts'	stringa		defaultOccStoreFinderConfig		@spartacus/storefinder/occ	feature-libs/storefinder/occ/adapters/default-occ-store-finder-config.ts:L16-L16

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
233	subscriptionBillByCode	'users/${userId}/subscriptionbills/${billId}?fields=FULL'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L27-L28
234	subscriptionBillsList	'users/${userId}/subscriptionbills?fields=FULL'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L26-L26
235	subscriptionByCode	'users/${userId}/subscriptions/${subscriptionCode}?fields=FULL'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L12-L13
236	subscriptionList	'users/${userId}/subscriptions'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L11-L11
237	subsequentDocuments	'users/${userId}/orders/${orderId}/subsequentDocuments'	stringa		defaultOccDocumentFlowConfigFactory	...documentFlowOccEndpoints (feature-libs/order/document-flow/occ/config/default-occ-document-flow-config-factory.ts:L20-L20)	@spartacus/order/document-flow/occ	feature-libs/order/document-flow/occ/config/default-occ-document-flow-config-factory.ts:L11-L11

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
238	subsequentDocumentsEntries	'users/${userId}/orders/${orderId}/subsequentDocuments/${documentCategory}/${documentId}/entries'	stringa		defaultOccDocumentFlowConfigFactory	...documentFlowOccEndpoints (feature-libs/order/document-flow/occ/config/default-occ-document-flow-config-factory.ts:L20-L20)	@spartacus/order/document-flow/occ	feature-libs/order/document-flow/occ/config/default-occ-document-flow-config-factory.ts:L12-L13
239	titles	'titles'	stringa		defaultOccUserProfileConfig		@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L19-L19
240	unitLevelOrderDetail	'orgUsers/\${userId}/orgUnits/orders/\${orderId}?fields=FULL'	stringa		defaultOccUnitOrderConfig		@spartacus/organization/unit-order/occ	feature-libs/organization/unit-order/occ/config/default-occ-organization-config.ts:L14-L14
241	unitLevelOrderHistory	'/orgUsers/\${userId}/orgUnits/orders'	stringa		defaultOccUnitOrderConfig		@spartacus/organization/unit-order/occ	feature-libs/organization/unit-order/occ/config/default-occ-organization-config.ts:L13-L13
242	updateCpqAttribute	'cpqconfigurator/${configurationId}/attributes/${attributeCode}?tabId=${tabId}'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L29-L30
243	updateCpqAttributeValueQuantity	'cpqconfigurator/${configurationId}/attributes/${attributeCode}/values/${attributeValueId}?tabId=${tabId}'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L31-L32
244	updateCpqConfigurationForCartEntry	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/cpqconfigurator'	stringa		defaultOccCpqConfiguratorConfigFactory		@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/default-occ-configurator-cpq-config.ts:L21-L22

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
245	updateEntries	'users/${userId}/carts/${cartId}/entries/${entryNumber}'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L21-L22
246	updateTextfieldConfigurationForCartEntry	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/configurator/textfield'	stringa		defaultOccConfiguratorTextfieldConfigFactory		@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/occ/default-occ-configurator-textfield-config.ts:L26-L27
247	updateVariantConfiguration	'ccpconfigurator/${configId}'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L19-L19
248	updateVariantConfigurationForCartEntry	'users/${userId}/carts/${cartId}/entries/${cartEntryNumber}/ccpconfigurator'	stringa		defaultOccVariantConfiguratorConfigFactory		@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/variant/default-occ-configurator-variant-config.ts:L27-L28
249	uploadAttachment	'/users/${customerId}/tickets/${ticketId}/events/${eventCode}/attachments'	stringa		defaultOccCustomerTicketingConfig		@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/config/default-occ-customer-ticketing-config.ts:L20-L21
250	user	'orgUsers/${userId}'	stringa		defaultB2bOccConfig	...defaultB2bUserAccountOccEndpoints (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L42-L42)	@spartacus/setup	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L19-L19
251	user	'users/${userId}'	stringa		defaultOccUserAccountConfig		@spartacus/user/account/occ	feature-libs/user/account/occ/adapters/config/default-occ-user-account-endpoint.config.ts:L13-L13

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
252	user Close Account	'users/${userId}'	stringa		default B2bOcc Config	...defaultB2bUserProfileOccEndpoints (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L43-L43)	@spartacus/setup	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L24-L24
253	user Forgot Password	'forgottenpasswordtokens'	stringa		default OccUser Profile Config		@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L14-L14
254	user Group	'/users/${userId}/orgUnitUserGroups/${userGroupId}'	stringa		default Occ Organization Config		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L35-L35
255	user Group AvailableOrder ApprovalPermissions	'/users/${userId}/orgUnitUserGroups/${userGroupId}/availableOrderApprovalPermissions'	stringa		default Occ Organization Config		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L36-L37
256	user Group AvailableOrg Customers	'/users/${userId}/orgUnitUserGroups/${userGroupId}/availableOrgCustomers'	stringa		default Occ Organization Config		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L38-L39
257	user Group Member	'/users/${userId}/orgUnitUserGroups/${userGroupId}/members/${orgCustomerId}'	stringa		default Occ Organization Config		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L42-L43
258	user Group Members	'/users/${userId}/orgUnitUserGroups/${userGroupId}/members'	stringa		default Occ Organization Config		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L40-L41

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
259	userGroupOrderApprovalPermission	'/users/${userId}/orgUnitUserGroups/${userGroupId}/orderApprovalPermissions/${orderApprovalPermissionCode}'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L46-L47
260	userGroupOrderApprovalPermissions	'/users/${userId}/orgUnitUserGroups/${userGroupId}/orderApprovalPermissions'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L44-L45
261	userGroups	'/users/${userId}/orgUnitUserGroups'	stringa		defaultOccOrganizationConfig		@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/config/default-occ-organization-config.ts:L34-L34
262	userRegister	'users'	stringa		defaultOccUserProfileConfig		@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L13-L13
263	userResetPassword	'resetpassword'	stringa		defaultOccUserProfileConfig		@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L16-L16
264	userRestoreToken	'passwordRestoreToken'	stringa		defaultOccUserProfileConfig		@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L15-L15
265	userUpdateLoginId	'users/${userId}/login'	stringa		defaultOccUserProfileConfig		@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L17-L17
266	userUpdatePassword	'users/${userId}/password'	stringa		defaultOccUserProfileConfig		@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/default-occ-user-profile-endpoint.config.ts:L18-L18

#	CHIAVE	TEMPLATE URL / VALORE	TIPO	SCOPE	CONFIG	VIA SPREAD	LIBRERIA	RIFERIMENTO
26	user Update Profile	'users/${userId}'	stringa		defaultB2bOccConfig	...defaultB2bUserProfileOccEndpoints (core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L43-L43)	@spartacus/setup	core-libs/setup/recipes/b2b/config/default-b2b-occ-config.ts:L23-L23
268	validate	'users/${userId}/carts/${cartId}/validate?fields=DEFAULT'	stringa		defaultOccCartConfigFactory		@spartacus/cart/base/occ	feature-libs/cart/base/occ/config/default-occ-cart-config-factory.ts:L31-L31
269	withdrawSubscription	'users/${userId}/subscriptions/${subscriptionCode}/withdrawal'	stringa		defaultOccSubscriptionBillingConfig	...subscriptionBillingEndpoints (feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L35-L35)	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/config/default-occ-subscription-billing-config.ts:L22-L23

Tabella 2 — Altri blocchi `endpoints` (non `backend.occ`)

#	PERCORSO	CHIAVE	VALORE	CONFIG	LIBRERIA	RIFERIMENTO
1	backend.opfApi.endpoints	getActiveConfigurations	'active-configurations'	defaultOpfApiBaseConfig	@spartacus/opf/base/opf-api	integration-libs/opf/base/opf-api/config/default-opf-api-base-config.ts:L13-L13
2	backend.opfApi.endpoints	getAfterRedirectScripts	'payments/${paymentSessionId}/after-redirect-scripts'	defaultOpfApiPaymentConfig	@spartacus/opf/payment/opf-api	integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L16-L17
3	backend.opfApi.endpoints	getApplePayWebSession	'payments/apple-pay-web-sessions'	defaultOpfApiQuickBuyConfig	@spartacus/opf/quick-buy/opf-api	integration-libs/opf/quick-buy/opf-api/config/default-opf-api-quick-buy-config.ts:L13-L13
4	backend.opfApi.endpoints	getCtaScripts	'payments/cta-scripts-rendering'	defaultOpfApiCtaConfig	@spartacus/opf/cta/opf-api	integration-libs/opf/cta/opf-api/config/default-opf-api-cta-config.ts:L13-L13
5	backend.opfApi.endpoints	initiatePayment	'payments'	defaultOpfApiPaymentConfig	@spartacus/opf/payment/opf-api	integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L18-L18

#	PERCORSO	CHIAVE	VALORE	CONFIG	LIBRERIA	RIFERIMENTO
6	backend.opfApi.endpoints	submitCompletePayment	'payments/${paymentSessionId}/submit-complete'	defaultOpfApiPaymentConfig	@spartacus/opf/payment/opf-api	integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L15-L15
7	backend.opfApi.endpoints	submitPayment	'payments/${paymentSessionId}/submit'	defaultOpfApiPaymentConfig	@spartacus/opf/payment/opf-api	integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L14-L14
8	backend.opfApi.endpoints	updatePaymentTransaction	'payments/${paymentSessionId}'	defaultOpfApiPaymentConfig	@spartacus/opf/payment/opf-api	integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L19-L19
9	backend.opfApi.endpoints	verifyPayment	'payments/${paymentSessionId}/verify'	defaultOpfApiPaymentConfig	@spartacus/opf/payment/opf-api	integration-libs/opf/payment/opf-api/config/default-opf-api-payment-config.ts:L13-L13

Tabella 3 — Augmentation di `OccEndpoints`

Interfaccia base: core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L299 (49 membri: product, productReviews, productReferences, productSearch, productSuggestions, component, components, pages, page, languages, currencies, countries, regions, addressCities, addressDistricts, paymentDetailsAll, paymentDetail, anonymousConsentTemplates, consentTemplates, consents, consentDetail, addresses, addressDetail, addressVerification, createVariantConfiguration, createTextfieldConfiguration, addTextfieldConfigurationToCart, readTextfieldConfigurationForCartEntry, updateTextfieldConfigurationForCartEntry, readVariantConfiguration, updateVariantConfiguration, addVariantConfigurationToCart, readVariantConfigurationForCartEntry, updateVariantConfigurationForCartEntry, readVariantConfigurationOverviewForOrderEntry, readVariantConfigurationPriceSummary, getVariantConfigurationOverview, searchConfiguratorVariants, customerCoupons, claimCoupon, claimCustomerCoupon, couponNotification, notificationPreference, productInterests, getProductInterests, baseSites, getActiveCostCenters, productSearchByCategory, productAvailabilities).

#	MODULO AUGMENTATO	EXTENDS	MEMBRI (DIRETTI + EREDITATI)	N.	LIBRERIA	RIFERIMENTO
1	@spartacus/core		asmCustomer360	1	@spartacus/asm/customer-360/occ	feature-libs/asm/customer-360/occ/model/occ-asm-customer-360-endpoints.model.ts:L10-L17
2	@spartacus/core		asmCustomerSearch, asmCustomerLists, asmBindCart, asmCreateCustomer, asmSessionEvent	5	@spartacus/asm/occ	feature-libs/asm/occ/model/occ-asm-endpoints.model.ts:L10-L21

#	MODULO AUGMENTATO	EXTENDS	MEMBRI (DIRETTI + EREDITATI)	N.	LIBRERIA	RIFERIMENTO
3	@spartacus/core	CartOccEndpoints (feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L9-L95)	carts, cart, createCart, deleteCart, addEntries, updateEntries, removeEntries, addEmail, saveCart, cartVoucher, cartApplyVoucher, cartRemoveVoucher, validate, cartAccessCode, cartGuestUser	15	@spartacus/cart/base/occ	feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L98-L98
4	@spartacus/core		savedCarts, saveCart, savedCart, restoreSavedCart, cloneSavedCart	5	@spartacus/cart/saved-cart/occ	feature-libs/cart/saved-cart/occ/model/occ-saved-cart-endpoints.model.ts:L10-L41
5	@spartacus/core	CheckoutB2BOccEndpoints (feature-libs/checkout/b2b/occ/model/occ-checkout-b2b-endpoints.model.ts:L9-L22)	paymentTypes, setCartCostCenter, setCartPaymentType	3	@spartacus/checkout/b2b/occ	feature-libs/checkout/b2b/occ/model/occ-checkout-b2b-endpoints.model.ts:L25-L25
6	@spartacus/core	CheckoutOccEndpoints (feature-libs/checkout/base/occ/model/occ-checkout-endpoints.model.ts:L9-L62)	setDeliveryAddress, cardTypes, setCartPaymentDetails, paymentProviderSubInfo, createPaymentDetails, getCheckoutDetails, createDeliveryAddress, removeDeliveryAddress, deliveryMode, setDeliveryMode, clearDeliveryMode, deliveryModes, setBillingAddress	13	@spartacus/checkout/base/occ	feature-libs/checkout/base/occ/model/occ-checkout-endpoints.model.ts:L65-L65
7	@spartacus/core		getTicket, getTickets, createTicketEvent, getTicketCategories, getTicketAssociatedObjects, createTicket, uploadAttachment, downloadAttachment	8	@spartacus/customer-ticketing/occ	feature-libs/customer-ticketing/occ/model/occ-customer-ticketing-endpoints.model.ts:L10-L60
8	@spartacus/core	DocumentFlowOccEndpoints (feature-libs/order/document-flow/occ/model/occ-document-flow-endpoints.model.ts:L9-L19)	subsequentDocuments, subsequentDocumentsEntries	2	@spartacus/order/document-flow/occ	feature-libs/order/document-flow/occ/model/occ-document-flow-endpoints.model.ts:L22-L22
9	@spartacus/core	OrderOccEndpoints (feature-libs/order/occ/model/occ-order-endpoints.model.ts:L9-L102)	orderHistory, orderDetail, quoteCode, consignmentTracking, cancelOrder, returnOrder, orderReturns, orderReturnDetail, cancelReturn, replenishmentOrderHistory, replenishmentOrderDetails, replenishmentOrderDetailsHistory, cancelReplenishmentOrder, placeOrder, placePaymentAuthorizedOrder, scheduleReplenishmentOrder, reorder, orderAttachments, downloadOrderAttachment	19	@spartacus/order/occ	feature-libs/order/occ/model/occ-order-endpoints.model.ts:L105-L105
10	@spartacus/core	AccountSummaryOccEndpoints (feature-libs/organization/account-summary/occ/model/occ-account-summary-endpoints.model.ts:L9-L33)	accountSummary, accountSummaryDocument, accountSummaryDocumentAttachment	3	@spartacus/organization/account-summary/occ	feature-libs/organization/account-summary/occ/model/occ-account-summary-endpoints.model.ts:L36-L36

#	MODULO AUGMENTATO EXTENDS	MEMBRI (DIRETTI + EREDITATI)	N.	LIBRERIA	RIFERIMENTO
11	@spartacus/core	b2bUsers, b2bUser, b2bUserApprovers, b2bUserApprover, b2bUserUserGroups, b2bUserUserGroup, b2bUserPermissions, b2bUserPermission, budget, budgets, costCentersAll, costCenter, costCenters, costCenterBudgets, costCenterBudget, orgUnits, orgUnitsAvailable, orgUnitsTree, orgUnitsApprovalProcesses, orgUnit, orgUnitUsers, orgUnitUserRoles, orgUnitUserRole, orgUnitApprovers, orgUnitApprover, orgUnitsAddresses, orgUnitsAddress, permissions, permission, orderApprovalPermissionTypes, userGroups, userGroup, userGroupAvailableOrderApprovalPermissions, userGroupAvailableOrgCustomers, userGroupMembers, userGroupMember, userGroupOrderApprovalPermissions, userGroupOrderApprovalPermission	38	@spartacus/organization/administration/occ	feature-libs/organization/administration/occ/model/occ-administration-endpoints.model.ts:L10-L239
12	@spartacus/core	orderApprovals, orderApproval, orderApprovalDecision	3	@spartacus/organization/order-approval	feature-libs/organization/order-approval/occ/model/occ-order-approval-endpoints.model.ts:L10-L29
13	@spartacus/core	unitLevelOrderHistory, unitLevelOrderDetail	2	@spartacus/organization/unit-order/occ	feature-libs/organization/unit-order/occ/model/occ-unit-order-endpoints.model.ts:L10-L24
14	@spartacus/core	organizationUserRegistration	1	@spartacus/organization/user-registration/occ	feature-libs/organization/user-registration/occ/model/occ-user-registration-endpoints.model.ts:L10-L17
15	@spartacus/core	pdfInvoicesListInvoices, pdfInvoicesDownloadInvoicePDF	2	@spartacus/pdf-invoices/occ	feature-libs/pdf-invoices/occ/model/occ-pdf-invoices.model.ts:L10-L26
16	@spartacus/core	storeDetails	1	@spartacus/pickup-in-store/occ	feature-libs/pickup-in-store/occ/model/occ-pickup-location-endpoints.model.ts:L10-L17
17	@spartacus/core	stock, stockAtStore	2	@spartacus/pickup-in-store/occ	feature-libs/pickup-in-store/occ/model/occ-stock-endpoints.model.ts:L10-L24

#	MODULO AUGMENTATO	EXTENDS	MEMBRI (DIRETTI + EREDITATI)	N.	LIBRERIA	RIFERIMENTO
18	@spartacus/core		getCpqAccessData, addCpqConfigurationToCart, updateCpqConfigurationForCartEntry, readCpqConfigurationForCartEntry, readCpqConfigurationForOrderEntry, createCpqConfiguration, readCpqConfiguration, readCpqConfigurationOverview, updateCpqAttribute, updateCpqAttributeValueQuantity, createCpqContainerRow, copyCpqContainerRow, removeCpqContainerRow, readCpqConfigurationForCartEntryFull, readCpqConfigurationForOrderEntryFull, readCpqConfigurationForQuoteEntryFull, readCpqConfigurationForSavedCartEntryFull	17	@spartacus/product-configurator/rulebased/cpq	feature-libs/product-configurator/rulebased/cpq/occ/model/occ-cpq-configurator-endpoints.model.ts:L9-L82
19	@spartacus/core		readVariantConfigurationOverviewForQuoteEntry, readVariantConfigurationOverviewForSavedCartEntry	2	@spartacus/product-configurator/rulebased	feature-libs/product-configurator/rulebased/occ/model/occ-variant-configurator-endpoints.model.ts:L9-L18
20	@spartacus/core		readTextfieldConfigurationForOrderEntry, readTextfieldConfigurationForQuoteEntry	2	@spartacus/product-configurator/textfield	feature-libs/product-configurator/textfield/occ/occ-configurator-textfield-endpoints.model.ts:L9-L12
21	@spartacus/core	FutureStockOccEndpoints (feature-libs/product/future-stock/occ/model/occ-future-stock-endpoints.model.ts:L9-L18)	futureStock, futureStocks	2	@spartacus/product/future-stock/occ	feature-libs/product/future-stock/occ/model/occ-future-stock-endpoints.model.ts:L21-L21
22	@spartacus/core	QuoteOccEndpoints (feature-libs/quote/occ/model/occ-quote-endpoints.model.ts:L9-L59)	getQuotes, createQuote, getOrderCode, getQuote, editQuote, performQuoteAction, addComment, addDiscount, addQuoteEntryComment, downloadAttachment	10	@spartacus/quote/occ	feature-libs/quote/occ/model/occ-quote-endpoints.model.ts:L62-L62
23	@spartacus/core		requestedDeliveryDate	1	@spartacus/requested-delivery-date/occ	feature-libs/requested-delivery-date/occ/model/occ-requested-delivery-date-endpoints.model.ts:L10-L17
24	@spartacus/core		store, stores, storescounts	3	@spartacus/storefinder/occ	feature-libs/storefinder/occ/model/occ-storefinder-endpoints.model.ts:L10-L29

#	MODULO AUGMENTATO	EXTENDS	MEMBRI (DIRETTI + EREDITATI)	N.	LIBRERIA	RIFERIMENTO
25	@spartacus/core	SubscriptionBillingOccEndpoints (feature-libs/subscription-billing/occ/model/occ-subscription-billing-endpoints.model.ts:L9-L64)	subscriptionList, subscriptionByCode, extensionEffectiveDate, extendSubscription, cancelSubscription, getEffectiveCancellationDate, withdrawSubscription, reverseCancellation, subscriptionBillsList, subscriptionBillByCode	10	@spartacus/subscription-billing/occ	feature-libs/subscription-billing/occ/model/occ-subscription-billing-endpoints.model.ts:L67-L67
26	@spartacus/core	UserAccountOccEndpoints (feature-libs/user/account/occ/adapters/config/occ-user-account-endpoint.model.ts:L9-L19)	user, createVerificationToken	2	@spartacus/user/account/occ	feature-libs/user/account/occ/adapters/config/occ-user-account-endpoint.model.ts:L21-L21
27	@spartacus/core	UserProfileOccEndpoints (feature-libs/user/profile/occ/adapters/config/occ-user-profile-endpoint.model.ts:L9-L54)	titles, userRegister, userForgotPassword, userRestoreToken, userResetPassword, userUpdateLoginId, userUpdatePassword, userUpdateProfile, userCloseAccount	9	@spartacus/user/profile/occ	feature-libs/user/profile/occ/adapters/config/occ-user-profile-endpoint.model.ts:L56-L56
28	@spartacus/core	DigitalPaymentsOccEndpoints (integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.model.ts:L9-L12)	paymentRequest, paymentDetails	2	@spartacus/digital-payments	integration-libs/digital-payments/src/checkout/adapters/config/occ-digital-payments-endpoint.model.ts:L14-L14
29	@spartacus/core		cartUserEmail	1	@spartacus/opf/checkout/root	integration-libs/opf/checkout/root/model/occ-opf-api-checkout-endpoints.model.ts:L10-L12
30	@spartacus/core	OccOpfGiftCardEndpoints (integration-libs/opf/gift-card/occ/model/occ-opf-gift-card-endpoints.model.ts:L9-L18)	applyGiftCard, removeGiftCard	2	@spartacus/opf/gift-card/occ	integration-libs/opf/gift-card/occ/model/occ-opf-gift-card-endpoints.model.ts:L21-L21
31	@spartacus/core	OccOpfPaymentEndpoints (integration-libs/opf/payment/occ/model/occ-opf-payment-endpoints.model.ts:L9-L14)	setCartPaymentOption	1	@spartacus/opf/payment/occ	integration-libs/opf/payment/occ/model/occ-opf-payment-endpoints.model.ts:L17-L17
32	@spartacus/core	OccOpfQuickBuyCartEndpoints (integration-libs/opf/quick-buy/occ/model/occ-opf-quick-buy-cart-endpoints.model.ts:L9-L30)	quickBuyCreateDeliveryAddress, quickBuySetBillingAddress, quickBuyDeliveryModes, quickBuySetDeliveryMode, quickBuySelectedDeliveryMode	5	@spartacus/opf/quick-buy/occ	integration-libs/opf/quick-buy/occ/model/occ-opf-quick-buy-cart-endpoints.model.ts:L33-L33

#	MODULO AUGMENTATO	EXTENDS	MEMBRI (DIRETTI + EREDITATI)	N.	LIBRERIA	RIFERIMENTO
33	@spartacus/core	PunchoutOccEndpoints (integration-libs/punchout/occ/model/occ-punchout.model.ts:L9-L22)	punchoutSession, punchoutSessionRequisition, punchoutSession, punchoutSessionRequisition	4	@spartacus/punchout/occ	integration-libs/punchout/occ/model/occ-punchout.model.ts:L25-L28
34	@spartacus/core	S4ServiceOccEndpoints (integration-libs/s4-service/order/occ/model/occ-s4-service-endpoints.model.ts:L9-L25)	cancelServiceOrder, rescheduleService, cancelServiceOrder, rescheduleService	4	@spartacus/s4-service/order	integration-libs/s4-service/order/occ/model/occ-s4-service-endpoints.model.ts:L28-L31
35	@spartacus/core	CheckoutServiceOrderOccEndpoints (integration-libs/s4-service/root/model/augmented-types.model.ts:L23-L28)	setServiceScheduleSlot	1	@spartacus/s4-service/root	integration-libs/s4-service/root/model/augmented-types.model.ts:L50-L50

Incroci

- Chiavi con default ma non dichiarate in `OccEndpoints`: `addWishlistEntry`, `getWishlistEntries`, `getWishlists`, `removeWishlistEntry`
- Membri di `OccEndpoints` senza default `backend.occ.endpoints` in ambito: —

A05

Guard

Guard di route

IN QUESTO CAPITOLO

Conteggi

Come rigenerare

Tabella 1 — Classi guard

Tabella 2 — Funzioni guard (CanActivateFn & co.)

Generata da `tools/extract2/a05_guards.py` allo SHA 5b7a1c2b. Non modificare a mano: rigenerare.

Tabella 1: classi che (a) dichiarano `implements CanActivate|CanActivateChild|CanDeactivate|CanMatch|CanLoad`, oppure (b) definiscono a primo livello un metodo `canActivate()`/`canActivateChild()`/`canDeactivate()`/`canMatch()`/`canLoad()` senza `implements` (guard "strutturali": la maggior parte delle guard Spartacus, perché le interfacce class-based sono deprecate in Angular), oppure (c) estendono ricorsivamente una delle precedenti. "Usi" conta le occorrenze del nome come elemento di array `guards: [...]` (config `cmsComponents`) o `canActivate|canActivateChild|canDeactivate|canMatch: [...]` (route) nell'ambito.

Tabella 2: funzioni guard tipizzate (`const x: CanActivateFn = ...` o funzioni che ritornano `Can*Fn`).

Conteggi

- Classi guard (righe tabella 1): **43**
- di cui con `implements Can*` esplicito: **4**
- di cui solo strutturali (metodo `canActivate()` ecc. senza `implements`): **36**
- di cui solo per ereditarietà (`extends` di una guard): **3**
- Classi con `CanActivate` (implements o metodo): **40**
- Funzioni guard tipizzate `Can*Fn` (tabella 2): **0**
- Entry point coinvolti: **21**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract2/a05_guards.py
# oppure tutte le appendici: bash tools/extract2/run_all.sh
```

Ambito: `git ls-files core-libs feature-libs integration-libs | grep '\.ts$' | grep -vE '\.spec\.ts$|/testing/|\.mock\.ts$|/schematics/'`. La colonna "Libreria" è l'entry point npm ricavato dal `package.json`/`ng-package.json` più vicino.

Tabella 1 — Classi guard

#	CLASSE	IMPLEMENTS	METODI GUARD	CLASSE PADRE	USI (GUARDS/ROUTE)	LIBRERIA	RIFERIMENTO
1	AdminGuard	—	canActivate()		guards: 9	@spartacus/organization/administration/core	feature-libs/organization/administration/core/guards/admin.guard.ts:L20-L50
2	ApproverGuard	—	canActivate()		guards: 6	@spartacus/organization/order-approval	feature-libs/organization/order-approval/core/guards/approver.guard.ts:L22-L54

#	CLASSE	IMPLEMENTS	METODI GUARD	CLASSE PADRE	USI (GUARDS/ROUTE)	LIBRERIA	RIFERIMENTO
3	AuthGuard	—	canActivate()		canActivate: 18, guards: 77	@spartacus/core	core-libs/core/src/auth/user-auth/guards/auth.guard.ts:L22-L43
4	BeforeCmsPageGuardService	—	canActivate()			@spartacus/storefront	core-libs/storefront/cms-structure/guards/before-cms-page-guard.service.ts:L18-L33
5	CancelServiceOrderGuard	—	canActivate()		guards: 1	@spartacus/s4-service/order	integration-libs/s4-service/order/components/guards/cancel-service-order.guard.ts:L16-L38
6	CartNotEmptyGuard	—	canActivate()		guards: 24	@spartacus/checkout/base/components	feature-libs/checkout/base/components/guards/cart-not-empty.guard.ts:L17-L40
7	CartValidationGuard	—	canActivate()		guards: 5	@spartacus/cart/base/core	feature-libs/cart/base/core/guards/cart-validation.guard.ts:L29-L103
8	CdcLoginAsGuestGuard	eredita da LoginAsGuestGuard		LoginAsGuestGuard		@spartacus/cdc/root	integration-libs/cdc/root/guards/cdc-login-as-guest.guard.ts:L13-L17
9	CdcLoginGuard	eredita da LoginGuard		LoginGuard		@spartacus/cdc/root	integration-libs/cdc/root/guards/cdc-login.guard.ts:L20-L32
10	CdcLogoutGuard	eredita da LogoutGuard		LogoutGuard		@spartacus/cdc/root	integration-libs/cdc/root/guards/cdc-logout.guard.ts:L26-L56
11	CheckoutAuthGuard	—	canActivate()		guards: 24	@spartacus/checkout/base/components	feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L25-L78
12	CheckoutB2BAuthGuard	—	canActivate()	CheckoutAuthGuard		@spartacus/checkout/b2b/components	feature-libs/checkout/b2b/components/guards/checkout-b2b-auth.guard.ts:L30-L89
13	CheckoutB2BStepsSetGuard	—	canActivate()	CheckoutStepsSetGuard		@spartacus/checkout/b2b/components	feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L32-L178
14	CheckoutGuard	—	canActivate()		guards: 1	@spartacus/checkout/base/components	feature-libs/checkout/base/components/guards/checkout.guard.ts:L21-L71
15	CheckoutServiceOrderStepsSetGuard	—	canActivate()	CheckoutB2BStepsSetGuard		@spartacus/s4-service/checkout	integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts:L21-L100
16	CheckoutStepsSetGuard	—	canActivate()		guards: 3	@spartacus/checkout/base/components	feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L36-L187
17	CmsPageGuard	—	canActivate()		canActivate: 41	@spartacus/storefront	core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L26-L92
18	CustomLoginGuard	CanActivate	canActivate()		guards: 2	@spartacus/core	core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L50-L138
19	ExternalRoutesGuard	—	canActivate()		canActivate: 1	@spartacus/core	core-libs/core/src/routing/external-routes/external-routes.guard.ts:L13-L42
20	FederatedLoginGuard	CanActivate	canActivate()			@spartacus/core	core-libs/core/src/auth/user-auth/guards/federated-login.guard.ts:L32-L105

#	CLASSE	IMPLEMENTS	METODI GUARD	CLASSE PADRE	USI (GUARDS/ROUTE)	LIBRERIA	RIFERIMENTO
21	GigyaRaasGuard	CanActivate	canActivate()		guards: 1	@spartacus/cdc/components	integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts:L32-L123
22	GuardsComposer	—	canActivate()			@spartacus/storefront	core-libs/storefront/cms-structure/services/guards-composer.ts:L31-L94
23	LoginAsGuestGuard	—	canActivate()		guards: 1	@spartacus/user/account/components	feature-libs/user/account/components/guards/login-as-guest.guard.ts:L20-L46
24	LoginGuard	—	canActivate()		canActivate: 1	@spartacus/storefront	core-libs/storefront/cms-components/user/login-route/login.guard.ts:L27-L66
25	LogoutGuard	—	canActivate()		canActivate: 1	@spartacus/storefront	core-libs/storefront/cms-components/user/logout/logout.guard.ts:L31-L111
26	NotAuthGuard	—	canActivate()		guards: 14	@spartacus/core	core-libs/core/src/auth/user-auth/guards/not-auth.guard.ts:L21-L41
27	NotCheckoutAuthGuard	—	canActivate()		guards: 1	@spartacus/checkout/base/components	feature-libs/checkout/base/components/guards/not-checkout-auth.guard.ts:L21-L45
28	OAuthCallbackGuard	—	canActivate()		guards: 1	@spartacus/core	core-libs/core/src/auth/user-auth/guards/oauth-callback.guard.ts:L25-L53
29	OpfB2bCheckoutStepsSetGuard	—	canActivate()	CheckoutStepsSetGuard		@spartacus/opf/b2b-checkout/root	integration-libs/opf/b2b-checkout/root/guards/opf-b2b-checkout-steps-set.guard.ts:L42-L193
30	OpfCheckoutAuthGuard	—	canActivate()	CheckoutAuthGuard		@spartacus/opf/checkout/root	integration-libs/opf/checkout/root/checkout-guard/opf-checkout-auth.guard.ts:L21-L63
31	OppsLoginRequiredGuard	—	canActivate()			@spartacus/opps/root	integration-libs/opps/root/login-required/opps-login-required.guard.ts:L22-L62
32	OrderCancellationGuard	—	canActivate()		guards: 1	@spartacus/order/components	feature-libs/order/components/amend-order/cancellations/order-cancellation.guard.ts:L17-L40
33	OrderConfirmationGuard	—	canActivate()		guards: 11	@spartacus/order/components	feature-libs/order/components/guards/order-confirmation.guard.ts:L17-L37
34	OrderReturnGuard	—	canActivate()		guards: 1	@spartacus/order/components	feature-libs/order/components/amend-order/returns/order-return.guard.ts:L17-L40
35	OrgUnitGuard	—	canActivate()		canActivate: 3	@spartacus/organization/administration/core	feature-libs/organization/administration/core/guards/org-unit.guard.ts:L17-L38
36	ProductMultiDimensionalSelectorGuard	—	canActivate()		guards: 1	@spartacus/product-multi-dimensional/selector/components	feature-libs/product-multi-dimensional/selector/components/guards/product-multi-dimensional-selector.guard.ts:L33-L111
37	ProductVariantsGuard	—	canActivate()		guards: 1	@spartacus/product/variants/components	feature-libs/product/variants/components/guards/product-variants.guard.ts:L26-L81

#	CLASSE	IMPLEMENTS	METODI GUARD	CLASSE PADRE	USI (GUARDS/ROUTE)	LIBRERIA	RIFERIMENTO
38	ProtectedRoutesGuard	—	canActivate()			@spartacus/core	core-libs/core/src/routing/protected-routes/protected-routes.guard.ts:L14-L34
39	PunchoutNavigationGuard	—	canActivate()			@spartacus/punchout/root	integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L26-L141
40	QuoteCartGuard	CanActivate	canActivate()		guards: 1	@spartacus/quote/components/cart-guard	feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L28-L81
41	ServiceOrderGuard	—	canActivate()		guards: 1	@spartacus/s4-service/order	integration-libs/s4-service/order/components/guards/service-order.guard.ts:L16-L35
42	UnitLevelOrdersViewerGuard	—	canActivate()		guards: 4	@spartacus/organization/unit-order/core	feature-libs/organization/unit-order/core/guards/unit-level-orders-viewer.guard.ts:L23-L55
43	UserGuard	—	canActivate()		canActivate: 5	@spartacus/organization/administration/core	feature-libs/organization/administration/core/guards/user.guard.ts:L17-L38

Tabella 2 — Funzioni guard (CanActivateFn & co.)

Nessuna const/function tipizzata CanActivateFn/CanActivateChildFn/CanDeactivateFn/CanMatchFn trovata in ambito. CanActivateFn compare solo come tipo nell'infrastruttura (es. BEFORE_CMS_PAGE_GUARD, CmsRoutesImplService, GuardsComposer).

Occorrenze del tipo (non commenti): core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L8-L8 (CanActivateFn), core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L23-L23 (CanActivateFn), core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L10-L10 (CanActivateFn), core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L165-L165 (CanActivateFn), core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L179-L179 (CanActivateFn), core-libs/storefront/cms-structure/services/guards-composer.ts:L9-L9 (CanActivateFn), core-libs/storefront/cms-structure/services/guards-composer.ts:L19-L19 (CanActivateFn)

A06

HTTP interceptor

Classi interceptor e registrazioni

IN QUESTO CAPITOLO

Conteggi

Come rigenerare

Tabella 1 — Classi interceptor

Tabella 2 — Registrazioni HTTP_INTERCEPTORS

Tabella 3 — Interceptor funzionali

Generata da `tools/extract2/a06_interceptors.py` allo SHA 5b7a1c2b. Non modificare a mano: rigenerare.

Tabella 1: classi che dichiarano `implements HttpInterceptor`, o che definiscono a primo livello un metodo `intercept()`, o che estendono (ricorsivamente) una di queste. La colonna "Registrazioni" rimanda alla tabella 2 (numero di provider `HTTP_INTERCEPTORS` con quella classe come `useExisting`/`useClass`).

Tabella 2: ogni object literal provider con `provide: HTTP_INTERCEPTORS`. "Contenitore" è la dichiarazione top-level che lo contiene (es. `const interceptors: Provider[]` poi importata da un modulo, o la classe `@NgModule`). L'ordine di registrazione determina l'ordine della catena: vedi i capitoli sul modulo HTTP.

Conteggi

- Classi interceptor (righe tabella 1): **22**
- di cui con `implements HttpInterceptor`: **22**
- di cui solo con metodo `intercept()` senza `implements`: **0**
- di cui solo per ereditarietà: **0**
- Registrazioni `{ provide: HTTP_INTERCEPTORS, ... }` (righe tabella 2): **22**
- Classi interceptor senza registrazione `HTTP_INTERCEPTORS` in ambito: **0**
- Interceptor funzionali (`HttpInterceptorFn` / `withInterceptors`): **0**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract2/a06_interceptors.py
# oppure tutte le appendici: bash tools/extract2/run_all.sh
```

Ambito: `git ls-files core-libs feature-libs integration-libs | grep '\.ts$' | grep -vE '\.spec\.ts$|/testing/|\.mock\.ts$|/schematics/'`. La colonna "Libreria" è l'entry point npm ricavato dal `package.json`/`ng-package.json` più vicino.

Tabella 1 — Classi interceptor

#	CLASSE	IMPLEMENTS HTTPINTERCEPTOR	INTERCEPT() PROPRIO	CLASSE PADRE	REGISTRAZIONI	LIBRERIA	RIFERIMENTO
1	AnonymousConsentsInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L28-L136
2	AuthInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L25-L111

#	CLASSE	IMPLEMENTS HTTPINTERCEPTOR	INTERCEPT() PROPRIO	CLASSE PADRE	REGISTRAZIONI	LIBRERIA	RIFERIMENTO
3	BlobErrorInterceptor	sì	sì		1	@spartacus/organization/account-summary/root	feature-libs/organization/account-summary/root/http-interceptors/blob-error.interceptor.ts:L23-L57
4	CheckoutCartInterceptor	sì	sì		1	@spartacus/checkout/base/root	feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L27-L95
5	ClientTokenInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L32-L112
6	CmsTicketInterceptor	sì	sì		1	@spartacus/smartedit/root	feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L20-L73
7	ConsentReferenceInterceptor	sì	sì		1	@spartacus/cds	integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts:L19-L46
8	DebugInterceptor	sì	sì		1	@spartacus/cds	integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts:L19-L42
9	HttpErrorHandlerInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L41-L99
10	HttpErrorInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L23-L66
11	HttpTimeoutInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L27-L125
12	OccOppsCouponCodesInterceptor	sì	sì		1	@spartacus/opps/root	integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L20-L65
13	OccPersonalizationIdInterceptor	sì	sì		1	@spartacus/tracking/personalization/root	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L21-L99
14	OccPersonalizationTimeInterceptor	sì	sì		1	@spartacus/tracking/personalization/root	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L21-L99
15	OccSegmentRefsInterceptor	sì	sì		1	@spartacus/segment-refs/root	integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L19-L87

#	CLASSE	IMPLEMENTS HTTPINTERCEPTOR	INTERCEPT() PROPRIO	CLASSE PADRE	REGISTRAZIONI	LIBRERIA	RIFERIMENTO
16	OpfGiftCardPaymentApiInterceptor	sì	sì		1	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/http-interceptors/opf-gift-card-payment-api.interceptor.ts:L33-L66
17	PunchoutCartInterceptor	sì	sì		1	@spartacus/punchout/root	integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts:L23-L50
18	SiteContextInterceptor	sì	sì		2	@spartacus/core	core-libs/core/src/occ/adapters/site-context/site-context.interceptor.ts:L26-L69
19	SiteContextInterceptor	sì	sì		2	@spartacus/asm/customer-360/root	feature-libs/asm/customer-360/root/interceptors/site-context.interceptor.ts:L26-L73
20	TokenRevocationInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L23-L59
21	UserIdHttpHeaderInterceptor	sì	sì		1	@spartacus/asm/root	feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L30-L85
22	WithCredentialsInterceptor	sì	sì		1	@spartacus/core	core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L21-L53

Tabella 2 — Registrazioni HTTP_INTERCEPTORS

#	CLASSE REGISTRATA	IMPORTATA DA	MODALITÀ	MULTI	CONTENITORE	LIBRERIA	RIFERIMENTO
1	AnonymousConsentsInterceptor	./anonymous-consents-interceptor	useExisting	true	interceptors	@spartacus/core	core-libs/core/src/anonymous-consents/http-interceptors/index.ts:L12-L16
2	ClientTokenInterceptor	./client-token.interceptor	useExisting	true	interceptors	@spartacus/core	core-libs/core/src/auth/client-auth/http-interceptors/index.ts:L12-L16
3	AuthInterceptor	./auth.interceptor	useExisting	true	interceptors	@spartacus/core	core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L13-L17
4	TokenRevocationInterceptor	./token-revocation.interceptor	useExisting	true	interceptors	@spartacus/core	core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L18-L22

#	CLASSE REGISTRATA	IMPORTATA DA	MODALITÀ	MULTI	CONTENITORE	LIBRERIA	RIFERIMENTO
5	HttpErrorHandlerInterceptor	./http-error-handler.interceptor	useExisting	true	HttpErrorHandlerModule	@spartacus/core	core-libs/core/src/error-handling/http-error-handler/http-error-handler.module.ts:L17-L21
6	HttpErrorInterceptor	./http-error.interceptor	useExisting	true	httpErrorInterceptors	@spartacus/core	core-libs/core/src/global-message/http-interceptors/index.ts:L66-L70
7	HttpTimeoutInterceptor	./http-timeout.interceptor	useExisting	true	HttpTimeoutModule	@spartacus/core	core-libs/core/src/http/http-timeout/http-timeout.module.ts:L15-L19
8	SiteContextInterceptor	./site-context.interceptor	useExisting	true	SiteContextOccModule	@spartacus/core	core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts:L26-L30
9	WithCredentialsInterceptor	./interceptors/with-credentials.interceptor	useExisting	true	BaseOccModule	@spartacus/core	core-libs/core/src/occ/base-occ.module.ts:L25-L29
10	SiteContextInterceptor	./interceptors/site-context.interceptor	useExisting	true	AsmCustomer360RootModule	@spartacus/asm/customer-360/root	feature-libs/asm/customer-360/root/asm-customer-360-root.module.ts:L29-L33
11	UserIdHttpHeaderInterceptor	./interceptors/user-id-http-header.interceptor	useExisting	true	AsmRootModule	@spartacus/asm/root	feature-libs/asm/root/asm-root.module.ts:L88-L92
12	CheckoutCartInterceptor	./checkout-cart.interceptor	useExisting	true	interceptors	@spartacus/checkout/base/root	feature-libs/checkout/base/root/http-interceptors/index.ts:L12-L16
13	BlobErrorInterceptor	./http-interceptors/blob-error.interceptor	useExisting	true	AccountSummaryRootModule	@spartacus/organization/account-summary/root	feature-libs/organization/account-summary/root/account-summary-root.module.ts:L52-L56
14	CmsTicketInterceptor	./cms-ticket.interceptor	useExisting	true	interceptors	@spartacus/smartedit/root	feature-libs/smartedit/root/http-interceptors/index.ts:L12-L16
15	OccPersonalizationIdInterceptor	./occ-personalization-id.interceptor	useExisting	true	interceptors	@spartacus/tracking/personalization/root	feature-libs/tracking/personalization/root/http-interceptors/index.ts:L14-L18

#	CLASSE REGISTRATA	IMPORTATA DA	MODALITÀ	MULTI	CONTENITORE	LIBRERIA	RIFERIMENTO
16	OccPersonalizationTimeInterceptor	./occ-personalization-time.interceptor	useExisting	true	interceptors	@spartacus/tracking/personalization/root	feature-libs/tracking/personalization/root/http-interceptors/index.ts:L19-L23
17	ConsentReferenceInterceptor	./http-interceptors/consent-reference-interceptor	useExisting	true	ProfileTagModule	@spartacus/cds	integration-libs/cds/src/profiletag/profile-tag.module.ts:L19-L23
18	DebugInterceptor	./http-interceptors/debug-interceptor	useExisting	true	ProfileTagModule	@spartacus/cds	integration-libs/cds/src/profiletag/profile-tag.module.ts:L24-L24
19	OpfGiftCardPaymentApiInterceptor	./http-interceptors	useClass	true	OpfGiftCardRootModule	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L57-L61
20	OccOppsCouponCodesInterceptor	./occ-opps-coupon-codes.interceptor	useExisting	true	oppsCouponCodesInterceptors	@spartacus/opps/root	integration-libs/opps/root/coupon-codes/http-interceptors/index.ts:L12-L16
21	PunchoutCartInterceptor	./punchout-cart.interceptor	useExisting	true	interceptors	@spartacus/punchout/root	integration-libs/punchout/root/interceptors/index.ts:L12-L16
22	OccSegmentRefsInterceptor	./occ-segment-refs.interceptor	useExisting	true	segmentRefsInterceptors	@spartacus/segment-refs/root	integration-libs/segment-refs/root/http-interceptors/index.ts:L12-L16

Classi senza registrazione diretta in ambito

Classi base astratte/estese, interceptor registrati altrove (es. app) o non registrati: —

Tabella 3 — Interceptor funzionali

Nessuna occorrenza di `HttpInterceptorFn` o `withInterceptors(` (esclusi commenti) in ambito: tutti gli interceptor Spartacus sono class-based e registrati via `HTTP_INTERCEPTORS`.

A07

Eventi (CxEvent)

Gerarchia degli eventi che derivano da `CxEvent`

IN QUESTO CAPITOLO

Conteggi

Come rigenerare

Tabella — Classi evento

Generata da `tools/extract2/a07_events.py` allo SHA 5b7a1c2b. Non modificare a mano: rigenerare.

Le classi evento sono individuate risalendo ricorsivamente le clausole `extends` a partire da `CxEvent` (`core-libs/core/src/event/cx-event.ts`). "Profondità" = distanza da `CxEvent`. "Registrazioni" individua (a) chiamate `f({ action, event: X, ... })` (oggetto con `event` e `action`, o callee con "register" nel nome) — `stateEventService.register` o helper come `registerMapped`/`buildRestoreSavedCartEvents` — con `action` mostrata, e (b) `<...>eventService.register(X, source$)`. "Dispatch" conta `.dispatch(new X(...))`, `.dispatch(createFrom(X, ...))` e `.dispatch(obj, X)`. Eventi senza registrazione/dispatch possono essere classi padre astratte, emessi tramite builder generici o registrati con forme non rilevate (NOTA in tal caso verificare sul codice).

Conteggi

- Classi evento (discendenti di `CxEvent`, incluso `CxEvent`): **115**
- Profondità massima della gerarchia: **3**
- Classi `abstract`: **15**
- Classi con `static type` letterale: **101**
- Classi usate come genitore di altri eventi: **17**
- Classi con almeno una registrazione rilevata: **47**
- Siti di registrazione rilevati (totale): **47**
- Classi con almeno un `dispatch` rilevato: **46**
- Classi istanziate con `new X(` in ambito: **3**
- Classi senza registrazione, dispatch né `new` rilevati: **19**
- Entry point coinvolti: **13**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract2/a07_events.py
# oppure tutte le appendici: bash tools/extract2/run_all.sh
```

Ambito: `git ls-files core-libs feature-libs integration-libs | grep '\.ts$' | grep -vE '\.spec\.ts$|/testing/|\.mock\.ts$|/schematics/'`. La colonna "Libreria" è l'entry point npm ricavato dal `package.json`/`ng-package.json` più vicino.

Tabella — Classi evento

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
1	Logout Event	Logout Event	CxEvent	1			eventService.register core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L51-L51			@spartacus/core	core-libs/core/src/auth/user-auth/events/user-auth.events.ts:L15-L20
2	Login Event	Login Event	CxEvent	1			register ← AuthActions.LOGIN core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L41-L44			@spartacus/core	core-libs/core/src/auth/user-auth/events/user-auth.events.ts:L28-L33
3	CxEvent	CxEvent		0	sì	41				@spartacus/core	core-libs/core/src/event/cx-event.ts:L12-L17
4	Module InitializedEvent	Module Initialized Event	CxEvent	1				core-libs/core/src/lazy-loading/lazy-modules.service.ts:L100-L100 core-libs/core/src/lazy-loading/lazy-modules.service.ts:L132-L132		@spartacus/core	core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L32
5	Facet Changed Event	Facet Changed Event	CxEvent	1			eventService.register core-libs/core/src/product/event/product-event.builder.ts:L31-L32			@spartacus/core	core-libs/core/src/product/event/product.events.ts:L12-L37
6	LanguageSetEvent	LanguageSet Event	CxEvent	1			eventService.register core-libs/core/src/site-context/events/site-context-event.builder.ts:L53-L53			@spartacus/core	core-libs/core/src/site-context/events/site-context.events.ts:L12-L18
7	CurrencySetEvent	CurrencySet Event	CxEvent	1			eventService.register core-libs/core/src/site-context/events/site-context-event.builder.ts:L71-L71			@spartacus/core	core-libs/core/src/site-context/events/site-context.events.ts:L23-L29
8	User Address Event		CxEvent	1	sì	5				@spartacus/core	core-libs/core/src/user/events/user.events.ts:L10-L12
9	Update User Address Event	Update User Address Event	User Address Event	2			register ← UserActions.UPDATE_USER_ADDRESS core-libs/core/src/user/events/user-event.builder.ts:L35-L38			@spartacus/core	core-libs/core/src/user/events/user.events.ts:L14-L18
10	Delete User Address Event	Delete User Address Event	User Address Event	2			register ← UserActions.DELETE_USER_ADDRESS core-libs/core/src/user/events/user-event.builder.ts:L49-L52			@spartacus/core	core-libs/core/src/user/events/user.events.ts:L20-L23

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
11	AddUserAddressEvent	AddUserAddressEvent	UserAddressEvent	2			register ← UserActions.ADD_USER_ADDRESS core-libs/core/src/user/events/user-event.builder.ts:L42-L45			@spartacus/core	core-libs/core/src/user/events/user.events.ts:L25-L28
12	LoadUserAddressesEvent	LoadUserAddressesEvent	UserAddressEvent	2				feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L91-L91		@spartacus/core	core-libs/core/src/user/events/user.events.ts:L30-L32
13	LoadUserPaymentMethodsEvent	LoadUserPaymentMethodsEvent	UserAddressEvent	2				feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L50-L50		@spartacus/core	core-libs/core/src/user/events/user.events.ts:L34-L36
14	SearchBoxSuggestionSelectedEvent	SearchBoxSuggestionSelectedEvent	CxEvent	1				core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L222-L222		@spartacus/storefront	core-libs/storefront/cms-components/navigation/search-box/search-box.events.ts:L12-L20
15	SearchBoxProductSelectedEvent	SearchBoxProductSelectedEvent	CxEvent	1				core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L205-L205		@spartacus/storefront	core-libs/storefront/cms-components/navigation/search-box/search-box.events.ts:L25-L32
16	ComponentEvent		CxEvent	1	sì	2				@spartacus/storefront	core-libs/storefront/cms-structure/page/component/events/component.event.ts:L15-L24
17	ComponentCreateEvent	ComponentCreate	ComponentEvent	2				core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L110-L110		@spartacus/storefront	core-libs/storefront/cms-structure/page/component/events/component.event.ts:L29-L38
18	ComponentDestroyEvent	ComponentDestroy	ComponentEvent	2				core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L114-L114		@spartacus/storefront	core-libs/storefront/cms-structure/page/component/events/component.event.ts:L43-L48

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
19	HomePageEvent	HomePageEvent	PageEvent	2			eventService.register core-libs/storefront/events/home/home-page-event.builder.ts:L23-L23			@spartacus/storefront	core-libs/storefront/events/home/home-page.events.ts:L12-L15
20	NavigationEvent	NavigationEvent	CxEvent	1			eventService.register core-libs/storefront/events/navigation/navigation-event.builder.ts:L38-L38			@spartacus/storefront	core-libs/storefront/events/navigation/navigation.event.ts:L13-L34
21	PageEvent		CxEvent	1	sì	5				@spartacus/storefront	core-libs/storefront/events/page/page.events.ts:L13-L18
22	ProductDetailsPageEvent	ProductDetailsPageEvent	PageEvent	2			eventService.register core-libs/storefront/events/product/product-page-event.builder.ts:L40-L41			@spartacus/storefront	core-libs/storefront/events/product/product-page.events.ts:L13-L20
23	CategoryPageResultsEvent	CategoryPageResultsEvent	PageEvent	2			eventService.register core-libs/storefront/events/product/product-page-event.builder.ts:L44-L45			@spartacus/storefront	core-libs/storefront/events/product/product-page.events.ts:L25-L31
24	SearchPageResultsEvent	SearchPageResultsEvent	PageEvent	2			eventService.register core-libs/storefront/events/product/product-page-event.builder.ts:L36-L37			@spartacus/storefront	core-libs/storefront/events/product/product-page.events.ts:L37-L42
25	CartPageEvent	CartPageEvent	PageEvent	2			eventService.register feature-libs/cart/base/core/event/cart-page-event.builder.ts:L23-L23			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart-page.events.ts:L12-L15
26	CartEvent		CxEvent	1	sì	19				@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L14-L33
27	CreateCartEvent	CreateCartEvent	CartEvent	2			register ← CartActions.CREATE_CART feature-libs/cart/base/core/event/cart-event.builder.ts:L119-L122			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L35-L40
28	CreateCartSuccessEvent	CreateCartSuccessEvent	CartEvent	2			register ← CartActions.CREATE_CART_SUCCESS feature-libs/cart/base/core/event/cart-event.builder.ts:L123-L126			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L42-L47
29	CreateCartFailEvent	CreateCartFailEvent	CartEvent	2			register ← CartActions.CREATE_CART_FAIL feature-libs/cart/base/core/event/cart-event.builder.ts:L127-L130			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L49-L55

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
30	CartAddEntryEvent	CartAddEntryEvent	CartEvent	2			registerMapped ← CartActions.CART_ADD_ENTRY feature-libs/cart/base/core/event/cart-event.builder.ts:L75-L78			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L57-L64
31	CartAddEntrySuccessEvent	CartAddEntrySuccessEvent	CartEvent	2			registerMapped ← CartActions.CART_ADD_ENTRY_SUCCESS feature-libs/cart/base/core/event/cart-event.builder.ts:L79-L82			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L66-L76
32	CartAddEntryFailEvent	CartAddEntryFailEvent	CartEvent	2			registerMapped ← CartActions.CART_ADD_ENTRY_FAIL feature-libs/cart/base/core/event/cart-event.builder.ts:L83-L86			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L78-L86
33	CartRemoveEntryFailEvent	CartRemoveEntryFailEvent	CartEvent	2			registerMapped ← CartActions.CART_REMOVE_ENTRY_FAIL feature-libs/cart/base/core/event/cart-event.builder.ts:L94-L97			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L88-L94
34	CartRemoveEntrySuccessEvent	CartRemoveEntrySuccessEvent	CartEvent	2			registerMapped ← CartActions.CART_REMOVE_ENTRY_SUCCESS feature-libs/cart/base/core/event/cart-event.builder.ts:L90-L93			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L96-L102
35	CartUpdateEntrySuccessEvent	CartUpdateEntrySuccessEvent	CartEvent	2			registerMapped ← CartActions.CART_UPDATE_ENTRY_SUCCESS feature-libs/cart/base/core/event/cart-event.builder.ts:L101-L104			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L104-L111
36	CartUpdateEntryFailEvent	CartUpdateEntryFailEvent	CartEvent	2			registerMapped ← CartActions.CART_UPDATE_ENTRY_FAIL feature-libs/cart/base/core/event/cart-event.builder.ts:L105-L108			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L113-L120
37	CartUiEventAddToCart	CartUiEventAddToCart	CxEvent	1					feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L311-L311	@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L122-L136
38	MergeCartSuccessEvent	MergeCartSuccessEvent	CartEvent	2			registerMapped ← CartActions.MERGE_CART_SUCCESS feature-libs/cart/base/core/event/cart-event.builder.ts:L112-L115			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L141-L157

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
39	LoadCartEvent	LoadCartEvent	CartEvent	2				feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L112-L112 feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L52-L52 feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L71-L71 feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L90-L90		@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L162-L167
40	RemoveCartEvent	RemoveCartEvent	CartEvent	2				feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L29-L29 feature-libs/checkout/scheduled-replenishment/root/events/checkout-scheduled-replenishment-event.listener.ts:L29-L29		@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L170-L175
41	DeleteCartEvent	DeleteCartEvent	CartEvent	2			register ← CartActions.DELETE_CART (factory) feature-libs/cart/base/core/event/cart-event.builder.ts:L137-L145			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L180-L185
42	DeleteCartSuccessEvent	DeleteCartSuccessEvent	CartEvent	2			register ← CartActions.DELETE_CART_SUCCESS (factory) feature-libs/cart/base/core/event/cart-event.builder.ts:L147-L155			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L187-L192
43	DeleteCartFailEvent	DeleteCartFailEvent	CartEvent	2			register ← CartActions.DELETE_CART_FAIL (factory) feature-libs/cart/base/core/event/cart-event.builder.ts:L157-L165			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L194-L199

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
44	AddCartVoucherEvent	AddCartVoucherEvent	CartEvent	2		2	register ← CartActions.CART_ADD_VOUCHER feature-libs/cart/base/core/event/cart-event.builder.ts:L169-L172			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L201-L207
45	AddCartVoucherSuccessEvent	AddCartVoucherSuccessEvent	AddCartVoucherEvent	3			register ← CartActions.CART_ADD_VOUCHER_SUCCESS feature-libs/cart/base/core/event/cart-event.builder.ts:L174-L177			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L209-L214
46	AddCartVoucherFailEvent	AddCartVoucherFailEvent	AddCartVoucherEvent	3			register ← CartActions.CART_ADD_VOUCHER_FAIL feature-libs/cart/base/core/event/cart-event.builder.ts:L179-L182			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L216-L222
47	RemoveCartVoucherEvent	RemoveCartVoucherEvent	CartEvent	2		2	register ← CartActions.CART_REMOVE_VOUCHER feature-libs/cart/base/core/event/cart-event.builder.ts:L186-L189			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L224-L230
48	RemoveCartVoucherSuccessEvent	RemoveCartVoucherSuccessEvent	RemoveCartVoucherEvent	3			register ← CartActions.CART_REMOVE_VOUCHER_SUCCESS feature-libs/cart/base/core/event/cart-event.builder.ts:L191-L194			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L232-L237
49	RemoveCartVoucherFailEvent	RemoveCartVoucherFailEvent	RemoveCartVoucherEvent	3			register ← CartActions.CART_REMOVE_VOUCHER_FAIL feature-libs/cart/base/core/event/cart-event.builder.ts:L196-L199			@spartacus/cart/base/root	feature-libs/cart/base/root/events/cart.events.ts:L239-L245
50	SavedCartEvent		CartEvent	2	sì	12				@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L12-L12
51	SaveCartEvent	SaveCartEvent	SavedCartEvent	3			register ← SavedCartActions.SAVE_CART (factory) feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L95-L104			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L16-L23
52	SaveCartSuccessEvent	SaveCartSuccessEvent	SavedCartEvent	3			buildSaveCartSuccessEvent ← SavedCartActions.SAVE_CART_SUCCESS feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L80-L83			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L25-L33

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
53	SaveCartFailEvent	SaveCartEvent	SavedCartEvent	3			register ← SavedCartActions.SAVE_CART_FAIL (factory) feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L85-L93			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L35-L42
54	RestoreSavedCartEvent	RestoreSavedCartEvent	SavedCartEvent	3			buildRestoreSavedCartEvents ← SavedCartActions.RESTORE_SAVED_CART feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L60-L63			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L44-L52
55	RestoreSavedCartSuccessEvent	RestoreSavedCartSuccessEvent	SavedCartEvent	3			buildRestoreSavedCartEvents ← SavedCartActions.RESTORE_SAVED_CART_SUCCESS feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L65-L68			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L54-L61
56	RestoreSavedCartFailEvent	RestoreSavedCartFailEvent	SavedCartEvent	3			buildRestoreSavedCartEvents ← SavedCartActions.RESTORE_SAVED_CART_FAIL feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L70-L73			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L63-L71
57	EditSavedCartEvent	EditSavedCartEvent	SavedCartEvent	3			register ← SavedCartActions.EDIT_SAVED_CART (factory) feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L126-L135			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L73-L80
58	EditSavedCartSuccessEvent	EditSavedCartSuccessEvent	SavedCartEvent	3			buildSaveCartSuccessEvent ← SavedCartActions.EDIT_SAVED_CART_SUCCESS feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L111-L114			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L81-L89
59	EditSavedCartFailEvent	EditSavedCartFailEvent	SavedCartEvent	3			register ← SavedCartActions.EDIT_SAVED_CART_FAIL (factory) feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L116-L124			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L90-L97
60	CloneSavedCartEvent	CloneSavedCartEvent	SavedCartEvent	3			buildRestoreSavedCartEvents ← SavedCartActions.CLONE_SAVED_CART feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L142-L145			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L99-L107

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
61	CloneSavedCartSuccessEvent	CloneSavedCartSuccessEvent	SavedCartEvent	3			buildRestoreSavedCartEvents ← SavedCartActions.CLONE_SAVED_CART_SUCCESS feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L147-L150			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L108-L115
62	CloneSavedCartFailEvent	CloneSavedCartFailEvent	SavedCartEvent	3			buildRestoreSavedCartEvents ← SavedCartActions.CLONE_SAVED_CART_FAIL feature-libs/cart/saved-cart/core/events/saved-cart-event.builder.ts:L152-L155			@spartacus/cart/saved-cart/root	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L116-L124
63	CheckoutCostCenterEvent		CheckoutEvent	2	sì	1				@spartacus/checkout/b2b/root	feature-libs/checkout/b2b/root/events/checkout-b2b.events.ts:L12-L12
64	CheckoutCostCenterSetEvent	CheckoutCostCenterSetEvent	CheckoutCostCenterEvent	3				feature-libs/checkout/b2b/core/facade/checkout-cost-center.service.ts:L39-L39		@spartacus/checkout/b2b/root	feature-libs/checkout/b2b/root/events/checkout-b2b.events.ts:L17-L27
65	CheckoutPaymentTypeEvent		CheckoutEvent	2	sì	3				@spartacus/checkout/b2b/root	feature-libs/checkout/b2b/root/events/checkout-b2b.events.ts:L32-L32
66	CheckoutPaymentTypesQueryReloadEvent	CheckoutPaymentTypesQueryReloadEvent	CheckoutPaymentTypeEvent	3				feature-libs/checkout/b2b/root/events/checkout-payment-type-event.listener.ts:L59-L59		@spartacus/checkout/b2b/root	feature-libs/checkout/b2b/root/events/checkout-b2b.events.ts:L37-L42
67	CheckoutPaymentTypesQueryResetEvent	CheckoutPaymentTypesQueryResetEvent	CheckoutPaymentTypeEvent	3				feature-libs/checkout/b2b/root/events/checkout-payment-type-event.listener.ts:L70-L70		@spartacus/checkout/b2b/root	feature-libs/checkout/b2b/root/events/checkout-b2b.events.ts:L47-L52
68	CheckoutPaymentTypeSetEvent	CheckoutPaymentTypeSetEvent	CheckoutPaymentTypeEvent	3				feature-libs/checkout/b2b/core/facade/checkout-payment-type.service.ts:L70-L70 integration-libs/opf/b2b-checkout/core/facade/opf-b2b-checkout-payment-type.service.ts:L56-L56		@spartacus/checkout/b2b/root	feature-libs/checkout/b2b/root/events/checkout-b2b.events.ts:L57-L72

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
69	CheckoutQueryReloadEvent	CheckoutQueryReloadEvent	CxEvent	1				feature-libs/checkout/base/root/events/checkout-query-event.listener.ts:L44-L44		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L12-L17

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
70	CheckoutQueryResetEvent	CheckoutQueryResetEvent	CxEvent	1				feature-libs/checkout/b2b/root/events/checkout-cost-center-event.listener.ts:L35-L35 feature-libs/checkout/b2b/root/events/checkout-payment-type-event.listener.ts:L48-L48 feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L119-L119 feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L104-L104 feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L119-L119 feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L129-L129 feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L139-L139 feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L50-L50 feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L69-L69 feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L22-L27

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
								ts:L88-L88 feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L57-L57 feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L65-L65 feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L38-L38 feature-libs/checkout/base/root/events/checkout-query-event.listener.ts:L59-L59 feature-libs/checkout/scheduled-replenishment/root/events/checkout-scheduled-replenishment-event.listener.ts:L38-L38 integration-libs/s4-service/root/events/checkout-service-details-event.listener.ts:L26-L26			
71	Checkout Event		CxEvent	1	sì	6				@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L32-L48
72	Checkout Delivery Address Event		CheckoutEvent	2	sì	3				@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L53-L53
73	Checkout Delivery Address Created Event	Checkout Delivery Address Created Event	Checkout Delivery Address Event	3				feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L53-L53		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L58-L67

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
74	CheckoutDeliveryAddressSetEvent	CheckoutDeliveryAddressSetEvent	CheckoutDeliveryAddressEvent	3				feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L79-L79		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L72-L81
75	CheckoutDeliveryAddressClearedEvent	CheckoutDeliveryAddressClearedEvent	CheckoutDeliveryAddressEvent	3				feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L105-L105		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L86-L91
76	CheckoutDeliveryModeEvent		CheckoutEvent	2	sì	5				@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L96-L96
77	CheckoutDeliveryModeSetEvent	CheckoutDeliveryModeSetEvent	CheckoutDeliveryModeEvent	3				feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L77-L77		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L101-L110
78	CheckoutDeliveryModeClearedEvent	CheckoutDeliveryModeClearedEvent	CheckoutDeliveryModeEvent	3				feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L100-L100		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L115-L120
79	CheckoutDeliveryModeClearedErrorEvent	CheckoutDeliveryModeClearedErrorEvent	CheckoutDeliveryModeEvent	3				feature-libs/checkout/base/core/facade/checkout-delivery-modes.service.ts:L114-L114		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L125-L130
80	CheckoutSupportedDeliveryModesQueryReloadEvent	CheckoutSupportedDeliveryModesQueryReloadEvent	CheckoutDeliveryModeEvent	3				feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L134-L134 feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L151-L151		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L135-L140

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
81	CheckoutSupportedDeliveryModesQueryResetEvent	CheckoutSupportedDeliveryModesQueryResetEvent	CheckoutDeliveryModeEvent	3				feature-libs/checkout/b2b/root/events/checkout-cost-center-event.listener.ts:L31-L31 feature-libs/checkout/b2b/root/events/checkout-payment-type-event.listener.ts:L44-L44 feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L114-L114 feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L77-L77 feature-libs/checkout/base/root/events/checkout-delivery-address-event.listener.ts:L99-L99 feature-libs/checkout/base/root/events/checkout-delivery-mode-event.listener.ts:L148-L148		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L145-L150
82	CheckoutPaymentDetailsEvent		CheckoutEvent	2	sì	4				@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L155-L155
83	CheckoutPaymentDetailsCreatedEvent	CheckoutPaymentDetailsCreatedEvent	CheckoutPaymentDetailsEvent	3				feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L68-L68		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L160-L169
84	CheckoutPaymentDetailsSetEvent	CheckoutPaymentDetailsSetEvent	CheckoutPaymentDetailsEvent	3				feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L95-L95		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L174-L183

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
85	CheckoutPaymentCardTypesQueryReloadEvent	CheckoutPaymentCardTypesQueryReloadEvent	CheckoutPaymentDetailsEvent	3				feature-libs/checkout/base/root/events/checkout-payment-event.listener.ts:L76-L76		@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L188-L193
86	CheckoutPaymentCardTypesQueryResetEvent	CheckoutPaymentCardTypesQueryResetEvent	CheckoutPaymentDetailsEvent	3						@spartacus/checkout/base/root	feature-libs/checkout/base/root/events/checkout.events.ts:L198-L203
87	GetTicketQueryResetEvent	GetTicketQueryResetEvent	CxEvent	1				feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L104-L104 feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L82-L82		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L9-L11

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
88	GetTicketQueryReloadEvent	GetTicketQueryReloadEvent	CxEvent	1				feature-libs/customer-ticketing/components/details/customer-ticketing-details/customer-ticketing-details.component.ts:L92-L92 feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L118-L118 feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L132-L132 feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L71-L71 feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L96-L96		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L13-L15
89	GetTicketsQueryResetEvents	GetTicketsQueryResetEvents	CxEvent	1				feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L83-L83		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L17-L19
90	GetTicketsQueryReloadEvents	GetTicketsQueryReloadEvents	CxEvent	1				feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L59-L59 feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L70-L70		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L21-L23

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
91	New Message Event	New Message Event	CxEvent	1				feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L128-L128		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L25-L27
92	Ticket Reopened Event	Ticket ReopenedEvent	CxEvent	1				feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L126-L126		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L29-L31
93	Ticket Closed Event	Ticket Closed Event	CxEvent	1				feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L120-L120		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L33-L35
94	Get Ticket Category Query Reset Event	Get Ticket CategoryQuery Reset Event	CxEvent	1				feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L84-L84		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L37-L39
95	Get Ticket Category Query Reload Event	Get Ticket CategoryQuery Reload Event	CxEvent	1						@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L41-L43
96	Get Ticket Associated Objects Query Reset Event	Get Ticket Associated Objects Query Reset Event	CxEvent	1				feature-libs/customer-ticketing/root/events/customer-ticketing-event.listener.ts:L85-L85		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L45-L47
97	Get Ticket Associated Objects Query Reload Event	Get Ticket Associated Objects Query Reload Event	CxEvent	1						@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L49-L51
98	Ticket Created Event	Ticket Created Event	CxEvent	1				feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L97-L97		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L52-L54

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
99	CreateEvent	CreateEvent	CxEvent	1						@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L55-L57
100	UploadAttachmentSuccessEvent	UploadAttachmentSuccessEvent	CxEvent	1				feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L161-L161		@spartacus/customer-ticketing/root	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L59-L61
101	OrderEvent		CxEvent	1	sì	2				@spartacus/order/root	feature-libs/order/root/events/order.events.ts:L14-L30
102	OrderPlacedEvent	OrderPlacedEvent	OrderEvent	2				feature-libs/order/core/facade/order.service.ts:L140-L140 feature-libs/order/core/facade/order.service.ts:L34-L34 feature-libs/order/core/facade/order.service.ts:L66-L66		@spartacus/order/root	feature-libs/order/root/events/order.events.ts:L35-L44
103	ReplenishmentOrderScheduledEvent	ReplenishmentOrderScheduledEvent	OrderEvent	2				feature-libs/order/core/facade/scheduled-replenishment-order.service.ts:L48-L48		@spartacus/order/root	feature-libs/order/root/events/order.events.ts:L49-L58
104	DownloadOrderInvoicesEvent	DownloadOrderInvoicesEvent	CxEvent	1					feature-libs/order/components/order-details/my-account-v2/order-details-actions/my-account-v2-order-details-actions.component.ts:L32-L32	@spartacus/order/root	feature-libs/order/root/events/order.events.ts:L63-L72

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
10	Quote Details Reload Query Event	Quote Details Reload Query Event	CxEvent	1				feature-libs/quote/components/comments/quote-comments.component.ts:L73-L73 feature-libs/quote/components/links/quote-links.component.ts:L49-L49 feature-libs/quote/components/links/quote-links.component.ts:L58-L58 feature-libs/quote/core/event/quote-cart-event.listener.ts:L48-L48 feature-libs/quote/core/facade/quote.service.ts:L122-L122 feature-libs/quote/core/facade/quote.service.ts:L151-L151 feature-libs/quote/core/facade/quote.service.ts:L211-L211 feature-libs/quote/core/facade/quote.service.ts:L359-L359 feature-libs/quote/core/facade/quote.service.ts:L442-L442		@spartacus/quote/core	feature-libs/quote/core/event/quote.events.ts:L9-L14

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
106	GetSubscriptionByCodeReloadEvent	GetSubscriptionByCodeReloadEvent	CxEvent	1				feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal-component.service.ts:L51-L51 feature-libs/subscription-billing/components/details/subscription-details.component.ts:L52-L52 feature-libs/subscription-billing/root/events/subscription-billing-event.listener.ts:L36-L36		@spartacus/subscription-billing/root	feature-libs/subscription-billing/root/events/subscription-billing.events.ts:L9-L11
107	GetSubscriptionListReloadEvent	GetSubscriptionListReloadEvent	CxEvent	1				feature-libs/subscription-billing/root/events/subscription-billing-event.listener.ts:L37-L37		@spartacus/subscription-billing/root	feature-libs/subscription-billing/root/events/subscription-billing.events.ts:L13-L15
108	GetSubscriptionBillsListReloadEvent	GetSubscriptionBillsListReloadEvent	CxEvent	1				feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L205-L205		@spartacus/subscription-billing/root	feature-libs/subscription-billing/root/events/subscription-billing.events.ts:L17-L19
109	GetSubscriptionBillByCodeReloadEvent	GetSubscriptionBillByCodeReloadEvent	CxEvent	1				feature-libs/subscription-billing/components/billing-details/subscription-billing-details.component.ts:L45-L45		@spartacus/subscription-billing/root	feature-libs/subscription-billing/root/events/subscription-billing.events.ts:L21-L23
110	UserAccountEvent		CxEvent	1	sì	1				@spartacus/user/account/root	feature-libs/user/account/root/events/user-account.events.ts:L10-L12
111	UserAccountChangedEvent	UserAccountChangedEvent	UserAccountEvent	2				feature-libs/user/profile/core/facade/user-profile.service.ts:L37-L37		@spartacus/user/account/root	feature-libs/user/account/root/events/user-account.events.ts:L14-L16

#	CLASSE	STATIC TYPE	CLASSE PADRE	PROF.	ABSTRACT	FIGLI	REGISTRAZIONI	DISPATCH	NEW	LIBRERIA	RIFERIMENTO
1 1 2	CdcLoadUserTokenFailEvent	CdcLoadUserTokenFailEvent	CxEvent	1			register ← CdcAuthActions.LOAD_CDC_USER_TOKEN_FAIL integration-libs/cdc/core/events/cdc-event.builder.ts:L34-L37			@spartacus/cdc/root	integration-libs/cdc/root/events/cdc-event.ts:L13-L18
1 1 3	CdcReConsentEvent	CdcReConsentEvent	CxEvent	1					integration-libs/cdc/root/service/cdc-js.service.ts:L840-L840	@spartacus/cdc/root	integration-libs/cdc/root/events/cdc-event.ts:L20-L31
1 1 4	CheckoutServiceDetailsEvent		CheckoutEvent	2	sì	1				@spartacus/s4-service/root	integration-libs/s4-service/root/events/checkout-service-details.events.ts:L13-L13
1 1 5	CheckoutServiceDetailsSetEvent	CheckoutServiceDetailsSetEvent	CheckoutServiceDetailsEvent	3				integration-libs/s4-service/checkout/core/facade/checkout-service-details.service.ts:L50-L50		@spartacus/s4-service/root	integration-libs/s4-service/root/events/checkout-service-details.events.ts:L18-L28

A08

Outlet

Enum `*Outlets` e nomi degli outlet

IN QUESTO CAPITOLO

Conteggi

Come rigenerare

Tabella 1 — Enum *Outlets

Tabella 2 — Nomi outlet nei template HTML

Tabella 3 — provideOutlet(...)

Generata da `tools/extract2/a08_outlets.py` allo SHA 5b7a1c2b. Non modificare a mano: rigenerare.

Tabella 1: membri degli `enum` il cui nome termina con `Outlets` (es. `CartOutlets`, `ProductListOutlets`), incluse le augmentation in `declare module`. `OutletPosition` è escluso (non è un elenco di nomi outlet).

Tabella 2: usi della direttiva outlet nei template `.html` (`git ls-files ... | grep '\.html$'`): `cxOutlet="nome"` (attributo statico), `[cxOutlet]="expr"`, `*cxOutlet="expr"`. Le espressioni `Enum.MEMBRO` o `proprietà.MEMBRO` (es. `cartOutlets.X` dove `cartOutlets = CartOutlets`) sono risolte sul valore dell'enum (per nome enum case-insensitive, altrimenti per nome membro se univoco). I riferimenti HTML sono nel formato `file.html:Lx-Lx`.

Tabella 3: chiamate `provideOutlet({ id, component, position })`.

Conteggi

- Enum `*Outlets` (incluse augmentation): **9**
- Membri totali negli enum: **38**
- File `.html` in ambito: **533**
- Usi `cxOutlet` nei template (statico, `[cxOutlet]`, `*cxOutlet`; commenti HTML esclusi): **85**
- Usi `cxOutletRef` nei template: **0**
- Nomi outlet distinti risolti nei template: **36**
- Usi non risolvibili staticamente (espressioni dinamiche): **17**
- Chiamate `provideOutlet({...})`: **35**
- Valori enum mai usati né in template né in `provideOutlet`: **3**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract2/a08_outlets.py
# oppure tutte le appendici: bash tools/extract2/run_all.sh
```

Ambito: `git ls-files core-libs feature-libs integration-libs | grep '\.ts$' | grep -vE '\.spec\.ts$|/testing/|\.mock\.ts$|/schematics/'`. La colonna "Libreria" è l'entry point npm ricavato dal `package.json`/`ng-package.json` più vicino.

Tabella 1 — Enum `*Outlets`

#	ENUM	MEMBRO	VALORE (NOME OUTLET)	USI TEMPLATE	PROVIDEOUTLET	AUGMENTATION DI	LIBRERIA	RIFERIMENTO
1	CartOutlets	ITEM	cx-cart-item	1			@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L11-L11

#	ENUM	MEMBRO	VALORE (NOME OUTLET)	USI TEMPLATE	PROVIDEOUTLET	AUGMENTATION DI	LIBRERIA	RIFERIMENTO
2	CartOutlets	LIST_ITEM	cx-cart-item-list-row	1			@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L12-L12
3	CartOutlets	ITEM_DETAILS	cx-cart-item.details	2	3		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L13-L13
4	CartOutlets	ITEM_CONFIGURATOR_ISSUES	cx-configurator-issues-notification	2	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L14-L14
5	CartOutlets	ITEM_VALIDATION_QUANTITY_HINT	cx-cart-item-validation-quantity-hint	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L15-L15
6	CartOutlets	ITEM_BUNDLE_DETAILS	cx-cart-item.bundle-details	2	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L16-L16
7	CartOutlets	ITEM_DELIVERY_DETAILS	cx-cart-item.delivery-details	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L17-L17
8	CartOutlets	ORDER_SUMMARY	cx-order-summary	3	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L18-L18
9	CartOutlets	CART_ITEM_LIST	cx-cart-item-list	17	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L19-L19
10	CartOutlets	ADD_TO_CART_CONTAINER	cx-add-to-cart-container				@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L20-L20
11	CartOutlets	PICKUP_INFO	cx-pickup-info		1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L21-L21
12	CartOutlets	ADD_TO_CART_PICKUP_OPTION	cx-add-to-cart-pickup-option	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L22-L22
13	CartOutlets	DELIVERY_MODE	cx-delivery-mode	6	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L23-L23
14	CartOutlets	ORDER_OVERVIEW	cx-order-overview	3	2		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L24-L24
15	CartOutlets	CPQ_QUOTE_MODULE	cx-cpq-quote	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L25-L25
16	CartOutlets	CPQ_QUOTE_HEADING	cx-cpq-quote-heading	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L26-L26
17	CartOutlets	CPQ_QUOTE	cx-cpq-quote-offer	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L27-L27
18	CartOutlets	SUBSCRIPTION_PRICE_HEADING	cx-subscription-cart-price-heading	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L28-L28
19	CartOutlets	SUBSCRIPTION_PRICE_BODY	cx-subscription-cart-price-body	2	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L29-L29

#	ENUM	MEMBRO	VALORE (NOME OUTLET)	USI TEMPLATE	PROVIDEOUTLET	AUGMENTATION DI	LIBRERIA	RIFERIMENTO
20	CartOutlets	ORDER_DETAILS_AFTER_PAYMENT_METHOD	cx-order-details-payment-method-details	1	1		@spartacus/cart/base/root	feature-libs/cart/base/root/models/cart-outlets.model.ts:L30-L30
21	CartOutlets	OPF_CHECKOUT_PICKUP_ITEMS	cx-opf-checkout-pickup-items	1		@spartacus/cart/base/root	@spartacus/opf/checkout/root	integration-libs/opf/checkout/root/model/augmented-cart-outlets.model.ts:L11-L11
22	OpfCheckoutOutlets	OPF_CHECKOUT_BEFORE_PAYMENT_OPTIONS	cx-opf-checkout-before-payment-options	1	3		@spartacus/opf/checkout/root	integration-libs/opf/checkout/root/model/opf-checkout-outlets.model.ts:L8-L8
23	OpfCheckoutOutlets	OPF_CHECKOUT_BEFORE_SAVED_PAYMENT_OPTIONS	cx-opf-checkout-before-saved-payment-options	1	1		@spartacus/opf/checkout/root	integration-libs/opf/checkout/root/model/opf-checkout-outlets.model.ts:L9-L9
24	OpfCheckoutOutlets	ORDER_DETAILS_AFTER_PAYMENT_OPTIONS	cx-opf-after-payment-options	1	1		@spartacus/opf/checkout/root	integration-libs/opf/checkout/root/model/opf-checkout-outlets.model.ts:L10-L10
25	Order Outlets	ORDER_CONSIGNMENT	cx-order-consignment	1	1		@spartacus/order/root	feature-libs/order/root/model/order-outlets.model.ts:L11-L11
26	Order Outlets	CONSIGNMENT_DELIVERY_INFO	cx-order-consignment-delivery-info	1	1		@spartacus/order/root	feature-libs/order/root/model/order-outlets.model.ts:L12-L12
27	Order Outlets	SERVICE_DETAILS	cx-service-details-card	1	1		@spartacus/order/root	feature-libs/order/root/model/order-outlets.model.ts:L13-L13
28	Product Detail Outlets	INTRO	PDP.INTRO				@spartacus/storefront	core-libs/storefront/cms-components/product/product-outlets.model.ts:L8-L8
29	Product Detail Outlets	PRICE	PDP.PRICE	1			@spartacus/storefront	core-libs/storefront/cms-components/product/product-outlets.model.ts:L9-L9
30	Product Detail Outlets	SHARE	PDP.SHARE				@spartacus/storefront	core-libs/storefront/cms-components/product/product-outlets.model.ts:L10-L10
31	Product Detail Outlets	SUMMARY	PDP.SUMMARY	1			@spartacus/storefront	core-libs/storefront/cms-components/product/product-outlets.model.ts:L11-L11
32	ProductList Outlets	ITEM_ACTIONS	cx-product-list-item.actions	2	1		@spartacus/storefront	core-libs/storefront/cms-components/product/product-outlets.model.ts:L14-L14
33	ProductList Outlets	ITEM_DETAILS	cx-product-list-item.details	2	2		@spartacus/storefront	core-libs/storefront/cms-components/product/product-outlets.model.ts:L15-L15
34	SearchBox Outlets	RECENT_SEARCHES	SearchBoxOutlets.RECENT_SEARCHES	2	1		@spartacus/storefront	core-libs/storefront/cms-components/navigation/search-box/search-box-outlets.model.ts:L8-L8

#	ENUM	MEMBRO	VALORE (NOME OUTLET)	USI TEMPLATE	PROVIDEOUTLET	AUGMENTATION DI	LIBRERIA	RIFERIMENTO
35	SearchBox Outlets	RECENT_SEARCHES_ HEADER	SearchBoxOutlets. RECENT_SEARCHES_ HEADER	1	1		@spartacus/ storefront	core-libs/storefront/cms-components/navigation/search-box/search-box-outlets.model.ts:L9-L9
36	SearchBox Outlets	TRENDING_SEARCHES	SearchBoxOutlets. TRENDING_SEARCHES	1	1		@spartacus/ storefront	core-libs/storefront/cms-components/navigation/search-box/search-box-outlets.model.ts:L10-L10
37	StoreFinder Outlets	PREFERRED_STORE	cx-pick-up-in-store-make-my-store	1	1		@spartacus/ storefinder/ root	feature-libs/storefinder/root/model/store-entities.model.ts:L10-L10
38	Storefront Outlets	STOREFRONT	cx-storefront	1			@spartacus/ storefront	core-libs/storefront/layout/main/storefront-outlets.model.ts:L8-L8

Tabella 2 — Nomi outlet nei template HTML

#	NOME OUTLET (VALORE)	USI	ESPRESSIONI SORGENTE	FILE	RIFERIMENTI
1	cx-cart-item-list	17	[cxOutlet]="CartOutlets.CART_ITEM_LIST" [cxOutlet]="cartOutlets.CART_ITEM_LIST"	15	feature-libs/cart/saved-cart/components/details/saved-cart-details-items/saved-cart-details-items.component.html:L5-L5 feature-libs/checkout/b2b/components/checkout-review-submit/checkout-review-submit.component.html:L205-L205 feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.html:L127-L127 feature-libs/checkout/base/components/checkout-review-submit/checkout-review-submit.component.html:L132-L132 feature-libs/checkout/base/components/checkout-review/checkout-review-shipping/checkout-review-shipping.component.html:L55-L55 feature-libs/order/components/order-confirmation/order-confirmation-items/order-confirmation-items.component.html:L12-L12 feature-libs/order/components/order-confirmation/order-confirmation-shipping/order-confirmation-shipping.component.html:L34-L34 feature-libs/order/components/order-details/order-detail-items/order-consigned-entries/order-consigned-entries.component.html:L29-L29 feature-libs/order/components/order-details/order-detail-items/order-detail-items.component.html:L71-L71 feature-libs/order/components/order-details/order-detail-items/order-detail-items.component.html:L96-L96 feature-libs/order/components/order-details/order-detail-items/order-detail-items.component.html:L128-L128 feature-libs/quote/components/items/quote-items.component.html:L27-L27 integration-libs/opf/checkout/components/opf-checkout-review-cart-details/opf-checkout-review-cart-details.component.html:L47-L47 integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.component.html:L66-L66

#	NOME OUTLET (VALORE)	USI	ESPRESSIONI SORGENTE	FILE	RIFERIMENTI
					integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.component.html:L202-L202 integration-libs/s4-service/order/components/cancel-service-order-headline/cancel-service-order-headline.component.html:L32-L32 integration-libs/s4-service/order/components/reschedule-service-order/reschedule-service-order.component.html:L56-L56
2	cx-delivery-mode	6	[cxOutlet]="CartOutlets.DELIVERY_MODE" [cxOutlet]="cartOutlets.DELIVERY_MODE"	5	feature-libs/checkout/b2b/components/checkout-review-submit/checkout-review-submit.component.html:L138-L138 feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.html:L55-L55 feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.html:L112-L112 feature-libs/checkout/base/components/checkout-review-submit/checkout-review-submit.component.html:L65-L65 integration-libs/s4-service/checkout/components/checkout-delivery-mode/service-checkout-delivery-mode.component.html:L51-L51 integration-libs/s4-service/checkout/components/checkout-review-submit/service-checkout-review-submit.component.html:L158-L158
3	(dinamico) getDataOutletRef(cell)	6	[cxOutlet]="getDataOutletRef(cell)"	1	core-libs/storefront/shared/components/table/table.component.html:L28-L28 core-libs/storefront/shared/components/table/table.component.html:L62-L62 core-libs/storefront/shared/components/table/table.component.html:L87-L87 core-libs/storefront/shared/components/table/table.component.html:L119-L119 core-libs/storefront/shared/components/table/table.component.html:L151-L151 core-libs/storefront/shared/components/table/table.component.html:L176-L176

#	NOME OUTLET (VALORE)	USI	ESPRESSIONI SORGENTE	FILE	RIFERIMENTI
4	(dinamico) getHeaderOutletRef(cell)	6	[cxOutlet]="getHeaderOutletRef(cell)"	1	core-libs/storefront/shared/components/table/table.component.html:L20-L20 core-libs/storefront/shared/components/table/table.component.html:L45-L45 core-libs/storefront/shared/components/table/table.component.html:L75-L75 core-libs/storefront/shared/components/table/table.component.html:L111-L111 core-libs/storefront/shared/components/table/table.component.html:L135-L135 core-libs/storefront/shared/components/table/table.component.html:L164-L164
5	cx-order-overview	3	[cxOutlet]="cartOutlets.ORDER_OVERVIEW"	1	feature-libs/order/components/order-details/order-overview/order-overview.component.html:L43-L43 feature-libs/order/components/order-details/order-overview/order-overview.component.html:L71-L71 feature-libs/order/components/order-details/order-overview/order-overview.component.html:L164-L164
6	cx-order-summary	3	[cxOutlet]="CartOutlets.ORDER_SUMMARY" [cxOutlet]="cartOutlets.ORDER_SUMMARY"	3	feature-libs/checkout/base/components/checkout-order-summary/checkout-order-summary.component.html:L2-L2 feature-libs/order/components/order-confirmation/order-confirmation-totals/order-confirmation-totals.component.html:L5-L5 feature-libs/order/components/order-details/order-detail-totals/order-detail-totals.component.html:L5-L5
7	SearchBoxOutlets.RECENT_SEARCHES	2	[cxOutlet]="searchBoxOutlets.RECENT_SEARCHES"	1	core-libs/storefront/cms-components/navigation/search-box/search-box.component.html:L143-L143 core-libs/storefront/cms-components/navigation/search-box/search-box.component.html:L155-L155
8	cx-cart-item.bundle-details	2	[cxOutlet]="CartOutlets.ITEM_BUNDLE_DETAILS"	2	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L57-L57 feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.html:L141-L141

#	NOME OUTLET (VALORE)	USI	ESPRESSIONI SORGENTE	FILE	RIFERIMENTI
9	cx-cart-item.details	2	[cxOutlet]="CartOutlets.ITEM_DETAILS"	2	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L54-L54 feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.html:L37-L37
10	cx-configurator-issues-notification	2	[cxOutlet]="CartOutlets.ITEM_CONFIGURATOR_ISSUES"	2	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L4-L4 feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.html:L3-L3
11	cx-product-list-item.actions	2	[cxOutlet]="ProductListOutlets.ITEM_ACTIONS"	2	core-libs/storefront/cms-components/product/product-list/product-grid-item/product-grid-item.component.html:L43-L43 core-libs/storefront/cms-components/product/product-list/product-list-item/product-list-item.component.html:L66-L66
12	cx-product-list-item.details	2	[cxOutlet]="ProductListOutlets.ITEM_DETAILS"	2	core-libs/storefront/cms-components/product/product-list/product-grid-item/product-grid-item.component.html:L23-L23 core-libs/storefront/cms-components/product/product-list/product-list-item/product-list-item.component.html:L33-L33
13	cx-subscription-cart-price-body	2	[cxOutlet]="CartOutlets.SUBSCRIPTION_PRICE_BODY"	2	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L98-L98 feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.html:L60-L60
14	PDP.PRICE	1	[cxOutlet]="outlets.PRICE"	1	core-libs/storefront/cms-components/product/product-summary/product-summary.component.html:L7-L7
15	PDP.SUMMARY	1	[cxOutlet]="outlets.SUMMARY"	1	core-libs/storefront/cms-components/product/product-summary/product-summary.component.html:L19-L19
16	SearchBoxOutlets.RECENT_SEARCHES_HEADER	1	[cxOutlet]="searchBoxOutlets.RECENT_SEARCHES_HEADER"	1	core-libs/storefront/cms-components/navigation/search-box/search-box.component.html:L295-L295
17	SearchBoxOutlets.TRENDING_SEARCHES	1	[cxOutlet]="searchBoxOutlets.TRENDING_SEARCHES"	1	core-libs/storefront/cms-components/navigation/search-box/search-box.component.html:L132-L132

#	NOME OUTLET (VALORE)	USI	ESPRESSIONI SORGENTE	FILE	RIFERIMENTI
18	cx-add-to-cart-pickup-option	1	[cxOutlet]="CartOutlets.ADD_TO_CART_PICKUP_OPTION"	1	feature-libs/cart/base/components/add-to-cart/add-to-cart.component.html:L25-L25
19	cx-cart-item	1	[cxOutlet]="CartOutlets.ITEM"	1	feature-libs/cart/base/components/cart-shared/cart-item/cart-item.component.html:L2-L2
20	cx-cart-item-list-row	1	[cxOutlet]="CartOutlets.LIST_ITEM"	1	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L2-L2
21	cx-cart-item-validation-quantity-hint	1	[cxOutlet]="CartOutlets.ITEM_VALIDATION_QUANTITY_HINT"	1	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L134-L134
22	cx-cart-item.delivery-details	1	[cxOutlet]="CartOutlets.ITEM_DELIVERY_DETAILS"	1	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L65-L65
23	cx-cpq-quote	1	[cxOutlet]="CartOutlets.CPQ_QUOTE_MODULE"	1	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L108-L108
24	cx-cpq-quote-heading	1	[cxOutlet]="CartOutlets.CPQ_QUOTE_HEADING"	1	feature-libs/cart/base/components/cart-shared/cart-item-list/cart-item-list.component.html:L22-L22
25	cx-cpq-quote-offer	1	[cxOutlet]="CartOutlets.CPQ_QUOTE"	1	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.html:L45-L45
26	cx-footer	1	cxOutlet="cx-footer"	1	core-libs/storefront/layout/main/storefront.component.html:L23-L23
27	cx-header	1	cxOutlet="cx-header"	1	core-libs/storefront/layout/main/storefront.component.html:L2-L2
28	cx-opf-after-payment-options	1	[cxOutlet]="opfCheckoutOutlets.ORDER_DETAILS_AFTER_PAYMENT_OPTIONS"	1	integration-libs/opf/payment/root/components/opf-payment-method-details/opf-payment-method-details.component.html:L11-L11
29	cx-opf-checkout-before-payment-options	1	[cxOutlet]="opfCheckoutOutlets.OPF_CHECKOUT_BEFORE_PAYMENT_OPTIONS"	1	integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.html:L21-L21
30	cx-opf-checkout-before-saved-payment-options	1	[cxOutlet]="opfCheckoutOutlets.OPF_CHECKOUT_BEFORE_SAVED_PAYMENT_OPTIONS"	1	integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.html:L64-L64

#	NOME OUTLET (VALORE)	USI	ESPRESSIONI SORGENTE	FILE	RIFERIMENTI
31	cx-opf-checkout-pickup-items	1	[cxOutlet]="cartOutlets.OPF_CHECKOUT_PICKUP_ITEMS"	1	integration-libs/opf/checkout/components/opf-checkout-review-cart-details/opf-checkout-review-cart-details.component.html:L20-L20
32	cx-order-consignment	1	[cxOutlet]="OrderOutlets.ORDER_CONSIGNMENT"	1	feature-libs/order/components/order-details/order-detail-items/order-consigned-entries/order-consigned-entries.component.html:L3-L3
33	cx-order-consignment-delivery-info	1	[cxOutlet]="OrderOutlets.CONSIGNMENT_DELIVERY_INFO"	1	feature-libs/order/components/order-details/order-detail-items/order-detail-items.component.html:L35-L35
34	cx-order-details-payment-method-details	1	[cxOutlet]="cartOutlets.ORDER_DETAILS_AFTER_PAYMENT_METHOD"	1	feature-libs/order/components/order-details/order-detail-billing/order-detail-billing.component.html:L10-L10
35	cx-pick-up-in-store-make-my-store	1	[cxOutlet]="StoreFinderOutlets.PREFERRED_STORE"	1	feature-libs/storefinder/components/store-finder-list-item/store-finder-list-item.component.html:L42-L42
36	cx-service-details-card	1	[cxOutlet]="orderOutlets.SERVICE_DETAILS"	1	feature-libs/order/components/order-details/order-overview/order-overview.component.html:L113-L113
37	cx-storefront	1	[cxOutlet]="StorefrontOutlets.STOREFRONT"	1	core-libs/storefront/layout/main/storefront.component.html:L1-L1
38	cx-subscription-cart-price-heading	1	[cxOutlet]="CartOutlets.SUBSCRIPTION_PRICE_HEADING"	1	feature-libs/cart/base/components/cart-shared/cart-item-list/cart-item-list.component.html:L14-L14
39	(dinamico) child.flexType	1	[cxOutlet]="child.flexType"	1	core-libs/storefront/cms-components/product/product-tabs/product-details-tab/product-details-tab.component.html:L7-L7
40	(dinamico) component.flexType	1	[cxOutlet]="component.flexType"	1	core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L9-L9
41	(dinamico) component?.flexType	1	[cxOutlet]="component?.flexType"	1	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.html:L5-L5
42	(dinamico) layoutName	1	[cxOutlet]="layoutName"	1	core-libs/storefront/cms-structure/page/page-layout/page-layout.component.html:L4-L4
43	(dinamico) position	1	[cxOutlet]="position"	1	core-libs/storefront/cms-structure/page/slot/page-slot.component.html:L3-L3

Tabella 3 — `provideOutlet(...)`

#	ID	NOME OUTLET RISOLTO	COMPONENT	POSITION	ALTRE CHIAVI	CONTENITORE	LIBRERIA	RIFERIMENTO
1	CartOutlets.ORDER_SUMMARY	cx-order-summary	OrderSummaryComponent	(default)		CartSharedModule	@spartacus/cart/base/components	feature-libs/cart/base/components/cart-shared/cart-shared.module.ts:L55-L58
2	CartOutlets.CART_ITEM_LIST	cx-cart-item-list	CartItemListComponent	(default)		CartSharedModule	@spartacus/cart/base/components	feature-libs/cart/base/components/cart-shared/cart-shared.module.ts:L59-L62
3	CartOutlets.ITEM_VALIDATION_QUANTITY_HINT	cx-cart-item-validation-quantity-hint	CartItemValidationQuantityHintComponent	OutletPosition.REPLACE		CartItemValidationQuantityHintModule	@spartacus/cart/base/components	feature-libs/cart/base/components/validation/cart-item-quantity-hint/cart-item-validation-quantity-hint.module.ts:L16-L20
4	CartOutlets.ITEM_DETAILS	cx-cart-item.details	EstimatedDeliveryDateComponent	OutletPosition.AFTER		ShowEstimatedDeliveryDateModule	@spartacus/estimated-delivery-date/show-estimated-delivery-date	feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L30-L34
5	OrderOutlets.CONSIGNMENT_DELIVERY_INFO	cx-order-consignment-delivery-info	OrderConfirmationShippingComponent	(default)		OrderConfirmationModule	@spartacus/order/components	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L136-L139
6	CartOutlets.ITEM_DELIVERY_DETAILS	cx-cart-item.delivery-details	CartPickupOptionsContainerComponent	OutletPosition.REPLACE		CartPickupOptionsContainerModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.module.ts:L23-L27
7	CartOutlets.ADD_TO_CART_PICKUP_OPTION	cx-add-to-cart-pickup-option	PdpPickupOptionsContainerComponent	OutletPosition.REPLACE		PdpPickupOptionsContainerModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.module.ts:L23-L27
8	OrderOutlets.ORDER_CONSIGNMENT	cx-order-consignment	PickupInStoreOrderConsignmentContainerComponent	OutletPosition.AFTER		OrderConsignmentContainerModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/pickup-in-store-order-consignment/pickup-in-store-order-consignment-container.module.ts:L22-L26
9	CartOutlets.PICKUP_INFO	cx-pickup-info	PickupInfoContainerComponent	OutletPosition.REPLACE		PickupInfoContainerModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/pickup-info-container/pickup-info-container.module.ts:L19-L23
10	StoreFinderOutlets.PREFERRED_STORE	cx-pick-up-in-store-make-my-store	SetPreferredStoreComponent	OutletPosition.REPLACE		SetPreferredStoreModule	@spartacus/pickup-in-store/components	feature-libs/pickup-in-store/components/container/set-preferred-store/set-preferred-store.module.ts:L28-L32

#	ID	NOME OUTLET RISOLTO	COMPONENT	POSITION	ALTRE CHIAVI	CONTENITORE	LIBRERIA	RIFERIMENTO
11	CartOutlets.ITEM_BUNDLE_DETAILS	cx-cart-item.bundle-details	ConfiguratorCartEntryBundleInfoComponent	OutletPosition.AFTER		ConfiguratorCartEntryBundleInfoModule	@spartacus/product-configurator/common	feature-libs/product-configurator/common/components/configurator-cart-entry-bundle-info/configurator-cart-entry-bundle-info.module.ts:L23-L27
12	CartOutlets.ITEM_DETAILS	cx-cart-item.details	ConfiguratorCartEntryInfoComponent	OutletPosition.AFTER		ConfiguratorCartEntryInfoModule	@spartacus/product-configurator/common	feature-libs/product-configurator/common/components/configurator-cart-entry-info/configurator-cart-entry-info.module.ts:L31-L35
13	CartOutlets.ITEM_CONFIGURATOR_ISSUES	cx-configurator-issues-notification	ConfiguratorIssuesNotificationComponent	OutletPosition.REPLACE		ConfiguratorIssuesNotificationModule	@spartacus/product-configurator/common	feature-libs/product-configurator/common/components/configurator-issues-notification/configurator-issues-notification.module.ts:L29-L33
14	ProductListOutlets.ITEM_ACTIONS	cx-product-list-item.actions	ConfigureProductComponent	OutletPosition.AFTER		ConfigureProductModule	@spartacus/product-configurator/common	feature-libs/product-configurator/common/components/configure-product/configure-product.module.ts:L38-L42
15	ProductListOutlets.ITEM_DETAILS	cx-product-list-item.details	ProductMultiDimensionalListItemDetailsComponent	OutletPosition.REPLACE		ProductMultiDimensionalListRootModule	@spartacus/product-multi-dimensional/list/root	feature-libs/product-multi-dimensional/list/root/product-multi-dimensional-list-root.module.ts:L37-L41
16	ProductListOutlets.ITEM_DETAILS	cx-product-list-item.details	ProductVariantStyleIconsComponent	OutletPosition.AFTER		ProductVariantsRootModule	@spartacus/product/variants/root	feature-libs/product/variants/root/product-variants-root.module.ts:L35-L39
17	CartOutlets.DELIVERY_MODE	cx-delivery-mode	DeliveryModeDatePickerComponent	OutletPosition.AFTER		RequestedDeliveryDateRootModule	@spartacus/requested-delivery-date/root	feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L37-L41
18	CartOutlets.ORDER_OVERVIEW	cx-order-overview	OrderOverviewDeliveryDateComponent	OutletPosition.AFTER		RequestedDeliveryDateRootModule	@spartacus/requested-delivery-date/root	feature-libs/requested-delivery-date/root/requested-delivery-date-root.module.ts:L42-L46
19	CartOutlets.SUBSCRIPTION_PRICE_HEADING	cx-subscription-cart-price-heading	SubscriptionCartPriceHeadingComponent	OutletPosition.AFTER		SubscriptionBillingRootModule	@spartacus/subscription-billing/root	feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L47-L51
20	CartOutlets.SUBSCRIPTION_PRICE_BODY	cx-subscription-cart-price-body	SubscriptionCartPriceBodyComponent	OutletPosition.AFTER		SubscriptionBillingRootModule	@spartacus/subscription-billing/root	feature-libs/subscription-billing/root/subscription-billing-root.module.ts:L52-L56
21	SearchBoxOutlets.RECENT_SEARCHES	SearchBoxOutlets.RECENT_SEARCHES	RecentSearchesComponent	OutletPosition.AFTER		RecentSearchesModule	@spartacus/cds	integration-libs/cds/src/recent-searches/recent-searches.module.ts:L35-L39
22	SearchBoxOutlets.RECENT_SEARCHES_HEADER	SearchBoxOutlets.RECENT_SEARCHES_HEADER	RecentSearchesHeaderComponent	OutletPosition.REPLACE		RecentSearchesModule	@spartacus/cds	integration-libs/cds/src/recent-searches/recent-searches.module.ts:L40-L44

#	ID	NOME OUTLET RISOLTO	COMPONENT	POSITION	ALTRE CHIAVI	CONTENITORE	LIBRERIA	RIFERIMENTO
23	SearchBoxOutlets.TRENDING_SEARCHES	SearchBoxOutlets.TRENDING_SEARCHES	TrendingSearchesComponent	OutletPosition.AFTER		TrendingSearchesModule	@spartacus/cds	integration-libs/cds/src/trending-searches/trending-searches.module.ts:L30-L34
24	CartOutlets.CPQ_QUOTE_MODULE	cx-cpq-quote	CpqQuoteDiscountComponent	OutletPosition.AFTER		CpqDiscountModule	@spartacus/cpq-quote/cpq-quote-discount	integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L42-L46
25	CartOutlets.CPQ_QUOTE_HEADING	cx-cpq-quote-heading	CpqQuoteHeadingComponent	OutletPosition.AFTER		CpqDiscountModule	@spartacus/cpq-quote/cpq-quote-discount	integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L47-L51
26	CartOutlets.CPQ_QUOTE	cx-cpq-quote-offer	CpqQuoteOfferComponent	OutletPosition.AFTER		CpqDiscountModule	@spartacus/cpq-quote/cpq-quote-discount	integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L52-L56
27	OpfCheckoutOutlets.OPF_CHECKOUT_BEFORE_SAVED_PAYMENT_OPTIONS	cx-opf-checkout-before-saved-payment-options	OpfGiftCardApplyComponent	OutletPosition.BEFORE		OpfGiftCardApplyModule	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/components/opf-gift-card-apply/opf-gift-card-apply.module.ts:L16-L20
28	CartOutlets.ORDER_DETAILS_AFTER_PAYMENT_METHOD	cx-order-details-payment-method-details	OpfGiftCardOrderDetailBillingComponent	(default)		OpfGiftCardOrderDetailsModule	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/components/opf-gift-card-order-details/opf-gift-card-order-details.module.ts:L31-L34
29	OpfCheckoutOutlets.ORDER_DETAILS_AFTER_PAYMENT_OPTIONS	cx-opf-after-payment-options	OpfGiftCardPaymentMethodDetailComponent	(default)		OpfGiftCardOrderDetailsModule	@spartacus/opf/gift-card/root	integration-libs/opf/gift-card/root/components/opf-gift-card-order-details/opf-gift-card-order-details.module.ts:L35-L38
30	CartOutlets.ORDER_OVERVIEW	cx-order-overview	OpfPaymentMethodDetailsComponent	OutletPosition.BEFORE		OpfPaymentMethodDetailsModule	@spartacus/opf/payment/root	integration-libs/opf/payment/root/components/opf-payment-method-details/opf-payment-method-details.module.ts:L21-L25
31	OpfCheckoutOutlets.OPF_CHECKOUT_BEFORE_PAYMENT_OPTIONS	cx-opf-checkout-before-payment-options	OpfTokenisationSavedCardsToggleComponent	OutletPosition.BEFORE		OpfTokenisationRootModule	@spartacus/opf/tokenisation/root	integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L41-L45
32	OpfCheckoutOutlets.OPF_CHECKOUT_BEFORE_PAYMENT_OPTIONS	cx-opf-checkout-before-payment-options	OpfTokenisationPaymentMethodComponent	OutletPosition.AFTER		OpfTokenisationRootModule	@spartacus/opf/tokenisation/root	integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L46-L50
33	OpfCheckoutOutlets.OPF_CHECKOUT_BEFORE_PAYMENT_OPTIONS	cx-opf-checkout-before-payment-options	OpfTokenisationNewPaymentsHeadingComponent	OutletPosition.AFTER		OpfTokenisationRootModule	@spartacus/opf/tokenisation/root	integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L51-L55
34	OrderOutlets.SERVICE_DETAILS	cx-service-details-card	ServiceDetailsCardComponent	(default)		ServiceDetailsCardModule	@spartacus/s4-service/order	integration-libs/s4-service/order/components/order-summary/service-details-card.module.ts:L24-L27
35	CartOutlets.ITEM_DETAILS	cx-cart-item.details	ScheduleLinesComponent	OutletPosition.AFTER		S4omRootModule	@spartacus/s4om/root	integration-libs/s4om/root/s4om-root.module.ts:L24-L28

Valori enum non usati in template `.html` né in `provideOutlet` (in ambito)

Possono essere usati in config TypeScript, in template inline, nelle app, o essere punti di estensione per i clienti: `PDP.INTRO`, `PDP.SHARE`, `cx-add-to-cart-container`

A09

Feature toggle

Flag di feature toggle

IN QUESTO CAPITOLO

Riepilogo

Elenco

Generata da `tools/extract/a09-feature-toggles.js` allo SHA 5b7a1c2b (`5b7a1c2bf2808d7380bc20596d10f4fa710c3da1`). Non modificare a mano.
Righe di tabella: **107**.
Rigenerazione: `cd docs/spartacus-book/tools/extract && npm install && node a09-feature-toggles.js`
Fonte: `FeatureTogglesInterface` e `defaultFeatureToggles` in `core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts`; `FeatureToggles` estende `FeatureTogglesInterface` in `core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts`. Conteggi usi = numero di occorrenze (tra parentesi: numero di file) di: **TS quoted** `'chiave'`/`"chiave"`/`'!chiave'`; **TS** `chiave:` proprietà `chiave:` o `chiave?:` (può includere omonimi non legati al toggle); **TS** `.chiave` accesso a proprietà (es. `this.featureToggles.chiave`; può includere omonimi); **HTML** `'chiave'`/`'!chiave'` nei template (es. `*cxFeature`); **SCSS** parola `chiave` nei file `.scss` (es. mixin `forFeature`), esclusi `/test/`. Ambito TS: file dell'ambito estrattore escluso il file sorgente; HTML/SCSS: file tracciati in core-libs, feature-libs, integration-libs esclusi spec e schematics.

Riepilogo

- Chiavi: **107**; default `true`: **32**; default `false`: **75**.
- Chiavi senza alcun uso rilevato: **0**.
- Chiavi presenti in `defaultFeatureToggles` ma non nell'interfaccia: nessuna.

Elenco

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
1	`useAdvancedGoogleMarkers`	When enabled, `GoogleMapRendererService` renders store locations using Google's `AdvancedMarkerElement` instead of the deprecated `Marker` class.	`false`	1 (1)	0 (0)	3 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts: L12-L20	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts: L1014-L1014
2	`useGoogleMapsAsyncLoading`	When enabled, the Google Maps JavaScript API script in `GoogleMapRendererService` is loaded with the `loading=async` URL parameter, opting into Google's async bootstrap loader.	`false`	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts: L22-L31	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts: L1015-L1015
3	`a11yKeyboardAccessibleZoom`	Adds a keyboard accessible zoom button to the `ProductImageZoomViewComponent`.	`true`	1 (1)	0 (0)	1 (1)	3 (1)	5 (2)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts: L32-L35	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts: L1017-L1017

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
4	a11yPreventCartItemsFormRedundantRecreation	When using CartItemListComponent as an outlet ([cxOutlet]="CartOutlets.CART_ITEM_LIST"): prevents the form from being recreated when neither the items nor other dependent properties (e.g., readonly) have changed.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L37-L41	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1018-L1018
5	a11yStoreFinderLabel	Adds label to 'StoreFinderSearchComponent' store search input field.	true	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L43-L46	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1019-L1019
6	a11yStoreFinderFocusOnBackButton	Moves focus to the 'Back' button when store details are shown in the store finder list, so keyboard users are not left without a focused element.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L48-L52	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1020-L1020
7	a11yB2BRegisterComponent	Enables the dedicated B2B register section on the login page, replacing the CMS-driven paragraph and link.	false	1 (1)	0 (0)	0 (0)	1 (1)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L54-L58	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1021-L1021
8	a11yLinkBtnsToTertiaryBtns	Replaces buttons resembling links with tetriary buttons in the following components: `AddToWishListComponent`, `ProductIntroComponent`, `ProductImageZoomTriggerComponent`	true	1 (1)	0 (0)	0 (0)	6 (3)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L60-L64	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1023-L1023
9	a11yAddPaddingToCarouselPanel	Adds horizontal padding to the 'carousel-panel' to fix the issue where the focus only covers three sides of the 'Previous slide' and 'Next slide' buttons within the carousel section.	true	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L66-L70	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1024-L1024

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
10	productListItemSummaryReadMore	Introduces the read more directive in product list item summary Affects: ProductListItemComponent	false	1 (1)	0 (0)	0 (0)	2 (1)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L72-L76	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1027-L1027
11	a11yFutureStockAccordionAriaControls	Fixes accessibility issue in FutureStockAccordionComponent where aria-controls references a non-existent element when accordion is collapsed.	true	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L78-L85	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1028-L1028
12	a11yNgSelectUnicodeCarets	Use unicode characters for ng-select dropdown carets so that OS themes can override the defaults by targetting text.	true	1 (1)	0 (0)	0 (0)	0 (0)	2 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L87-L91	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1025-L1025
13	a11yPreventWindowsHighContrastOverride	When enabled, prevents Windows high contrast mode from overriding the Spartacus theme.	false	1 (1)	0 (0)	1 (1)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L93-L98	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1026-L1026
14	searchBoxRecentSearchesRemoval	Updates recent-searches UX in SearchBoxComponent and CDS recent searches.	false	1 (1)	0 (0)	8 (2)	9 (2)	9 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L100-L113	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1041-L1041
15	searchBoxEmptyQueryResultsPanel	Controls the empty-query results panel in SearchBoxComponent.	false	5 (2)	0 (0)	14 (2)	2 (1)	4 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L115-L131	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1042-L1042

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
16	cdsBottomHeaderSlotAdjustPosition	Corrects BottomHeaderSlot layout when CDS registers MerchandisingCarouselComponent beside BreadcrumbComponent (e.g. on search results pages in sample data).	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L133-L151	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1043-L1043
17	authorizationCodeFlowByDefault	When enabled, sets the default oAuth configuration to use authorization code flow with PKCE.	true	3 (2)	0 (0)	23 (17)	5 (3)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L153-L161	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1029-L1029
18	authorizationCodeFlowByDefaultCsrfTokenRefresh	When enabled, refreshes the CSRF token before submitting the login form in the Authorization Code Flow.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L163-L172	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1030-L1030
19	enablePasswordExpiredErrorTranslation	When enabled, translates the "Password expired" error message to the user's selected language using Spartacus i18n.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L174-L179	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1031-L1031
20	enableMediaPrefix	When enabled, the media prefix from the backend config will be used when constructing media URLs in the MediaService.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L181-L186	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1032-L1032
21	a11yStoreFinderListItemFocus	Fixes focus ring on store name links overflowing into the address text below.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L188-L192	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1036-L1036

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
22	a11yFixSearchBoxDoubleFocus	Fixes double focus indicator on the search input field in SearchBoxComponent when navigating with the keyboard.	false	1 (1)	0 (0)	0 (0)	0 (0)	2 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L194-L203	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1037-L1037
23	a11yCustomerTicketingVisualFocusFix	Fixes keyboard focus not being visible when tabbing between some buttons on Customer Ticketing dialog.	true	1 (1)	0 (0)	0 (0)	0 (0)	2 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L205-L209	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1033-L1033
24	a11yMessagingListKeyboardFocus	Fixes keyboard focus and screen reader date announcement for the chat messaging list items.	false	1 (1)	0 (0)	1 (1)	5 (2)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L211-L218	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1034-L1034
25	orderOverviewCardsInlinePadding	Applies a 10rem inline-start padding to the order overview cards (.cx-order-details-cards) on large screens.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L220-L225	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1035-L1035
26	a11yFacetFilterByLabel	Adds Filter By label to product facets when in desktop mode.	true	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L227-L230	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1038-L1038
27	removeDuplicatedOrderHistoryHeader	When enabled, this fixes the issue of duplicated Order History headers on the Order History page.	true	0 (0)	0 (0)	0 (0)	1 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L232-L235	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1039-L1039

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
28	a11yCard NotificationMessage	When enabled, the organization's table component will stop re-rendering its rows each data update.	true	0 (0)	0 (0)	7 (5)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L237-L242	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1040-L1040
29	a11yIncreaseContastGlobalMessageCloseButton	When enabled, increases the color contrast of the close button in the global message component to meet WCAG contrast requirements.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L244-L249	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1022-L1022
30	enableB2BCostCenterSearch	When enabled, allows searching cost centers by name in the organization.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L251-L254	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1045-L1045
31	enableB2BUnitSearch	When enabled, allows searching B2B units by name in the organization administration.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L256-L260	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1044-L1044
32	enableB2BCustomerSearch	When enabled, allows searching B2B customers by name in the organization.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L262-L265	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1046-L1046
33	a11yCarouselPreventNavigationFocus	When enabled (default: true), carousel navigation buttons call preventDefault on mousedown to fix unwanted blur in Safari when the carousel is inside modals or search boxes (broken by default in Safari).	true	0 (0)	0 (0)	0 (0)	4 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L267-L275	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1047-L1047

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
34	a11yCarousel PreventIndicator Focus	In CarouselComponent, indicator buttons (the slide dots) call preventDefault() on mousedown.	false	1 (1)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L277-L291	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1048-L1048
35	a11yNgSelect ReadonlyInput Value	Sets the ng-select (readonly) input value from the selected option text, so that JAWS screen reader announces the selected value instead of "blank" when ngSelect's input element is in focus.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L293-L298	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1049-L1049
36	a11yPassword VisibilityToggle	Fixes doubled screen reader output by providing a static title and aria-label to the password visibility toggle.	true	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L300-L304	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1050-L1050
37	showOnly ActiveCurrencies	When enabled, only active currencies will be displayed in the currency selector.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L306-L311	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1051-L1051
38	a11yAddedTo CartDialog Heading	Improves accessibility of the "Added to Cart" dialog by using a semantic h2 heading instead of a div for the dialog title.	true	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L313-L319	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1052-L1052
39	a11yList SemanticsFor Facets	Ensures the facet component displays elements with proper listbox semantics.	true	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L321-L326	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1053-L1053

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
40	a11yFilteredFacetAnnouncement	In ActiveFacetsComponent, announces filter changes to screen readers.	false	0 (0)	0 (0)	0 (0)	8 (3)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L328-L333	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1054-L1054
41	a11yCartItemListHideEmptyOutlets	Hides empty outlet wrapper elements in the cart item list table when they have no content.	true	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L335-L340	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1055-L1055
42	a11yReviewsKeyboardControls	When enabled, adds arrow key navigation between reviews and uses proper list semantics so screen readers announce list position Affects: ProductReviewsComponent	true	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L342-L347	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1056-L1056
43	a11yShowMoreReviewsFocusVisible	When enabled, scrolls the 'Show More/Less Reviews' button into view after clicking it, ensuring the focused element remains visible in the viewport.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L349-L354	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1057-L1057
44	a11yCartQuickOrderFormEnableSubmitAndAddValidation	Use on existing form buttons that are programatically disabled/enabled.	true	0 (0)	0 (0)	1 (1)	3 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L356-L362	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1058-L1058
45	a11yVocalizeDropdownItemCount	When enabled, adds vocalization of dropdown item count when dropdown gains focus.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L364-L368	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1063-L1063

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
46	a11yItemCounterValueText	When enabled, the ItemCounterComponent quantity input exposes an aria-valuetext equal to its literal value.	false	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L370-L378	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1064-L1064
47	a11yRestoreFocusOnNgSelect	When enabled, keystrokes inside an ng-select (combobox dropdown) are treated as navigation rather than form filling.	false	0 (0)	0 (0)	3 (3)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L380-L387	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1065-L1065
48	a11yKeepFocusOnConsentManagementButtons	When enabled change disabled to aria-disabled on consent management buttons to keep focus when loading state toggle Affects: ConsentManagementFormComponent	false	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L389-L393	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1066-L1066
49	useEnhancedSecurePasswordValidators	When enabled, forms using CustomFormValidators.securePasswordValidators will include: CustomFormValidators.mustEndWithLegalCharacter	true	0 (0)	0 (0)	5 (5)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L395-L399	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1067-L1067
50	enableApplyVoucherEndpoint	When enabled, uses POST /carts/{cartId}/applyVoucher with the voucherId in the request body instead of POST /carts/{cartId}/vouchers?voucherId={voucherId}.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L401-L406	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1068-L1068
51	enableRemoveVoucherEndpoint	When enabled, uses POST /carts/{cartId}/removeVoucher with the voucherId in the request body instead of DELETE /carts/{cartId}/vouchers/{voucherId}.	true	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L408-L413	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1069-L1069

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
52	showSortFieldsOnlyAtTop	When enabled, shows sort fields only at the top of the table.	false	0 (0)	0 (0)	0 (0)	4 (3)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L415-L419	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1070-L1070
53	showRequiredAsterisks	When enabled, displays required field asterisks for form fields.	true	0 (0)	0 (0)	0 (0)	3 (2)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L421-L424	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1071-L1071
54	a11yConsentManagementFocusPreservation	Preserves keyboard focus on consent checkboxes after toggling.	true	0 (0)	0 (0)	3 (2)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L426-L432	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1059-L1059
55	a11yDeliveryModeFocusPreservation	Preserves keyboard focus on the selected delivery mode radio button during checkout when navigating with the keyboard.	false	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L434-L439	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1060-L1060
56	a11yFormFieldSectionLegend	When enabled, wraps form controls in a <fieldset> with aria-labelledby referencing the visible page heading, so screen readers announce the section heading when any field in the group receives focus.	false	0 (0)	0 (0)	0 (0)	7 (7)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L441-L450	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1061-L1061
57	a11yImproveCheckoutFocus	Improve auto focus during checkout process.	false	0 (0)	0 (0)	2 (1)	6 (3)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L452-L456	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1062-L1062

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
58	enableExpiredRefreshTokenHandlers	When enabled, AuthHttpHeaderService executes DI-provided ExpiredRefreshTokenHandler to take over handleExpiredRefreshToken() behavior in case of expired refresh token scenarios.	true	0 (0)	0 (0)	2 (2)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L458-L464	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1072-L1072
59	alignNavigationMenuWithHeader	When enabled, sytling is changed on navigation header and menu to be more cohesive.	false	1 (1)	0 (0)	0 (0)	0 (0)	2 (2)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L466-L469	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1016-L1016
60	enableCartReloadOnContextChange	When enabled, fixes a known issue where the cart sometimes does not reload properly on context(language or currency) change, deleting items from the cart (more specifically on the following sequence : logout - log back in - context change)	false	0 (0)	0 (0)	3 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L471-L475	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1073-L1073
61	opfPaymentVerificationCheckProcessingCartOnErrorOnly		false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L482-L482	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1074-L1074
62	a11yQuickOrderResetFocus	When enabled, the Quick Order search input keeps focus after the reset button is cleared, instead of losing focus to the next tabbable element.	false	0 (0)	0 (0)	3 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L484-L490	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1075-L1075
63	a11yCouponNotificationChannelsLinkStyling	When enabled, the "Notification Channels" link in the My Coupons page is styled as a link (blue, underlined) instead of plain text.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L492-L496	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1076-L1076

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
64	a11ySite ContextCaret Click	When enabled, fixes the caret visibility on the Language and Theme selectors by wrapping the select and caret icon in a container that displays the focus ring around both elements instead of overlapping the caret.	false	2 (2)	0 (0)	0 (0)	4 (2)	2 (2)	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L498-L504	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L1077-L1077
65	redirectOnly OnTrue NavigationEnd	When enabled, fixes a known issue where the last remembered route after logout is the route to which the logout has redirected	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L506-L509	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L1078-L1078
66	pageLink Sanitize CanonicalUrl	When enabled, sanitizes the URL used to compute the page's canonical URL (in PageLinkService.get CanonicalUrl).	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L511-L518	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L1079-L1079
67	opfUseDestroy Ref	When enabled, OPF components use DestroyRef + takeUntilDestroyed for subscription management instead of manual Subscription objects and ngOn Destroy.	false	0 (0)	0 (0)	12 (8)	0 (0)	0 (0)	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L520-L524	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L1080-L1080
68	enable Hierarchical AddressFormat	When enabled, the address book and address form support hierarchical address formats (e.g. Chinese addresses), which require selecting region (province), city and district as chained dropdowns, and skip OCC address verification for the supported countries.	false	0 (0)	0 (0)	10 (3)	6 (1)	0 (0)	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L526-L540	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L1081-L1081
69	addTitleTo AddressCard	When enabled, the title on the address is added to the line containing the full name (Card.boldText).	false	0 (0)	1 (1)	9 (7)	1 (1)	0 (0)	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L542-L545	core-libs/ core/src/ features- config/ feature- toggles/ config/ feature- toggles.ts: L1082-L1082

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
70	opfCheckoutUseUpdatePaymentTransaction		false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L552-L552	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1083-L1083
71	enableCartSlowNetworkResilience	Enables the CXSPA-10582 fixes that make the active cart resilient to slow networks: rapid multi-product add-to-cart bursts no longer lose line items, the Place Order / Proceed-to-Checkout buttons stay disabled until cart writes settle, and the mini-cart / cart page surface an "updating" indicator so users see the in-flight state.	false	3 (3)	0 (0)	6 (6)	11 (5)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L554-L583	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1084-L1084
72	a11yAddToWishListBtnMargin	When enabled, adds an 8px top margin to the "Add to Wish List" button for consistent spacing.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L585-L590	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1086-L1086
73	a11yRegistrationTermsAsteriskMargin	When enabled, adds an inline margin of 6px to the required asterisk next to the Terms & Conditions link on the registration page.	false	2 (2)	0 (0)	0 (0)	0 (0)	2 (2)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L592-L597	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1085-L1085
74	a11yProductListItemNameMargin	When enabled, applies a 6px bottom margin to product names in both product grid and product list items for consistent spacing.	false	2 (2)	0 (0)	0 (0)	0 (0)	2 (2)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L599-L604	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1087-L1087
75	propagateLogoutToAllTabs	When enabled, logging out on a tab will issue logout on all other open tabs.	false	0 (0)	0 (0)	3 (2)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L606-L610	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1088-L1088

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
76	useConfigurableLogoutRedirect	When enabled, the post-logout redirect respects the `logout.redirectRoute` option in `LogoutConfig`.	false	0 (0)	0 (0)	4 (2)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L612-L629	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1089-L1089
77	asyncAuthConfigInitializer	When enabled, adds support for asynchronous configuration of the oAuth service and adds a default initializer to adjust the oauth client details based on URL context parameters.	false	0 (0)	0 (0)	9 (3)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L631-L637	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1090-L1090
78	siteIsolationForCustomLoginPage	When enabled, adds site isolation decorator to the user credentials submitted during the Custom Login Page form submission.	false	0 (0)	0 (0)	3 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L639-L643	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1091-L1091
79	a11yNavigationSpaceKeyOnKeyUp	When enabled, the navigation menu buttons (e.g. "My Account") and dropdown headers activate on spacebar key *release* (`keyup`) rather than key *press* (`keydown`), preventing the action from firing repeatedly while the key is held down.	false	1 (1)	0 (0)	4 (2)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L645-L653	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1093-L1093
80	applyBaseSiteThemeFromCms	When enabled, the storefront's active theme follows the `theme` field of the active base site (configured in SAP Commerce BackOffice).	false	2 (2)	0 (0)	4 (2)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L655-L670	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1092-L1092
81	b2bCheckoutShippingAddressFilter	When enabled, only addresses with `shippingAddress` not explicitly set to `false` are shown on the B2B checkout delivery address step.	false	0 (0)	0 (0)	2 (2)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L672-L676	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1094-L1094

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
82	improvedTabStyling	Refines the cx-tab active/hover border: anchors it to the bottom of the button, and rounds its bottom corners.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L678-L682	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1095-L1095
83	mergeGuestCartOnCodeFlowLogin	When enabled, a guest cart is merged into the user cart after login even when the login uses the OAuth 2.1 authorization-code flow (which fully re-bootstraps the SPA and wipes in-memory state).	false	1 (1)	0 (0)	3 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L684-L698	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1099-L1099
84	reloadOnLanguageChange	When enabled, triggers a full page reload after a language switch so that all href attributes (banners, navigation links, mini-cart, login) reflect the new language immediately without requiring a manual page refresh.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L700-L710	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1096-L1096
85	productConfiguratorConsolidatedButtonDisabling	When enabled, the product configurator product card action buttons are consistently disabled while a configuration update round trip is in progress (disableActions$).	false	1 (1)	0 (0)	0 (0)	14 (2)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L712-L721	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1097-L1097
86	productConfiguratorCPQContainer	When enabled, CPQ sapContainers are converted for the storefront: CpqConfiguratorNormalizer maps CONTAINER attributes to Configurator.UiType.CONTAINER, attaches container rows and nested container-row groups, and CpqConfiguratorOverviewNormalizer adds bundle overview groups for selected container rows.	false	25 (11)	0 (0)	16 (14)	14 (5)	12 (7)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L723-L788	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1098-L1098
87	a11yFormErrorIconContrast	When enabled, the form-error icon glyph (cx-form-errors) uses the --cx-color-danger-accent color instead of --cx-color-inverse, so it stays legible on the danger background in the high-contrast dark theme (black glyph instead of white).	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L790-L798	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1100-L1100

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
88	a11yFocus Indicator Contrast	When enabled, the default theme's keyboard focus indicator color (`--cx-color-visual-focus`) is darkened so the focus outline of all UI elements (input fields, search box, comboboxes, checkboxes, radio buttons, etc.) meets the WCAG 1.4.11 non-text contrast requirement of >=3:1 against the adjacent surface.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L800-L807	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1101-L1101
89	a11yPrimary ColorContrast	When enabled, the default theme's primary color (`--cx-color-primary`) is darkened so the contrast meets the contrast requirement of >= 4.5:1.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L809-L815	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1102-L1102
90	a11yDisabled Button Contrast	When enabled, disabled action buttons (`.btn-primary`, `.btn-secondary`, `.btn-tertiary`) use the new `--cx-color-disabled` token instead of `--cx-color-border-focus`, so their border/background/text meet the WCAG 1.4.11 3:1 (7:1 in high-contrast themes) non-text contrast requirement.	false	1 (1)	0 (0)	0 (0)	0 (0)	3 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L817-L824	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1103-L1103
91	a11yImprove AddressForm Focus	When enabled, the address form applies the `cxFocus` directive with autofocus to manage initial keyboard focus and uses new directive cxFocusFirstInvalidField.	false	0 (0)	0 (0)	2 (1)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L826-L832	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1104-L1104
92	a11yFocus BreadcrumbOn Navigation	When enabled, after navigating to a `CategoryPage` (e.g. a Product Listing Page) via a header navigation link, keyboard focus moves to the first anchor inside `cx-breadcrumb` (resolved via `StorefrontComponent.categoryPageFocusSelector`) rather than the first focusable element in `<main>`.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L834-L844	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1105-L1105
93	cart Validation Display Backend Messages	When enabled (with the `cart.validation.enabled` config), surfaces backend min/max order quantity validation in the cart: highlights violating rows and shows a per-item `Min qty` / `Max qty` hint parsed from the backend `statusMessage`, re-validating on entry and on quantity change.	false	5 (3)	0 (0)	4 (4)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L846-L853	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1106-L1106

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
94	configuratorIssuesNotificationForConfigurableOnly	When enabled, ConfiguratorIssuesNotificationComponent only shows its cart "issues" notification for configurable products, so non-configurable entries (e.g. with min/max quantity validation errors) no longer trigger it.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L855-L862	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1107-L1107
95	globalMessageCloseButtonPadding	When enabled, GlobalMessageComponent reserves space for the close button so that long, multi-line message text does not render underneath it.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L864-L870	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1108-L1108
96	a11yNavigationChevronContrast	When enabled, the navigation chevron (--list-bg) rendered on the organization (My Company) list rows and detail navigation cards uses a higher-contrast stroke so it meets the WCAG 1.4.11 non-text contrast requirement (>=3:1 in the default theme; 7:1 in high-contrast themes).	false	1 (1)	0 (0)	0 (0)	0 (0)	7 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L872-L883	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1109-L1109
97	showWarningMessageOnRequoteButtonClick	When enabled, the requote button when clicked in the cancelled quote details page will show a warning message to the user that the quote-cart should have a minimum threshold value of items to be able to proceed with the requote process.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L885-L892	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1110-L1110
98	oauthCallbackPage	Enables support for a dedicated oAuth callback page to be used for the Return URI in Authorization Code Flow.	false	2 (1)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L894-L914	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1111-L1111
99	enableFormFieldMaxLength	When enabled, Validators.maxLength is applied to all text form fields in address and registration forms, providing visible error feedback instead of silently blocking input at the HTML level.	false	0 (0)	0 (0)	5 (4)	15 (3)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L916-L924	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1112-L1112

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
100	a11yCvvInfoIconKeyboardAccessible	Makes the CVV (Card Verification Value) info icon in the checkout payment form keyboard accessible by rendering it as a focusable button.	false	1 (1)	0 (0)	1 (1)	2 (1)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L926-L932	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1113-L1113
101	a11yBannerArrowKeyNavigation	When enabled, adds arrow key navigation to organization administration navigation tiles/cards on the My Company landing page (CompanyPageTemplate), turning the banner tile group into a composite keyboard widget.	false	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L934-L941	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1114-L1114
102	a11yTableKeyboardNavigation	When enabled, adds up/down arrow key navigation between table rows, turning the table into a composite keyboard widget.	false	1 (1)	0 (0)	0 (0)	2 (1)	2 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L943-L949	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1115-L1115
103	a11yPaginationKeyboardNavigation	When enabled, replaces per-link tabindex binding in PaginationComponent with roving tabindex (left/right arrow key navigation across page links).	false	0 (0)	0 (0)	1 (1)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L951-L956	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1116-L1116
104	a11yCarouselItemArrowKeyNavigation	When enabled, adds `cxFocusableCarouselItem` to product carousel items so arrow key navigation works inside the carousel.	false	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L958-L964	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1117-L1117
105	a11yCouponDialogResetButtonKeyboardAccessible	When enabled, the RESET button in the "Add To Your Coupon List" claim dialog is rendered as a proper `<button>` element instead of an `<a role="button">` without an `href`, making it reachable and operable with the keyboard.	false	0 (0)	0 (0)	0 (0)	2 (1)	0 (0)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L966-L973	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1118-L1118

#	CHIAVE	DESCRIZIONE (PRIMA FRASE JSDOC)	DEFAULT	TS QUOTED	TS CHIAVE:	TS . CHIAVE	HTML	SCSS	RIFERIMENTO (INTERFACCIA)	RIFERIMENTO (DEFAULT)
106	a11yInStockInfoTextContrast	When enabled, the "In Stock" / "Out of Stock" info text in the AddToCartComponent uses --cx-color-text instead of --cx-color-secondary, ensuring the text meets the WCAG 1.4.3 Level AA minimum contrast ratio of 4.5:1 against all background surfaces.	false	1 (1)	0 (0)	0 (0)	0 (0)	1 (1)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L975-L983	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1119-L1119
107	a11yHighContrastFix	When enabled, improves contrast in high-contrast themes across multiple components that were missing HC overrides or had insufficient contrast ratios (below 7:1 AAA): - StoreComponent / StoreSearchComponent (pickup-in-store): "View Hours" button, "Use My Location" link, and stock/opening-hours icons (CXSPA-12810) - CardComponent: delete-address confirmation message text (CXSPA-12821) - CustomerTicketingListComponent: table column header <th> cells (CXSPA-12801) - QuickOrderComponent: ng-select dropdown panel items (CXSPA-12775) - ProductReviewsComponent: "Read More" / "Read Less" links (CXSPA-12804) - StoreFinderListItemComponent: store links including hover state (CXSPA-12786) - SearchBoxComponent: search-box-v2 variant results panel (CXSPA-12789) - SiteContextSelectorComponent: language/currency/theme <select> background (CXSPA-12783) Affected SCSS files: - feature-libs/pickup-in-store/styles/_store.scss - feature-libs/pickup-in-store/styles/_store-search.scss - core-libs/styles/scss/components/misc/card/_card.scss - feature-libs/customer-ticketing/styles/components/_customer-ticketing-list.scss - feature-libs/cart/quick-order/styles/_quick-order.scss - core-libs/styles/scss/components/product/details/_product-reviews.scss - feature-libs/storefinder/styles/components/_store-finder-list-item.scss - core-libs/styles/scss/components/product/search/_searchbox.scss - core-libs/styles/scss/components/layout/header/_site-context-selector.scss	false	13 (13)	0 (0)	0 (0)	0 (0)	14 (13)	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L985-L1010	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L1120-L1120

A10

API deprecate (@deprecated)

Simboli marcati `@deprecated`

IN QUESTO CAPITOLO

Conteggi per libreria

Conteggi per tipo di simbolo

Elenco

Generata da `tools/extract/a10-deprecated.js` allo SHA 5b7a1c2b (`5b7a1c2bf2808d7380bc20596d10f4fa710c3da1`). Non modificare a mano.
Righe di tabella: **165**.
Rigenerazione: `cd docs/spartacus-book/tools/extract && npm install && node a10-deprecated.js`
Ogni riga è un'occorrenza di `@deprecated` in un commento (JSDoc o `//`) dei file dell'ambito (esclusi spec, testing, mock, schematics). *Simbolo*: dichiarazione a cui il commento è agganciato (leading trivia, risolto con ts-morph), qualificata con il contenitore. *Testo*: resto della riga del tag; se vuoto, riga di commento successiva. *Riferimento*: dalla riga del tag alla riga d'inizio del simbolo.

Conteggi per libreria

LIBRERIA	OCCORRENZE
@spartacus/cart	2
@spartacus/cdc	1
@spartacus/checkout	2
@spartacus/core	14
@spartacus/epd-visualization	91
@spartacus/opf	8
@spartacus/order	2
@spartacus/organization	7
@spartacus/product-configurator	12
@spartacus/quote	1
@spartacus/s4-service	2
@spartacus/setup	1
@spartacus/storefinder	6
@spartacus/storefront	13
@spartacus/subscription-billing	3
Totale	**165**

Conteggi per tipo di simbolo

KIND (TS)	OCCORRENZE
ClassDeclaration	40
variable	29
MethodDeclaration	20
PropertyDeclaration	20
InterfaceDeclaration	20
PropertySignature	13

KIND (TS)	OCCORRENZE
EnumDeclaration	9
FunctionDeclaration	7
parameter	3
SetAccessor	2
IndexSignature	1
ModuleDeclaration	1

Occorrenze senza testo dopo il tag: 1. Occorrenze di `@deprecated` dentro stringhe/template (escluse): 0.

Elenco

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
1	`CartItemListComponent.setLoading`	SetAccessor	Use `cartIsLoading`	@spartacus/cart	`feature-libs/cart/base/components/cart-shared/cart-item-list/cart-item-list.component.ts:L104-L106`
2	`MergeCart`	ClassDeclaration	since 221121.1	@spartacus/cart	`feature-libs/cart/base/core/store/actions/cart.action.ts:L182-L186`
3	`CdcLogoutGuard.constructor(cms)`	parameter	since 2611 - only used in the legacy code path when {@link FeatureToggles.useConfigurableLogoutRedirect} is disabled.	@spartacus/cdc	`integration-libs/cdc/root/guards/cdc-logout.guard.ts:L29-L31`
4	`CheckoutDeliveryAddressComponent.getCardRole`	MethodDeclaration	since 221121.8. No longer used internally.	@spartacus/checkout	`feature-libs/checkout/base/components/checkout-delivery-address/checkout-delivery-address.component.ts:L389-L393`
5	`CheckoutPaymentFormComponent.useExtractedBillingAddressComponent`	PropertyDeclaration	This property is obsolete since 2211.42	@spartacus/checkout	`feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-form/checkout-payment-form.component.ts:L173-L175`
6	`USE_AUTHORIZATION_CODE_FLOW_BY_DEFAULT`	variable	since 221121.1	@spartacus/core	`core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L12-L15`
7	`PageMetaService.constructor(platformId)`	parameter	(will be removed in 12 months)	@spartacus/core	`core-libs/core/src/cms/facade/page-meta.service.ts:L39-L40`
8	`locationInitializedFactory`	FunctionDeclaration	since 2211.22 - should not be a public API	@spartacus/core	`core-libs/core/src/config/config-initializer/config-initializer.module.ts:L33-L35`
9	`FeaturesConfig`	ClassDeclaration	Please use a different util to provide values of feature toggles:	@spartacus/core	`core-libs/core/src/features-config/config/features-config.ts:L11-L27`
10	`FeaturesConfig.features`	PropertyDeclaration	Please use a different util to provide values of feature toggles:	@spartacus/core	`core-libs/core/src/features-config/config/features-config.ts:L33-L48`
11	`FeaturesConfig.level`	PropertySignature	Please use named feature flags instead of feature level.	@spartacus/core	`core-libs/core/src/features-config/config/features-config.ts:L54-L70`

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
12	`FeaturesConfig.[_: string]`	IndexSignature	Please use a different util to provide values of feature toggles:	@spartacus/core	`core-libs/core/src/features-config/config/features-config.ts:L73-L88`
13	`FeaturesConfig.disableConfigUpdates`	PropertySignature	- this feature will be moved into INJECTION_TOKEN in https://jira.tools.sap/browse/CXSPA-13592	@spartacus/core	`core-libs/core/src/features-config/config/features-config.ts:L102-L116`
14	`FeatureLevelDirective`	ClassDeclaration	Unused directive. Please use named feature flags instead of feature level.	@spartacus/core	`core-libs/core/src/features-config/directives/feature-level.directive.ts:L11-L13`
15	`FeatureConfigService`	ClassDeclaration	- Use FeatureToggles instead	@spartacus/core	`core-libs/core/src/features-config/services/feature-config.service.ts:L16-L18`
16	`FeatureConfigService.isLevel`	MethodDeclaration	- features level is no longer used	@spartacus/core	`core-libs/core/src/features-config/services/feature-config.service.ts:L25-L27`
17	`isInLevel`	FunctionDeclaration	- features level is no longer used	@spartacus/core	`core-libs/core/src/features-config/utils/feature-config-utils.ts:L16-L18`
18	`isFeatureLevel`	FunctionDeclaration	- features level is no longer used	@spartacus/core	`core-libs/core/src/features-config/utils/feature-config-utils.ts:L37-L39`
19	`SiteContextRoutesHandler.init`	MethodDeclaration	since 2211.32. Use `initOnce()` instead for safe idempotent initialization.	@spartacus/core	`core-libs/core/src/site-context/services/site-context-routes-handler.ts:L51-L54`
20	`cs`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/cs/index.ts:L10-L12`
21	`de`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/de/index.ts:L10-L12`
22	`en`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/en/index.ts:L10-L12`
23	`es_CO`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/es_CO/index.ts:L10-L12`
24	`es`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/es/index.ts:L10-L12`
25	`fr`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/fr/index.ts:L10-L12`
26	`hi`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/hi/index.ts:L10-L12`
27	`hu`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/hu/index.ts:L10-L12`
28	`id`	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	`integration-libs/epd-visualization/assets/translations/id/index.ts:L10-L12`

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
29	it	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/it/index.ts:L10-L12
30	ja	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/ja/index.ts:L10-L12
31	ko	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/ko/index.ts:L10-L12
32	pl	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/pl/index.ts:L10-L12
33	pt	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/pt/index.ts:L10-L12
34	ru	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/ru/index.ts:L10-L12
35	epdVisualizationTranslationChunksConfig	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/translations.ts:L10-L12
36	zh_TW	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/zh_TW/index.ts:L10-L12
37	zh	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/assets/translations/zh/index.ts:L10-L12
38	EpdVisualizationComponentsModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/epd-visualization-components.module.ts:L12-L14
39	VisualPickingProductFilterComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-filter/visual-picking-product-filter.component.ts:L14-L16
40	VisualPickingProductFilterModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-filter/visual-picking-product-filter.module.ts:L16-L18
41	VisualPickingProductFilterService	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-filter/visual-picking-product-filter.service.ts:L13-L15
42	CompactAddToCartComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/compact-add-to-cart/compact-add-to-cart.component.ts:L15-L17
43	CompactAddToCartModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/compact-add-to-cart/compact-add-to-cart.module.ts:L21-L23

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
44	VisualPickingProductListItem	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/model/visual-picking-product-list-item.model.ts:L12-L14
45	PagedListComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/paged-list/paged-list.component.ts:L31-L33
46	PagedListModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/paged-list/paged-list.module.ts:L15-L17
47	VisualPickingProductListComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/visual-picking-product-list.component.ts:L25-L27
48	VisualPickingProductListModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/visual-picking-product-list.module.ts:L22-L24
49	VisualPickingProductListService	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/visual-picking-product-list.service.ts:L32-L34
50	VisualPickingTabComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.component.ts:L22-L24
51	VisualPickingTabModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.module.ts:L17-L19
52	VisualPickingTabService	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.service.ts:L28-L30
53	NavigationMode	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/navigation-mode.ts:L8-L10
54	NodeContentType	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/node-content-type.ts:L8-L10
55	LoadedSceneInfo	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/scene-load-info.ts:L11-L13
56	SceneLoadInfo	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/scene-load-info.ts:L19-L21
57	SceneLoadState	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/scene-load-state.ts:L8-L10
58	SelectionDisplayMode	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/selection-display-mode.ts:L10-L12

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
59	SelectionMode	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/selection-mode.ts:L10-L12
60	VisualizationLookupResult	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/visualization-load-info.ts:L10-L12
61	VisualizationLoadStatus	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/visualization-load-info.ts:L20-L22
62	VisualizationLoadInfo	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/visualization-load-info.ts:L32-L34
63	ZoomTo	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/models/zoom-to.ts:L8-L10
64	VisualViewerAnimationSliderComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/toolbar/visual-viewer-animation-slider/visual-viewer-animation-slider.component.ts:L21-L23
65	VisualViewerAnimationSliderModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/toolbar/visual-viewer-animation-slider/visual-viewer-animation-slider.module.ts:L13-L15
66	VisualViewerAnimationSliderService	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/toolbar/visual-viewer-animation-slider/visual-viewer-animation-slider.service.ts:L18-L20
67	VisualViewerToolbarButtonComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/toolbar/visual-viewer-toolbar-button/visual-viewer-toolbar-button.component.ts:L11-L13
68	VisualViewerToolbarButtonModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/toolbar/visual-viewer-toolbar-button/visual-viewer-toolbar-button.module.ts:L13-L15
69	VisualViewerComponent	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/visual-viewer.component.ts:L25-L27
70	VisualViewerModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/visual-viewer.module.ts:L17-L19
71	VisualViewerService	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L72-L74
72	NODES_RESPONSE_NORMALIZER	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/scene/converters.ts:L12-L14
73	NodesResponse	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/scene/nodes-response.ts:L8-L10
74	TreeNode	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/scene/nodes-response.ts:L15-L17

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
75	UsageId	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/scene/nodes-response.ts:L23-L25
76	UsageIdKey	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/scene/nodes-response.ts:L31-L33
77	SceneAdapter	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/scene/scene.adapter.ts:L11-L13
78	SceneConnector	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/scene/scene.connector.ts:L13-L15
79	LOOKUP_VISUALIZATIONS_RESPONSE_NORMALIZER	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/visualization/converters.ts:L12-L14
80	LookupVisualizationsResponse	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/visualization/lookup-visualizations-response.ts:L10-L12
81	VisualizationAdapter	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/visualization/visualization.adapter.ts:L12-L14
82	VisualizationConnector	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/connectors/visualization/visualization.connector.ts:L14-L16
83	EpdVisualizationCoreModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/epd-visualization-core.module.ts:L13-L15
84	NodeIdProductCodes	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L21-L23
85	SceneNodeToProductLookupService	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L29-L31
86	VisualizationLookupService	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/core/services/visualization-lookup/visualization-lookup.service.ts:L22-L24
87	StorageV1Adapter	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/epd-visualization-api/adapters/storage-v1/storage-v1.adapter.ts:L33-L35
88	VisualizationV1Adapter	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/epd-visualization-api/adapters/visualization-v1/visualization-v1.adapter.ts:L34-L36
89	EpdVisualizationApiModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/epd-visualization-api/epd-visualization-api.module.ts:L16-L18
90	EpdVisualizationModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/epd-visualization.module.ts:L13-L15
91	epdVisualizationConfigValidator	FunctionDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config-validator.ts:L14-L16

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
92	EpdVisualizationConfig	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L13-L15
93	EpdVisualizationInnerConfig	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L27-L29
94	Ui5Config	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L49-L51
95	VisualizationApiConfig	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L63-L65
96	UsageIdConfig	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L75-L77
97	VisualPickingConfig	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L97-L99
98	'@spartacus/core'	ModuleDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L107-L109
99	getEpdVisualizationDefaultConfig	FunctionDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/config/epd-visualization-default-config.ts:L10-L12
100	EpdVisualizationRootModule	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/epd-visualization-root.module.ts:L29-L31
101	EPD_VISUALIZATION_FEATURE	variable	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/feature-name.ts:L8-L10
102	UsageIdDefinition	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/models/usage-ids/usage-id-definition.ts:L8-L10
103	UsageId	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/models/usage-ids/usage-id.ts:L8-L10
104	UsageIdKey	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/models/usage-ids/usage-id.ts:L16-L18
105	ContentType	EnumDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/models/visualizations/content-type.ts:L12-L14
106	VisualizationInfo	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/models/visualizations/visualization-info.ts:L12-L14
107	Listener	InterfaceDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/util/event-listener-utils.ts:L10-L12
108	EventListenerUtils	ClassDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/util/event-listener-utils.ts:L19-L21

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
109	getUrl	FunctionDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/util/url-utils.ts:L8-L10
110	isHttpOrHttps	FunctionDeclaration	since v221121.5.0 - The epd-visualization integration library will be removed in the future.	@spartacus/epd-visualization	integration-libs/epd-visualization/root/util/url-utils.ts:L19-L21
111	OpfCheckoutPaymentAndReviewComponent.opfBaseFacade	PropertyDeclaration	since 221121.21. No longer used internally.	@spartacus/opf	integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L83-L86
112	OpfSharedPaymentSubmitOptions.cartId	PropertySignature	Property no longer used. Cart ID is currently derived from the OTP access code header.	@spartacus/opf	integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L46-L47
113	OpfSharedPaymentSubmitCompleteOptions.cartId	PropertySignature	Property no longer used. Cart ID is currently derived from the OTP access code header.	@spartacus/opf	integration-libs/opf/global-functions/core/services/opf-global-functions-shared.service.ts:L59-L60
114	OpfPaymentGlobalMethods.cartId	PropertySignature	Property no longer used. Cart ID is currently derived from the OTP access code header.	@spartacus/opf	integration-libs/opf/payment/root/model/opf-payment.model.ts:L26-L27
115	OpfPaymentGlobalMethods.cartId	PropertySignature	Property no longer used. Cart ID is currently derived from the OTP access code header.	@spartacus/opf	integration-libs/opf/payment/root/model/opf-payment.model.ts:L37-L38
116	OpfPaymentGlobalMethods.cartId	PropertySignature	Property no longer used. Cart ID is currently derived from the OTP access code header.	@spartacus/opf	integration-libs/opf/payment/root/model/opf-payment.model.ts:L47-L48
117	OpfPaymentConfig.cartId	PropertySignature	Property no longer used. Cart ID is currently derived from the OTP access code header.	@spartacus/opf	integration-libs/opf/payment/root/model/opf-payment.model.ts:L217-L218
118	ApplePaySessionVerificationRequest.cartId	PropertySignature	Property no longer used. Cart ID is currently derived from the OTP access code header.	@spartacus/opf	integration-libs/opf/quick-buy/root/model/opf-apple-pay.model.ts:L12-L13
119	OrderHistoryService.getQueryParams	MethodDeclaration	: Method doesn't pass facade method's requirements (returns void or Observable)	@spartacus/order	feature-libs/order/core/facade/order-history.service.ts:L221-L223
120	OrderHistoryFacade.getQueryParams	MethodDeclaration	: Method doesn't pass facade method's requirements (returns void or Observable)	@spartacus/order	feature-libs/order/root/facade/order-history.facade.ts:L145-L148
121	newBudgetTableConfig	variable	Use budgetTableConfig instead.	@spartacus/organization	feature-libs/organization/administration/components/budget/budget.config.ts:L130-L132
122	newCostCenterTableConfig	variable	Use costCenterTableConfig instead.	@spartacus/organization	feature-libs/organization/administration/components/cost-center/cost-center.config.ts:L157-L159
123	newPermissionTableConfig	variable	Use permisionTableConfig instead.	@spartacus/organization	feature-libs/organization/administration/components/permission/permission.config.ts:L105-L107
124	ToggleLinkCellComponent.restoreFocus	MethodDeclaration		@spartacus/organization	feature-libs/organization/administration/components/unit/list/toggle-link/toggle-link-cell.component.ts:L190-L192
125	newUnitsTableConfig	variable	Use unitsTableConfig instead.	@spartacus/organization	feature-libs/organization/administration/components/unit/units.config.ts:L327-L329

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
126	`newUserGroupTableConfig`	variable	Use `userGroupTableConfig` instead.	@spartacus/organization	`feature-libs/organization/administration/components/user-group/user-group.config.ts:L201-L203`
127	`newUserTableConfig`	variable	Use `userTableConfig` instead.	@spartacus/organization	`feature-libs/organization/administration/components/user/user.config.ts:L254-L256`
128	`ConfiguratorAttributeProductCardComponentOptions.attributeId`	PropertySignature	since 221121.23 - Use `this.getAttributeCode(this.attribute)` instead which will be	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/attribute/product-card/configurator-attribute-product-card.component.ts:L79-L83`
129	`ConfiguratorAttributeProductCardComponentOptions.attributeLabel`	PropertySignature	since 221121.23 - Use `attribute.label` instead which will be	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/attribute/product-card/configurator-attribute-product-card.component.ts:L85-L89`
130	`ConfiguratorAttributeProductCardComponentOptions.attributeName`	PropertySignature	since 221121.23 - Use `attribute.name` instead which will be	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/attribute/product-card/configurator-attribute-product-card.component.ts:L91-L95`
131	`ConfiguratorConflictAndErrorMessagesComponent.configuration$`	PropertyDeclaration	since 221121.23 - Use `messages$` instead, which only exposes	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/configurator-conflict-and-error-messages/configurator-conflict-and-error-messages.component.ts:L36-L41`
132	`ConfiguratorGroupTitleComponent.ghostStyle`	PropertyDeclaration	since 221121.23 - Prefer the `#ghostGroup` template	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/group-title/configurator-group-title.component.ts:L46-L53`
133	`ConfiguratorOverviewFilterButtonComponent.ghostStyle`	PropertyDeclaration	since 221121.23 - Prefer the `#ghostFilterButton` template	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/overview-filter-button/configurator-overview-filter-button.component.ts:L50-L57`
134	`ConfiguratorOverviewFormComponent.ghostStyle`	PropertyDeclaration	since 221121.23 - Prefer the `#ghostForm` template	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/overview-form/configurator-overview-form.component.ts:L79-L86`
135	`ConfiguratorOverviewMenuComponent.VARIANT_CONFIG_OVERVIEW_NAVIGATION_SLOT`	PropertyDeclaration	since 221121.23 - The overview menu resolves its host page slot	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/overview-menu/configurator-overview-menu.component.ts:L46-L50`
136	`ConfiguratorOverviewSidebarComponent.ghostStyle`	PropertyDeclaration	since 221121.23 - Prefer the `#ghostSidebar` template	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/overview-sidebar/configurator-overview-sidebar.component.ts:L47-L54`
137	`ConfiguratorProductTitleComponent.ghostStyle`	PropertyDeclaration	since 221121.23 - Prefer the `#ghostProductTitle` template	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/product-title/configurator-product-title.component.ts:L73-L80`
138	`ConfiguratorProductTitleComponent.product$`	PropertyDeclaration	since 221121.23 - Use `productTitleData$` and read	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/product-title/configurator-product-title.component.ts:L119-L123`
139	`ConfiguratorTabBarComponent.ghostStyle`	PropertyDeclaration	since 221121.23 - Prefer the `#ghostTabBar` template	@spartacus/product-configurator	`feature-libs/product-configurator/rulebased/components/tab-bar/configurator-tab-bar.component.ts:L43-L50`
140	`QuoteLinksComponent.isShowDownloadProposalButtonFeatureEnabled`	MethodDeclaration	This method is obsolete since 2211.41	@spartacus/quote	`feature-libs/quote/components/links/quote-links.component.ts:L99-L101`

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
141	S4ServiceOrderDetailActionsComponent.displayActions$	PropertyDeclaration	since 221121.1 - Displaying the individual action buttons will depend on their boolean flags from API response.	@spartacus/s4-service	integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.component.ts:L50-L52
142	S4ServiceOrderDetailActionsComponent.checkServiceStatus	MethodDeclaration	since 221121.1 - Displaying the individual action buttons will depend on their boolean flags from API response.	@spartacus/s4-service	integration-libs/s4-service/order/components/s4-service-order-detail-actions/s4-service-order-detail-actions.component.ts:L61-L64
143	SsrOptimizationOptions.debug	PropertySignature	- This flag is not used anymore since v2211.27.	@spartacus/setup	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L135-L143
144	StoreFinderService.getStoreLatitude	MethodDeclaration	Please use StoreLocationService instead,	@spartacus/storefinder	feature-libs/storefinder/core/facade/store-finder.service.ts:L231-L235
145	StoreFinderService.getStoreLongitude	MethodDeclaration	Please use StoreLocationService instead,	@spartacus/storefinder	feature-libs/storefinder/core/facade/store-finder.service.ts:L240-L244
146	StoreFinderService.getDirections	MethodDeclaration	Please use StoreLocationService instead,	@spartacus/storefinder	feature-libs/storefinder/core/facade/store-finder.service.ts:L249-L254
147	StoreFinderFacade.getStoreLatitude	MethodDeclaration	Please use StoreLocationService instead,	@spartacus/storefinder	feature-libs/storefinder/root/facade/store-finder.facade.ts:L61-L63
148	StoreFinderFacade.getStoreLongitude	MethodDeclaration	Please use StoreLocationService instead,	@spartacus/storefinder	feature-libs/storefinder/root/facade/store-finder.facade.ts:L65-L67
149	StoreFinderFacade.getDirections	MethodDeclaration	Please use StoreLocationService instead,	@spartacus/storefinder	feature-libs/storefinder/root/facade/store-finder.facade.ts:L69-L71
150	TabParagraphContainerComponent.activeTabNum	PropertyDeclaration	This method will be removed.	@spartacus/storefront	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L60-L62
151	TabParagraphContainerComponent.ariaLabel	PropertyDeclaration	This method will be removed.	@spartacus/storefront	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L64-L66
152	TabParagraphContainerComponent.children	PropertyDeclaration	This method will be removed.	@spartacus/storefront	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L69-L71
153	TabParagraphContainerComponent.tabTitleParams	PropertyDeclaration	This method will be removed.	@spartacus/storefront	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L78-L80
154	TabParagraphContainerComponent.select	MethodDeclaration	This method will be removed.	@spartacus/storefront	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L145-L147
155	TabParagraphContainerComponent.ngOnInit	MethodDeclaration	This method will be removed.	@spartacus/storefront	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L161-L163
156	SiteContextSelectorComponent.siteContextService	PropertyDeclaration	since 2011.21 removed unused property	@spartacus/storefront	core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.component.ts:L44-L46

#	SIMBOLO	KIND	TESTO DEL TAG (PRIMA RIGA)	LIBRERIA	RIFERIMENTO
157	`ProductIntroComponent.areReviewsAvailable$`	PropertyDeclaration	This observable controlled whether we should show reviews based	@spartacus/storefront	`core-libs/storefront/cms-components/product/product-intro/product-intro.component.ts:L46-L50`
158	`ProductScrollComponent.setConfig`	SetAccessor	Use `scrollConfig` instead	@spartacus/storefront	`core-libs/storefront/cms-components/product/product-list/container/product-scroll/product-scroll.component.ts:L40-L42`
159	`FacetComponent.toggleGroup`	MethodDeclaration	: Header will no longer be used in favour of TabComponent headers.	@spartacus/storefront	`core-libs/storefront/cms-components/product/product-list/product-facet-navigation/facet/facet.component.ts:L93-L99`
160	`LogoutGuard.constructor(cms)`	parameter	since 221121.21 - only used in the legacy code path when {@link FeatureToggles.useConfigurableLogoutRedirect} is disabled.	@spartacus/storefront	`core-libs/storefront/cms-components/user/logout/logout.guard.ts:L37-L39`
161	`OutletContextData.context`	PropertyDeclaration	since 3.0 - use `context$` observable instead	@spartacus/storefront	`core-libs/storefront/cms-structure/outlet/outlet.model.ts:L37-L39`
162	`layoutConfig`	variable	Use `layoutConfigFactory` and `provideConfigFactory(layoutConfigFactory)` instead.	@spartacus/storefront	`core-libs/storefront/recipes/config/layout-config.ts:L25-L27`
163	`SubscriptionBillingListComponent.onFilterDateChange`	MethodDeclaration	- Selection of date range and relevant methods have been moved to DateRangeModalComponent	@spartacus/subscription-billing	`feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L118-L120`
164	`SubscriptionBillingListComponent.onResetFilterDate`	MethodDeclaration	- Selection of date range and relevant methods have been moved to DateRangeModalComponent	@spartacus/subscription-billing	`feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L139-L141`
165	`SubscriptionBillingListComponent.onDateFilterSubmit`	MethodDeclaration	- Selection of date range and relevant methods have been moved to DateRangeModalComponent	@spartacus/subscription-billing	`feature-libs/subscription-billing/components/list/billing/subscription-billing-list.component.ts:L156-L158`

A11

Glossario

Termini del libro

IN QUESTO CAPITOLO

Conteggi
Come rigenerare
Indice
A
B
C
D
E
F
G
H
I
L
M

Generata da `tools/extract3/a11_glossario.py` allo SHA 5b7a1c2b (`5b7a1c2bf2808d7380bc20596d10f4fa710c3da1`). Non modificare a mano: rigenerare.

Glossario dei termini chiave del libro. Ogni definizione riassume ciò che afferma il **capitolo principale** linkato (e, se presente, il capitolo linkato dentro il testo); i riferimenti `path:Lx-Ly` sono copiati da quel capitolo e lo script verifica che vi compaiano letteralmente. La riga *Citato in* riporta i capitoli in cui il termine compare (conteggio delle occorrenze nel testo fuori dai blocchi di codice, con la regex della voce in `tools/extract3/a11_termini.py`).

Le definizioni sono curate a mano in `tools/extract3/a11_termini.py`; lo script estrae i candidati, conta le citazioni e verifica i riferimenti. Per i simboli TypeScript vedi anche A12 — Indice dei simboli.

Conteggi

- Voci definite: **58**
- di cui dall'elenco base: **45**
- Candidati estratti (grassetto o backtick in almeno 3 capitoli): **514**
- Candidati coperti da una voce definita: **91**
- Candidati non definiti elencati in fondo (almeno 6 capitoli): **89**
- Capitoli scansionati: **51**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract3/a11_glossario.py
# oppure entrambe le appendici: bash tools/extract3/run_all.sh
```

Indice

- **A**: Adapter · APP_INITIALIZER
- **B**: Base site · BaseCoreModule / BaseStorefrontModule · Breakpoint
- **C**: CMS · cmsComponents · CmsPageGuard · Command / Query · CONFIG_INITIALIZER · ConfigChunk / DefaultConfigChunk · Connector · Converter / normalizer / serializer
- **D**: deepMerge
- **E**: EntityLoaderState · Entry point · Environment injector · EventService
- **F**: Facade · facadeFactory · Feature module · Feature toggle · featureModules
- **G**: getStable
- **H**: HTTP_INTERCEPTORS · HttpErrorHandler · Hydration
- **I**: Injection token
- **L**: LAUNCH_CALLER · LayoutConfig · LoaderState · LoggerService

- **M**: Migrazione (ng update) · MODULE_INITIALIZER · Multi provider
- **N**: NodeInjector
- **O**: OCC · OccEndpoints · OptimizedSsrEngine · Outlet
- **P**: Page layout · PageContext · PageMetaResolver · PendingTasks · Process state · provideConfig / provideDefaultConfig · public_api
- **R**: Root module (feature lib) · Route semantica
- **S**: Schematics · SERVER_REQUEST_URL / WindowRef · Site context · Slot · SSR (Server-Side Rendering)
- **T**: Tipi augmentabili · TransferState
- **U**: UnifiedInjector
- **Z**: Zoneless

A

Adapter

Classe astratta da cui dipende il dominio (`ProductAdapter`, `CmsPageAdapter`, `SiteAdapter`, …); il modulo OCC la lega via DI all'implementazione concreta (`OccProductAdapter`, …). Cambiare backend significa fornire un altro adapter, senza toccare facade e store.

- Capitolo principale: 21 — Livello OCC: adapter, connector, converter, endpoint
- Riferimenti: `core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30`, `core-libs/core/src/occ/adapters/product/product-occ.module.ts:L30-L76`
- Citato in 35 capitoli; i più frequenti oltre al principale: 52 (211), 24 (126), 25 (113), 23 (102), 36 (102), 35 (101), 15 (92), 27 (78) e altri 26

APP_INITIALIZER

`InjectionToken` di Angular per funzioni eseguite all'avvio che possono restituire Observable o Promise; `provideAppInitializer(fn)` lo avvolge in `makeEnvironmentProviders`. Il token è deprecato dalla v19 in favore di `provideAppInitializer`. Spartacus lo usa, tra l'altro, per eseguire i `CONFIG_INITIALIZER` e i `MODULE_INITIALIZER` dei moduli eager.

- Capitolo principale: 01 — Dependency Injection in Angular e in Spartacus
- Riferimenti: `angular:packages/core/src/application/application_init.ts:L150-L152`, `angular:packages/core/src/application/application_init.ts:L165-L202`
- Citato in 29 capitoli; i più frequenti oltre al principale: 10 (38), 11 (13), 33 (13), 13 (12), 34 (12), 12 (9), 05 (7), 14 (7) e altri 20

B

Base site

Parametro di contesto che identifica il sito SAP Commerce (il segmento `/electronics-spa/` nel path OCC), inserito nell'URL delle chiamate tramite `OccEndpointsService.getBaseUrl()`. La configurazione `context` può es-

sere derivata dinamicamente dal base site del backend il cui `urlPatterns` corrisponde all'URL corrente (`SiteContextConfigInitializer`, scope `context`).

- Capitolo principale: 13 — Site context e site theme
- Riferimenti: `core-libs/core/src/occ/services/occ-endpoints.service.ts:L86-L102`, `core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L24-L27`
- Citato in 23 capitoli; i più frequenti oltre al principale: 10 (49), 21 (35), 22 (25), 34 (19), 46 (17), 35 (12), 11 (10), 25 (10) e altri 14

BaseCoreModule / BaseStorefrontModule

Punto di aggregazione dell'inizializzazione: importando `BaseStorefrontModule` (che importa per primo `BaseCoreModule.forRoot()`, poi l'infrastruttura CMS/layout/routing/media/outlet) un'applicazione ottiene in un colpo solo l'intera catena di inizializzatori. `BaseCoreModule` aggrega 17 moduli core in un ordine esplicito.

- Capitolo principale: 10 — Bootstrap e ordine di inizializzazione
- Riferimenti: `core-libs/storefront/base-storefront.module.ts:L26-L49`, `core-libs/core/src/base-core.module.ts:L26-L53`
- Citato in 31 capitoli; i più frequenti oltre al principale: 02 (33), 06 (10), 41 (8), 47 (7), 25 (4), 52 (4), 11 (3), 12 (3) e altri 22

Breakpoint

Fascia di larghezza della finestra (`xs`…`xl`, enum `BREAKPOINT`). `BreakpointService` trasforma la larghezza in un valore osservabile, con fallback "mobile first" in SSR; `LayoutConfig` può definire slot diversi per breakpoint.

- Capitolo principale: 18 — Layout, breakpoint, launch dialog e accessibilità
- Riferimenti: `core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L40-L46`, `core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L123-L131`
- Citato in 35 capitoli; i più frequenti oltre al principale: 16 (36), 38 (20), 51 (12), 45 (8), 46 (8), 17 (6), 10 (4), 29 (4) e altri 26

C

CMS

Spartacus è uno storefront guidato dal CMS: le pagine non sono route Angular statiche ma una struttura pagina → template → slot → componenti che arriva da SAP Commerce Cloud (o da config statica o da un CMS di terze parti tramite adapter). Il modulo `cms` di core ottiene la struttura da un `PageContext`, separa struttura e dati dei componenti e la espone con la facade `CmsService`.

- Capitolo principale: 15 — CMS (core): pagine, slot, componenti, caricamento
- Riferimenti: `core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L30-L54`, `core-libs/core/src/cms/facade/cms.service.ts:L33-L46`
- Citato in 46 capitoli; i più frequenti oltre al principale: 28 (40), 45 (33), 51 (31), 53 (29), 16 (28), 33 (28), 31 (24), 03 (21) e altri 37

cmsComponents

Chiave di configurazione con cui lo storefront mappa i tipi di componente CMS su componenti Angular: la definizione è in core, la lettura nello storefront. L'implementazione può essere una classe Angular, `() => import(...)` o un web component, anche dentro feature module lazy (16).

- Capitolo principale: 15 — CMS (core): pagine, slot, componenti, caricamento
- Riferimenti: `core-libs/core/src/cms/config/cms-config.ts:L124-L138`, `core-libs/storefront/cms-structure/services/cms-components.service.ts:L67-L67`
- Citato in 28 capitoli; i più frequenti oltre al principale: 16 (23), 12 (18), 51 (14), 36 (13), 53 (11), 33 (8), 28 (7), 34 (6) e altri 19

CmsPageGuard

Guard della route wildcard `**` aggiunta in coda alla config del Router: carica la pagina CMS che corrisponde al `PageContext` della navigazione in corso e, se non esiste, mostra la pagina `notFound`.

- Capitolo principale: 14 — Routing: route semantiche, URL, page guard, contesto
- Riferimenti: `core-libs/storefront/cms-structure/routing/cms-route/add-cms-route.ts:L12-L25`, `core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L51-L84`
- Citato in 19 capitoli; i più frequenti oltre al principale: 05 (25), 51 (9), 10 (3), 15 (3), 16 (3), 27 (3), 31 (3), 12 (2) e altri 10

Command / Query

Astrazioni di `@spartacus/core` alternative a NgRx, usate dalle feature più recenti (es. checkout). `Command` modella le scritture con una strategia di concorrenza esplicita (`Queue`, `Parallel`, `CancelPrevious`, `ErrorPrevious`) e parte anche senza sottoscrizione; `Query` modella le letture con stato `{loading, error, data}`, caricamento lazy alla sottoscrizione e invalidazione tramite eventi (`reloadOn`, `resetOn`).

- Capitolo principale: 20 — Facade, command & query, eventi
- Riferimenti: `core-libs/core/src/util/command-query/command.service.ts:L27-L38`, `core-libs/core/src/util/command-query/command.service.ts:L122-L132`, `core-libs/core/src/util/command-query/query.service.ts:L33-L60`
- Citato in 15 capitoli; i più frequenti oltre al principale: 23 (17), 26 (13), 31 (13), 35 (7), 36 (7), 46 (6), 27 (5), 29 (4) e altri 6

CONFIG_INITIALIZER

Meccanismo per la configurazione che arriva in ritardo durante il bootstrap (config asincrona, ad esempio dal backend): gli initializer sono orchestrati da `ConfigInitializerService`, che tiene traccia degli scope in corso. Un location initializer trattiene la navigazione iniziale finché `getStable()` non emette.

- Capitolo principale: 11 — Configurazione: provideConfig, default, merge, validator, initializer
- Riferimenti: `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L130-L175`, `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L35-L37`
- Citato in 10 capitoli; i più frequenti oltre al principale: 10 (22), 01 (19), 13 (4), 37 (2), 50 (2), 02 (1), 06 (1), 14 (1) e altri 1

ConfigChunk / DefaultConfigChunk

I due multi token su cui si contribuisce configurazione: `DefaultConfigChunk` per le librerie, `ConfigChunk` per l'applicazione. I due insiemi vengono fusi separatamente e il root viene combinato per ultimo, quindi qualsiasi `ConfigChunk` vince su qualsiasi `DefaultConfigChunk` indipendentemente dall'ordine di import.

- Capitolo principale: 11 — Configurazione: provideConfig, default, merge, validator, initializer
- Riferimenti: `core-libs/core/src/config/config-tokens.ts:L49-L62`, `core-libs/core/src/config/config-tokens.ts:L10-L12`
- Citato in 17 capitoli; i più frequenti oltre al principale: 01 (20), 50 (13), 12 (10), 42 (10), 47 (6), 51 (6), 53 (6), 03 (3) e altri 8

Connector

Servizio `providedIn: 'root'` che delega all'adapter astratto della propria area (es. `ProductConnector` → `ProductAdapter`): è ciò che il dominio chiama, mentre l'implementazione concreta resta sostituibile.

- Capitolo principale: 21 — Livello OCC: adapter, connector, converter, endpoint
- Riferimenti: `core-libs/core/src/product/connectors/product/product.connector.ts:L13-L33`
- Citato in 25 capitoli; i più frequenti oltre al principale: 52 (62), 15 (59), 30 (50), 25 (32), 23 (25), 24 (17), 27 (15), 32 (15) e altri 16

Converter / normalizer / serializer

I dati raw (`Occ.*`) passano per `ConverterService`, che applica in catena tutti i converter registrati come multi provider su un `InjectionToken`: i *normalizer* convertono backend→UI, i *serializer* UI→backend.

- Capitolo principale: 21 — Livello OCC: adapter, connector, converter, endpoint
- Riferimenti: `core-libs/core/src/util/converter.service.ts:L13-L30`, `core-libs/core/src/util/converter.service.ts:L130-L137`
- Citato in 37 capitoli; i più frequenti oltre al principale: 52 (190), 46 (111), 15 (77), 31 (72), 13 (67), 25 (56), 24 (50), 30 (49) e altri 28

D

deepMerge

Merge ricorsivo che muta il primo argomento, usato dalle factory dei token `DefaultConfig`, `RootConfig` e `Config` per trasformare i chunk di configurazione in un unico oggetto.

- Capitolo principale: 11 — Configurazione: provideConfig, default, merge, validator, initializer
- Riferimenti: `core-libs/core/src/config/utils/deep-merge.ts:L18-L49`, `core-libs/core/src/config/config-tokens.ts:L10-L47`
- Citato in 26 capitoli; i più frequenti oltre al principale: 47 (7), 50 (7), 01 (6), 12 (6), 42 (6), 13 (5), 52 (5), 53 (5) e altri 17

E

EntityLoaderState

`EntityState<LoaderState<T>>`: una mappa di `LoaderState` indicizzata per id di entità, prodotta da `entityLoaderReducer` (composizione di `entityReducer` e `loaderReducer`). Il selettore restituisce lo stato iniziale per id inesistenti.

- Capitolo principale: 19 — State management: NgRx, loader/entity/process state
- Riferimenti: `core-libs/core/src/state/utils/entity-loader/entity-loader-state.ts:L10-L10`, `core-libs/core/src/state/utils/entity-loader/entity-loader.reducer.ts:L13-L25`
- Citato in 2 capitoli; i più frequenti oltre al principale: 15 (2)

Entry point

Pacchetto importabile (`@spartacus/<lib>` o secondario `@spartacus/<lib>/<entry>`) con il proprio `ng-package.json` e `public_api.ts`; è il contratto pubblico. Per convenzione `root/` è eager (facade astratte, modelli, eventi, token, config), mentre `core/`, `components/`, `occ/` sono lazy.

- Capitolo principale: 54 — API pubblica vs interna, upgrade, anti-pattern, checklist
- Riferimenti: `core-libs/setup/ssr/ng-package.json:L1-L6`, `docs/libs/library-structure.md:L4-L16`
- Citato in 26 capitoli; i più frequenti oltre al principale: 32 (8), 36 (6), 10 (5), 12 (4), 35 (4), 47 (4), 28 (3), 30 (3) e altri 17

Environment injector

Nodo dell'albero "environment" degli injector (`R3Injector`): il root injector applicativo di un avvio standalone è un `EnvironmentNgModuleRefAdapter` figlio del platform injector, e gli injector dei moduli lazy sono `NgModuleRef` con scope solo `'environment'`. `createEnvironmentInjector` è l'API pubblica per crearne uno figlio. Spartacus crea i componenti dei moduli lazy usando l'injector del modulo come environment injector (03).

- Capitolo principale: 01 — Dependency Injection in Angular e in Spartacus
- Riferimenti: `angular:packages/core/src/application/create_application.ts:L64-L79`, `angular:packages/core/src/render3/ng_module_ref.ts:L82-L95`, `angular:packages/core/src/render3/ng_module_ref.ts:L193-L205`
- Citato in 11 capitoli; i più frequenti oltre al principale: 03 (44), 02 (19), 17 (7), 12 (3), 04 (2), 06 (2), 10 (2), 07 (1) e altri 2

EventService

Bus di eventi tipizzato per classi: si può registrare una sorgente observable per un tipo di evento, dispatchare un'istanza o ascoltare un tipo (e quindi anche i suoi sottotipi). `StateEventService` trasforma azioni NgRx in eventi, e `@spartacus/storefront` aggiunge una catena di eventi di pagina per tracking e personalizzazione.

- Capitolo principale: 20 — Facade, command & query, eventi
- Riferimenti: `core-libs/core/src/event/event.service.ts:L47-L113`, `core-libs/core/src/state/event/state-event.service.ts:L16-L76`

- Citato in 22 capitoli; i più frequenti oltre al principale: 50 (13), 19 (8), 12 (5), 22 (4), 33 (4), 53 (4), 25 (3), 54 (3) e altri 13

F

Facade

Classe astratta che i componenti (e le altre feature) iniettano come API stabile della feature. Con `facadeFactory` viene fornita nel root injector come proxy che, alla prima chiamata di metodo o alla prima sottoscrizione di una proprietà, scarica il modulo lazy della feature e inoltra la chiamata all'implementazione reale: per questo un header può iniettare `ActiveCartFacade` senza portare nel bundle iniziale il codice del carrello.

- Capitolo principale: 20 — Facade, command & query, eventi
- Riferimenti: `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L22-L36`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24`, `feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L46`
- Citato in 42 capitoli; i più frequenti oltre al principale: 25 (184), 23 (153), 31 (148), 27 (147), 12 (137), 26 (123), 53 (115), 35 (112) e altri 33

facadeFactory

Funzione che crea la facade proxy chiamando `inject(FacadeFactoryService).create(descriptor)`, quindi va usata in un injection context (tipicamente `useFactory`). Il descrittore ha `facade` e `feature` obbligatori, `methods`, `properties` e `async` opzionali; la chiamata di un metodo del proxy scarica il modulo ed esegue il metodo reale anche se il chiamante non si sottoscrive.

- Capitolo principale: 20 — Facade, command & query, eventi
- Riferimenti: `core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24`, `core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L10-L37`, `core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L70-L82`
- Citato in 20 capitoli; i più frequenti oltre al principale: 12 (9), 35 (6), 53 (6), 25 (5), 26 (4), 36 (4), 01 (3), 29 (3) e altri 11

Feature module

Unità di codice di una feature lib caricata con il sistema di lazy loading di Spartacus, indipendente dal router: il codice va caricato quando il CMS chiede un tipo di componente della feature, non quando il router entra in una route. Il pattern è "root module eager + modulo lazy": il root module dichiara quali componenti CMS la feature copre, l'app fornisce dove trovare il codice (`module: () => import(...)`).

- Capitolo principale: 12 — Feature module e lazy loading in Spartacus
- Riferimenti: `feature-libs/cart/base/root/cart-base-root.module.ts:L77-L81`, `feature-libs/cart/base/root/cart-base-root.module.ts:L26-L83`, `projects/storefrontapp/src/app/spartacus/features/cart/cart-base-feature.module.ts:L23-L55`
- Citato in 34 capitoli; i più frequenti oltre al principale: 03 (19), 47 (17), 53 (15), 20 (12), 27 (11), 50 (11), 43 (10), 33 (9) e altri 25

Feature toggle

Flag booleano con nome che protegge una correzione scritta accanto al vecchio comportamento, così che il cliente la attivi quando è pronto. L'API attuale è `FeatureToggles` con `provideFeatureToggles(...)`, la legacy `FeaturesConfig` è deprecata; `FeaturesConfigModule.forRoot()` installa un ponte toggle → config perché la direttiva `*cxFeature` passa ancora dal servizio legacy.

- Capitolo principale: 42 — Feature toggle e deprecazioni
- Riferimenti: `core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L10-L14`, `core-libs/core/src/features-config/features-config.module.ts:L28-L52`, `core-libs/core/src/features-config/directives/feature.directive.ts:L11-L17`
- Citato in 35 capitoli; i più frequenti oltre al principale: 54 (23), 44 (19), 47 (17), 43 (10), 13 (7), 04 (6), 22 (6), 23 (6) e altri 26

featureModules

Chiave di configurazione dichiarativa che associa un nome di feature a un `import()` dinamico, a delle dipendenze e ai tipi di componente CMS che la feature implementa. È l'unico modo di dichiarare le feature lazy (con `provideConfig` / `provideDefaultConfig`).

- Capitolo principale: 12 — Feature module e lazy loading in Spartacus
- Riferimenti: `core-libs/core/src/cms/config/cms-config.ts:L109-L134`
- Citato in 29 capitoli; i più frequenti oltre al principale: 47 (15), 53 (12), 33 (11), 36 (10), 03 (8), 16 (8), 50 (7), 34 (6) e altri 20

G

getStable

Metodo di `ConfigInitializerService` che restituisce subito la configurazione se è stabile, altrimenti attende che gli scope richiesti non siano più in corso. Il JSDoc lo indica come il modo raccomandato per leggere la config da codice che può girare prima della fine dell'inizializzazione.

- Capitolo principale: 11 — Configurazione: provideConfig, default, merge, validator, initializer
- Riferimenti: `core-libs/core/src/config/config-initializer/config-initializer.service.ts:L35-L37`
- Citato in 13 capitoli; i più frequenti oltre al principale: 10 (23), 13 (11), 50 (9), 22 (8), 01 (6), 54 (3), 12 (2), 16 (2) e altri 4

H

HTTP_INTERCEPTORS

Multi token su cui Spartacus registra i propri interceptor HTTP con `useExisting`; l'app demo li abilita con `provideHttpClient(withFetch(), withInterceptorsFromDi())`. Poiché l'ordine di registrazione determina l'ordine nell'array, `ErrorHandlingModule.forRoot()` è importato per primo in `BaseCoreModule`.

- Capitolo principale: 01 — Dependency Injection in Angular e in Spartacus
- Riferimenti: `core-libs/core/src/auth/user-auth/http-interceptors/index.ts:L12-L23`, `projects/storefrontapp/src/app/app.config.ts:L26-L35`, `core-libs/core/src/base-core.module.ts:L28-L28`
- Citato in 16 capitoli; i più frequenti oltre al principale: 06 (18), 21 (4), 33 (4), 35 (4), 50 (4), 34 (3), 41 (3), 22 (2) e altri 7

HttpErrorHandler

Classe astratta degli handler di global message per gli errori HTTP, uno per status: ne sono registrati 8 come multi provider con `useExisting` e `HttpErrorInterceptor` li raccoglie con `UnifiedInjector.getMulti`. L'`hasMatch` di default confronta lo status della risposta con `responseStatus`.

- Capitolo principale: 41 — Errori HTTP, error handling, logger e global message
- Riferimenti: `core-libs/core/src/global-message/http-interceptors/index.ts:L22-L63`, `core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L42-L44`
- Citato in 12 capitoli; i più frequenti oltre al principale: 50 (12), 31 (6), 32 (6), 28 (5), 06 (4), 54 (4), 25 (3), 30 (3) e altri 3

Hydration

Nel browser Angular non ricostruisce da zero il DOM ricevuto dal server ma lo "reclama" nodo per nodo, grazie alle annotazioni (`ngh`, `__nghData__`) scritte durante il render server.

- Capitolo principale: 07 — SSR, hydration e TransferState
- Riferimenti: `angular:packages/core/src/hydration/annotate.ts:L243-L301`, `angular:packages/core/src/hydration/api.ts:L211-L228`
- Citato in 25 capitoli; i più frequenti oltre al principale: 06 (11), 02 (7), 45 (6), 47 (5), 05 (4), 39 (4), 16 (3), 19 (3) e altri 16

I

Injection token

Chiave DI non legata a una classe: `InjectionToken<T>` ha una descrizione usata solo per debug e, se al costruttore si passano `options`, un `ɵprov` con `providedIn` che vale `'root'` per default. Due istanze con la stessa descrizione sono token diversi. In Spartacus sono token DI, tra gli altri, la configurazione globale e i normalizer OCC.

- Capitolo principale: 01 — Dependency Injection in Angular e in Spartacus
- Riferimenti: `angular:packages/core/src/di/injection_token.ts:L110-L116`, `angular:packages/core/src/di/injection_token.ts:L26-L29`
- Citato in 16 capitoli; i più frequenti oltre al principale: 11 (5), 14 (5), 21 (5), 46 (5), 05 (3), 15 (3), 16 (3), 23 (3) e altri 7

L

LAUNCH_CALLER

Chiave che identifica un componente aperto da `LaunchDialogService` (dialog, popover, sidebar, ma anche navigazioni); il servizio sceglie una di quattro strategie di rendering in base alla configurazione. È uno dei tipi augmentabili di storefront (54).

- Capitolo principale: 18 — Layout, breakpoint, launch dialog e accessibilità
- Riferimenti: `core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L78-L97`, `core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L18-L41`
- Citato in 12 capitoli; i più frequenti oltre al principale: 31 (8), 54 (7), 25 (4), 29 (4), 32 (4), 30 (2), 34 (2), 24 (1) e altri 3

LayoutConfig

Configurazione che dice quali slot CMS mostrare in ogni sezione (header, footer, ...) o page template, eventualmente diversi per breakpoint, più i breakpoint stessi, la strategia di deferred loading e la configurazione dei dialog (`launch`). Il design adattivo è per breakpoint: il DOM viene (ri)renderizzato, non nascosto via CSS.

- Capitolo principale: 18 — Layout, breakpoint, launch dialog e accessibilità
- Riferimenti: `core-libs/storefront/layout/config/layout-config.ts:L90-L125`, `core-libs/storefront/layout/config/layout-config.ts:L84-L89`
- Citato in 12 capitoli; i più frequenti oltre al principale: 16 (7), 51 (7), 17 (5), 29 (4), 27 (3), 30 (3), 26 (2), 32 (2) e altri 3

LoaderState

Forma generica di una cella di stato NgRx caricata dal backend: `{ loading?, error?, success?, value? }`, con stato iniziale a flag `false` e `value: undefined`. È gestita dall'higher-order reducer `loaderReducer`, che interpreta `action.meta` invece di `action.type`.

- Capitolo principale: 19 — State management: NgRx, loader/entity/process state
- Riferimenti: `core-libs/core/src/state/utils/loader/loader-state.ts:L7-L12`, `core-libs/core/src/state/utils/loader/loader.reducer.ts:L11-L16`
- Citato in 6 capitoli; i più frequenti oltre al principale: 30 (2), 32 (2), 15 (1), 23 (1), 29 (1)

LoggerService

Servizio `providedIn: 'root'` che delega al `console` ed è pensato come extension point per redirigere i log; in SSR è sostituito da una versione server.

- Capitolo principale: 41 — Errori HTTP, error handling, logger e global message
- Riferimenti: `core-libs/core/src/logger/logger.service.ts:L9-L22`, `core-libs/setup/ssr/logger/services/server-logger-service-factory.ts:L12-L17`
- Citato in 10 capitoli; i più frequenti oltre al principale: 39 (18), 07 (4), 37 (4), 11 (2), 12 (2), 46 (2), 20 (1), 33 (1) e altri 1

M

Migrazione (ng update)

Trasformazione eseguita da `ng update` per adattare il codice del cliente ai breaking change: `ng-update.migrations` del `package.json` di `@spartacus/schematics` punta a `migrations.json`, che allo SHA contiene 25 voci per quattro versioni. Gli upgrade più invasivi sono affidati a schematics di modernizzazione lanciati a mano (`modernize-app-*`).

- Capitolo principale: 43 — Schematics e migrazioni di Spartacus
- Riferimenti: `core-libs/schematics/package.json:L46-L47`, `core-libs/schematics/src/migrations/migrations.json:L3-L127`
- Citato in 16 capitoli; i più frequenti oltre al principale: 08 (147), 54 (94), 44 (16), 02 (15), 04 (11), 42 (10), 07 (7), 33 (2) e altri 7

MODULE_INITIALIZER

Hook di inizializzazione per moduli lazy: fa per un modulo lazy ciò che `APP_INITIALIZER` fa per l'app. Se il modulo è eager viene eseguito in un `APP_INITIALIZER`, se è lazy alla creazione del modulo (01).

- Capitolo principale: 12 — Feature module e lazy loading in Spartacus
- Riferimenti: `core-libs/core/src/lazy-loading/tokens.ts:L9-L20`
- Citato in 12 capitoli; i più frequenti oltre al principale: 10 (13), 03 (12), 01 (11), 17 (7), 53 (5), 25 (4), 51 (3), 02 (2) e altri 3

Multi provider

Provider con `multi: true`: l'injector crea un record unico per il token e vi accumula i provider, per cui l'ordine di registrazione diventa l'ordine dell'array finale. Un multi token si ferma al primo injector che ha il record, quindi i multi provider registrati in moduli lazy non si vedono dal root: Spartacus li raccoglie da tutti gli injector con `UnifiedInjector.getMulti`. Converter, interceptor HTTP e chunk di configurazione sono multi provider.

- Capitolo principale: 01 — Dependency Injection in Angular e in Spartacus
- Riferimenti: `angular:packages/core/src/di/r3_injector.ts:L497-L512`, `angular:packages/core/src/di/r3_injector.ts:L347-L371`, `core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93`
- Citato in 30 capitoli; i più frequenti oltre al principale: 52 (15), 50 (10), 41 (9), 46 (9), 15 (6), 21 (6), 35 (6), 06 (5) e altri 21

N

NodeInjector

Injector dell'albero degli elementi: ogni componente/direttiva ne ha uno legato a `(tNode, lView)`. La ricerca risale i node injector (con un bloom filter) e infine ricade sull'environment injector; `Self`/`Host` limitano la risalita e impediscono il fallback.

- Capitolo principale: 01 — Dependency Injection in Angular e in Spartacus
- Riferimenti: `angular:packages/core/src/render3/di.ts:L893-L908`, `angular:packages/core/src/render3/di.ts:L407-L439`
- Citato in 1 capitoli

O

OCC

Omni Commerce Connect, l'API REST di SAP Commerce Cloud. Il "livello OCC" di Spartacus separa la logica di dominio dal formato delle chiamate HTTP con adapter astratti e connector, endpoint configurabili, converter impilabili e scope di caricamento.

- Capitolo principale: 21 — Livello OCC: adapter, connector, converter, endpoint
- Riferimenti: `core-libs/core/src/occ/services/occ-endpoints.service.ts:L111-L150`, `core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L54-L59`
- Citato in 38 capitoli; i più frequenti oltre al principale: 23 (30), 52 (26), 31 (24), 22 (21), 34 (19), 46 (19), 15 (16), 24 (16) e altri 29

OccEndpoints

Interfaccia della configurazione `backend.occ.endpoints`, in cui ogni voce è un template di URL o una mappa scope → template (`OccEndpoint`). È un tipo augmentabile: le feature lib la estendono via module augmentation di `@spartacus/core`.

- Capitolo principale: 21 — Livello OCC: adapter, connector, converter, endpoint
- Riferimenti: `core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L33`, `core-libs/core/public_api.ts:L50-L51`, `feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L97-L98`
- Citato in 11 capitoli; i più frequenti oltre al principale: 52 (11), 25 (4), 10 (2), 27 (2), 31 (2), 36 (2), 54 (2), 23 (1) e altri 2

OptimizedSsrEngine

Wrapper di `@spartacus/setup/ssr` attorno alla funzione view-engine Express che aggiunge timeout con fallback CSR, limite di concorrenza, cache in memoria dei render, rendering strategy per richiesta, condivisione di un render tra richieste identiche e protezione dai render appesi. Le opzioni e i loro default sono in `SsrOptimizationOptions`.

- Capitolo principale: 39 — SSR ottimizzato
- Riferimenti: `core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L37-L57`, `core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L18-L181`
- Citato in 3 capitoli; i più frequenti oltre al principale: 07 (6), 47 (6)

Outlet

Punto nominato in un template in cui l'applicazione può inserire contenuto prima, dopo o al posto di quello di default (`OutletPosition`: `before`, `after`, `replace`), senza fare fork dei template. Si dichiara con `[cxOutlet]`, si

registra con `[cxOutletRef]` o con `provideOutlet()`; il registro è `OutletService`. Storefront, layout di pagina, slot e componenti CMS sono tutti avvolti in outlet.

- Capitolo principale: 17 — Outlet
- Riferimenti: `core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L13`, `core-libs/storefront/cms-structure/outlet/outlet.service.ts:L10-L18`, `core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L34-L47`
- Citato in 41 capitoli; i più frequenti oltre al principale: 51 (227), 53 (64), 18 (62), 32 (48), 16 (36), 47 (29), 28 (25), 36 (25) e altri 32

P

Page layout

Disposizione degli slot di una pagina: `PageLayoutComponent` (`cx-page-layout`) calcola nome del layout e slot e rende un outlet e un `cx-page-slot` per slot; `PageLayoutService` risolve quali slot mostrare e in che ordine da `LayoutConfig`, in base a page template, sezione e breakpoint.

- Capitolo principale: 16 — CMS (storefront): rendering dinamico delle pagine
- Riferimenti: `core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L54-L80`
- Citato in 16 capitoli; i più frequenti oltre al principale: 30 (10), 51 (9), 14 (5), 18 (4), 25 (4), 05 (3), 10 (2), 15 (2) e altri 7

PageContext

Contesto che identifica una pagina CMS: `{ id: string; type?: PageType }`, con gli id speciali `HOME_PAGE_CONTEXT` e `SMART_EDIT_CONTEXT`; `PageType` vale `ContentPage`, `ProductPage`, `CategoryPage` o `CatalogPage`. È calcolato da un serializer custom del router-state NgRx (14).

- Capitolo principale: 15 — CMS (core): pagine, slot, componenti, caricamento
- Riferimenti: `core-libs/core/src/routing/models/page-context.model.ts:L15-L30`, `core-libs/core/src/model/cms.model.ts:L38-L42`
- Citato in 5 capitoli; i più frequenti oltre al principale: 14 (27), 51 (9), 05 (7), 33 (1)

PageMetaResolver

Resolver dei meta di pagina (`PageMeta`: title, heading, description, robots, image, breadcrumbs, canonical). `PageMetaService` sceglie tra più `PageMetaResolver` quello con lo score migliore per tipo, template e uid della pagina; il resolver espone metodi `resolve*` che restituiscono `Observable` e una configurazione decide quale metodo alimenta quale proprietà.

- Capitolo principale: 40 — SEO e PWA
- Riferimenti: `core-libs/core/src/cms/page/page-meta.resolver.ts:L31-L45`, `core-libs/core/src/cms/page/config/default-page-meta.config.ts:L9-L40`
- Citato in 10 capitoli; i più frequenti oltre al principale: 52 (18), 24 (7), 31 (6), 28 (4), 26 (3), 51 (3), 15 (2), 46 (2) e altri 1

PendingTasks

API pubblica di Angular che tiene traccia dei task che bloccano la stabilità dell'applicazione; in v21 `ApplicationRef.isStable` deriva dal contatore di `PendingTasksInternal`, ed è ciò che l'SSR attende prima di serializzare. Spartacus la usa per tenere aperta la stabilità durante l'`import()` dinamico dei componenti CMS lazy.

- Capitolo principale: 04 — Change detection, Zone, zoneless e signals
- Riferimenti: `angular:packages/core/src/pending_tasks.ts:L18-L42`, `angular:packages/core/src/application/application_ref.ts:L326-L334`, `core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L60-L71`
- Citato in 8 capitoli; i più frequenti oltre al principale: 07 (16), 03 (6), 16 (6), 06 (3), 02 (2), 12 (1), 51 (1)

Process state

`ProcessesLoaderState` aggiunge a `LoaderState` il contatore `processesCount` (stabile = nessun processo e nessun caricamento). `ProcessModule` registra la feature NgRx generica `process`, un `EntityLoaderState` indicizzato per process id, per tracciare operazioni "fire and forget" senza una slice dedicata per ognuna.

- Capitolo principale: 19 — State management: NgRx, loader/entity/process state
- Riferimenti: `core-libs/core/src/state/utils/processes-loader/processes-loader-state.ts:L9-L11`, `core-libs/core/src/process/store/process-store.module.ts:L13-L17`, `core-libs/core/src/state/utils/processes-loader/processes-loader.selectors.ts:L9-L17`
- Citato in 7 capitoli; i più frequenti oltre al principale: 25 (12), 30 (4), 27 (3), 02 (2), 28 (1), 42 (1)

provideConfig / provideDefaultConfig

Helper che producono i provider dei chunk di configurazione: `provideConfig` registra un `ValueProvider` su `ConfigChunk` (o su `DefaultConfigChunk` se `defaultConfig`), `provideDefaultConfig` su `DefaultConfigChunk`; le varianti `*Factory` producono un `FactoryProvider`. Una regola ESLint di workspace vieta `provideConfig()` nei file `**/*.module.ts` delle librerie.

- Capitolo principale: 11 — Configurazione: provideConfig, default, merge, validator, initializer
- Riferimenti: `core-libs/core/src/config/config-providers.ts:L18-L27`, `core-libs/core/src/config/config-providers.ts:L55-L61`, `core-libs/core/src/config/config-providers.ts:L69-L79`
- Citato in 45 capitoli; i più frequenti oltre al principale: 50 (41), 01 (32), 51 (29), 53 (27), 52 (19), 54 (19), 42 (18), 47 (15) e altri 36

public_api

File indicato come `entryFile` nell'`ng-package.json` di ogni pacchetto o entry point: ciò che è raggiungibile da lì è la superficie pubblica coperta dal versionamento semantico. Il resto, anche se `export`ato in un file sorgente, è implementazione e può cambiare senza preavviso.

- Capitolo principale: 54 — API pubblica vs interna, upgrade, anti-pattern, checklist
- Riferimenti: `core-libs/core/ng-package.json:L1-L7`, `core-libs/core/public_api.ts:L7-L36`
- Citato in 45 capitoli; i più frequenti oltre al principale: 53 (70), 52 (55), 50 (34), 36 (24), 33 (22), 51 (19), 10 (15), 34 (15) e altri 36

R

Root module (feature lib)

Modulo piccolo di una feature lib (es. `CartBaseRootModule`), importato eager, che dichiara quali componenti CMS la feature copre. Le facade astratte vivono nello stesso entry point `root` come classi `providedIn: 'root'` con factory `facadeFactory`.

- Capitolo principale: 12 — Feature module e lazy loading in Spartacus
- Riferimenti: `feature-libs/cart/base/root/cart-base-root.module.ts:L77-L81`, `feature-libs/cart/base/root/facade/active-cart.facade.ts:L13-L16`
- Citato in 23 capitoli; i più frequenti oltre al principale: 53 (17), 35 (13), 24 (5), 32 (5), 25 (4), 26 (4), 33 (4), 34 (4) e altri 14

Route semantica

Meccanismo che toglie gli URL dal codice dei componenti: `RoutingConfig.routing.routes` è un dizionario nome route → `{ paths, matchers, paramsMapping, ... }`; le route Angular dichiarano solo il nome in `data.cxRoute` e all'avvio `ConfigurableRoutesService` ne riscrive `path`/`matcher`. I link si generano con i comandi `{ cxRoute, params }`, tradotti da `SemanticPathService` e dalla pipe `cxUrl`.

- Capitolo principale: 14 — Routing: route semantiche, URL, page guard, contesto
- Riferimenti: `core-libs/core/src/routing/configurable-routes/routes-config.ts:L16-L57`, `core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L77-L109`, `core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L41-L68`
- Citato in 26 capitoli; i più frequenti oltre al principale: 51 (12), 05 (11), 36 (9), 30 (8), 18 (7), 28 (7), 31 (6), 20 (5) e altri 17

S

Schematics

Trasformazioni di codice su un file system virtuale, motore di `ng add`, `ng generate` e `ng update` (08). Il pacchetto `@spartacus/schematics` le usa per installare Spartacus in un'app Angular, per i generatori di supporto e per le migrazioni; le sue utility servono agli schematics delle altre librerie ma non sono API pubblica.

- Capitolo principale: 43 — Schematics e migrazioni di Spartacus
- Riferimenti: `core-libs/schematics/package.json:L43-L89`, `core-libs/schematics/index.ts:L7-L13`
- Citato in 34 capitoli; i più frequenti oltre al principale: 08 (490), 44 (64), 02 (55), 36 (49), 34 (48), 54 (48), 31 (41), 33 (34) e altri 25

SERVER_REQUEST_URL / WindowRef

`SERVER_REQUEST_URL` e `SERVER_REQUEST_ORIGIN` sono i token con cui Spartacus espone URL e origin della richiesta in SSR; li consuma `WindowRef`, che in SSR restituisce una `location` parziale con solo `href` e `origin` e lancia un errore se i token mancano.

- Capitolo principale: 07 — SSR, hydration e TransferState
- Riferimenti: `core-libs/core/src/util/ssr.tokens.ts:L9-L21`, `core-libs/core/src/window/window-ref.ts:L19-L27`, `core-libs/core/src/window/window-ref.ts:L62-L89`
- Citato in 16 capitoli; i più frequenti oltre al principale: 46 (22), 13 (9), 45 (5), 40 (3), 19 (2), 34 (2), 47 (2), 14 (1) e altri 7

Site context

Sottosistema di `@spartacus/core` che mantiene la terna attiva base site / lingua / valuta, la inizializza (URL → `localStorage` → default di config), la codifica nell'URL e la propaga a router, HTTP e UI. È guidato dalla chiave di configurazione `context`, in cui il default di ogni parametro è il primo elemento del suo array.

- Capitolo principale: 13 — Site context e site theme
- Riferimenti: `core-libs/core/src/site-context/providers/context-ids.ts:L7-L10`, `core-libs/core/src/site-context/config/site-context-config.ts:L10-L19`, `core-libs/core/src/site-context/config/context-config-utils.ts:L28-L34`
- Citato in 26 capitoli; i più frequenti oltre al principale: 10 (29), 06 (16), 21 (16), 45 (15), 11 (14), 05 (12), 02 (10), 37 (8) e altri 17

Slot

Area di una pagina CMS che contiene componenti. Nel modello normalizzato lo slot porta solo riferimenti leggeri ai componenti (`uid`, `typeCode`, `flexType`, `properties`), mentre i dati completi stanno in un ramo separato dello store indicizzato per `uid`. Lato storefront ogni slot è reso da `PageSlotComponent` (16).

- Capitolo principale: 15 — CMS (core): pagine, slot, componenti, caricamento
- Riferimenti: `core-libs/core/src/cms/model/content-slot-component-data.model.ts:L7-L12`, `core-libs/core/src/cms/store/cms-state.ts:L22-L62`
- Citato in 22 capitoli; i più frequenti oltre al principale: 16 (122), 51 (47), 17 (36), 18 (25), 33 (19), 04 (6), 30 (6), 32 (5) e altri 13

SSR (Server-Side Rendering)

L'applicazione viene avviata in Node su una platform server con DOM emulato, si attende che diventi stabile e il DOM viene serializzato in HTML. Spartacus usa il `CommonEngine` di `@angular/ssr/node` avvolto in `CxCommonEngine` e in un `ngExpressEngine`, ed espone URL e origin della richiesta con i token `SERVER_REQUEST_URL` / `SERVER_REQUEST_ORIGIN`.

- Capitolo principale: 07 — SSR, hydration e TransferState
- Riferimenti: `angular:packages/platform-server/src/utils.ts:L342-L380`, `core-libs/setup/ssr/engine/cx-common-engine.ts:L7-L23`, `core-libs/core/src/util/ssr.tokens.ts:L9-L21`
- Citato in 41 capitoli; i più frequenti oltre al principale: 44 (36), 41 (26), 39 (24), 46 (21), 04 (20), 16 (20), 37 (19), 47 (18) e altri 32

T

Tipi augmentabili

Tipi pensati per la module augmentation, riesportati per nome nei blocchi `AUGMENTABLE_TYPES_START..._END` dei `public_api.ts` (in core, ad esempio `Config`, `Product`, `OccEndpoints`; in storefront `ICON_TYPE`, `BREAKPOINT`, `LAUNCH_CALLER`). Le librerie stesse li estendono e aggiungono i valori runtime degli enum con un cast; i `const enum` sono vietati da una regola ESLint.

- Capitolo principale: 54 — API pubblica vs interna, upgrade, anti-pattern, checklist
- Riferimenti: `core-libs/core/public_api.ts:L38-L58`, `core-libs/storefront/public_api.ts:L32-L36`, `feature-libs/checkout/b2b/root/model/augmented-types.ts:L11-L23`
- Citato in 41 capitoli; i più frequenti oltre al principale: 52 (46), 53 (21), 36 (20), 46 (18), 26 (16), 18 (15), 32 (15), 08 (12) e altri 32

TransferState

Dizionario chiave→valore JSON che il server serializza in `<script id="ng-state">` e il browser rilegge al bootstrap. Spartacus vi trasferisce lo state NgRx (non le risposte HTTP) con un meta-reducer, sotto la chiave `cx-state`, e per questo disattiva la HTTP transfer cache di Angular con `withNoHttpTransferCache()`.

- Capitolo principale: 07 — SSR, hydration e TransferState
- Riferimenti: `angular:packages/core/src/transfer_state.ts:L66-L79`, `core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16`, `projects/storefrontapp/src/app/app.config.ts:L29-L29`
- Citato in 20 capitoli; i più frequenti oltre al principale: 19 (29), 15 (6), 13 (5), 02 (4), 39 (4), 06 (3), 03 (2), 04 (2) e altri 11

U

UnifiedInjector

Ponte fra il DI di Angular e i moduli lazy di Spartacus: `injectors$` emette il root injector e poi l'injector di ogni modulo lazy inizializzato; `get` interroga il root normalmente e gli altri con `{self: true}`; `getMulti` accumula gli array di un multi token da tutti gli injector. Così i servizi del root vedono multi provider e config chunk registrati in moduli caricati più tardi.

- Capitolo principale: 01 — Dependency Injection in Angular e in Spartacus
- Riferimenti: `core-libs/core/src/lazy-loading/unified-injector.ts:L32-L35`, `core-libs/core/src/lazy-loading/unified-injector.ts:L52-L68`, `core-libs/core/src/lazy-loading/unified-injector.ts:L79-L93`
- Citato in 20 capitoli; i più frequenti oltre al principale: 03 (13), 12 (11), 11 (4), 16 (4), 21 (4), 28 (4), 40 (4), 50 (4) e altri 11

Z

Zoneless

Modalità di change detection senza Zone.js: sono le API Angular (`markForCheck`, `setInput`, listener, signal letti in template, effect…) a notificare uno scheduler che pianifica un tick. Dalla v21 è il default di Angular, ma la demo Spartacus fornisce esplicitamente `provideZoneChangeDetection({ eventCoalescing: true })`, quindi gira con Zone.

- Capitolo principale: 04 — Change detection, Zone, zoneless e signals
- Riferimenti: `angular:packages/core/src/application/create_application.ts:L64-L71`, `projects/storefrontapp/src/app/app.config.ts:L26-L33`
- Citato in 4 capitoli; i più frequenti oltre al principale: 02 (16), 06 (1), 07 (1)

Candidati ricorrenti non definiti

Termini in grassetto o in backtick che compaiono in almeno 6 capitoli e non sono coperti da una voce del glossario (estrazione automatica, senza definizione). I simboli dichiarati nel codice sono descritti in A12.

TERMINE	FONTE	CAPITOLI
`Config`	backtick	34
`forRoot()`	backtick	24
`inject()`	backtick	18
`BehaviorSubject`	backtick	17
`EMPTY`	backtick	16
`TypeError`	backtick	15
`RoutingConfig`	backtick	14
`AppModule`	backtick	12
`ReplaySubject`	backtick	12
`UrlTree`	backtick	12
`import()`	backtick	11
`ModuleWithProviders`	backtick	11
`SpartacusFeaturesModule`	backtick	11
`ConfigurationService`	backtick	10
`getActive()`	backtick	10
`Injector`	backtick	10
`Map`	backtick	10
`AuthGuard`	backtick	9

TERMINE	FONTE	CAPITOLI
CxEvent	backtick	9
initialize()	backtick	9
SpartacusModule	backtick	9
USE_CLIENT_TOKEN	backtick	9
AppRoutingModule	backtick	8
BEFORE_CMS_PAGE_GUARD	backtick	8
CmsComponentsService	backtick	8
CmsConfig	backtick	8
CmsFeaturesService	backtick	8
ComponentFactoryResolver	backtick	8
DefaultConfig	backtick	8
eager	grassetto	8
Error	backtick	8
getBaseUrl()	backtick	8
LANGUAGE_CHANGE	backtick	8
LaunchDialogService	backtick	8
NgModuleRef	backtick	8
R3Injector	backtick	8
RootConfig	backtick	8
Router	backtick	8
SERVER_REQUEST_ORIGIN	backtick	8
ActionsSubject	backtick	7
CartBaseModule	backtick	7
CmsRoutesImplService	backtick	7
FeatureModuleConfig	backtick	7
HttpErrorInterceptor	backtick	7
isDevMode()	backtick	7
LazyModulesService	backtick	7
LOCATION_INITIALIZED	backtick	7
LOCATION_INITIALIZED_MULTI	backtick	7
NavigationEnd	backtick	7
OrderEntry	backtick	7
PageSlotComponent	backtick	7
Renderer2	backtick	7

TERMINE	FONTE	CAPITOLI
StorefrontComponent	backtick	7
Subject	backtick	7
APP_BOOTSTRAP_LISTENER	backtick	6
CancelPrevious	backtick	6
CART_BASE_FEATURE	backtick	6
CartBaseWrapperModule	backtick	6
ComponentFactory	backtick	6
ComponentRef	backtick	6
ComponentWrapperDirective	backtick	6
ConfigInitializer	backtick	6
ConfigInitializerService	backtick	6
CURRENCY_CHANGE	backtick	6
ENVIRONMENT_INITIALIZER	backtick	6
ErrorHandler	backtick	6
FeatureModulesService	backtick	6
get()	backtick	6
GuardsComposer	backtick	6
HttpClient	backtick	6
init()	backtick	6
lazy	grassetto	6
LazyComponentHandler	backtick	6
LoginEvent	backtick	6
LOGOUT	backtick	6
LogoutEvent	backtick	6
OccConfig	backtick	6
OccEndpointsService	backtick	6
OrderModule	backtick	6
OrderWrapperModule	backtick	6
PageMetaService	backtick	6
Product	backtick	6
redux devtools	grassetto	6
Set	backtick	6
TestConfigModule	backtick	6
TranslationChunksConfig	backtick	6

TERMINE	FONTE	CAPITOLI
`using()`	backtick	6
`ViewContainerRef`	backtick	6
`withInterceptorsFromDi()`	backtick	6

Legenda capitoli:

- 01 Dependency Injection in Angular e in Spartacus
- 02 NgModule, standalone e bootstrap dell'applicazione
- 03 Lazy loading: router e moduli caricati a runtime
- 04 Change detection, Zone, zoneless e signals
- 05 Il Router di Angular
- 06 HttpClient e interceptor
- 07 SSR, hydration e TransferState
- 08 Schematics e builder
- 10 Bootstrap e ordine di inizializzazione
- 11 Configurazione: provideConfig, default, merge, validator, initializer
- 12 Feature module e lazy loading in Spartacus
- 13 Site context e site theme
- 14 Routing: route semantiche, URL, page guard, contesto
- 15 CMS (core): pagine, slot, componenti, caricamento
- 16 CMS (storefront): rendering dinamico delle pagine
- 17 Outlet
- 18 Layout, breakpoint, launch dialog e accessibilità
- 19 State management: NgRx, loader/entity/process state
- 20 Facade, command & query, eventi
- 21 Livello OCC: adapter, connector, converter, endpoint
- 22 Autenticazione: OAuth, token, interceptor, refresh
- 23 Utente, profilo, consensi
- 24 Prodotto e ricerca
- 25 Carrello
- 26 Checkout
- 27 Ordini
- 28 B2B: organization
- 29 ASM (Assisted Service Module)
- 30 Product configurator
- 31 Quote, customer ticketing, subscription billing
- 32 Pickup in store, store finder, delivery date
- 33 Tracking, SmartEdit, Qualtrics
- 34 Integration libs: CDC, CDP, CDS, segment-refs
- 35 Integration libs: OPF, digital payments, OPPS

- 36 Integration libs: S/4, OMF, CPQ quote, punchout, EPD
- 37 Internazionalizzazione (i18n)
- 38 Stili e theming
- 39 SSR ottimizzato
- 40 SEO e PWA
- 41 Errori HTTP, error handling, logger e global message
- 42 Feature toggle e deprecazioni
- 43 Schematics e migrazioni di Spartacus
- 44 Strategia di test e tooling del monorepo
- 45 Componenti CMS e shared components di storefront
- 46 Utility, window e modelli di dominio
- 47 L'app demo storefrontapp
- 50 Ricette: configurazione, DI, interceptor, eventi
- 51 Ricette: componenti CMS, outlet, route e layout
- 52 Ricette: OCC, adapter, converter, endpoint, meta resolver
- 53 Ricette: feature lib lazy, carrello/checkout, stili, traduzioni
- 54 API pubblica vs interna, upgrade, anti-pattern, checklist

A12

Indice dei simboli

Simbolo → dichiarazione → capitoli

IN QUESTO CAPITOLO

Conteggi
Come rigenerare
Indice per lettera
A
B
C
D
E
F
G
H
I
J
K

Generata da `tools/extract3/a12_indice_simboli.py` allo SHA 5b7a1c2b (`5b7a1c2bf2808d7380bc20596d10f4fa710c3da1`). Non modificare a mano: rigenerare.

Indice dei simboli TypeScript citati in backtick nei capitoli e dichiarati come `export` nel codice. Per ogni simbolo: tipo e posizione della dichiarazione (`path:Lx-Ly`; se il nome è dichiarato in più file si mostra la dichiarazione di produzione più vicina alla radice e il numero di dichiarazioni ulteriori) e i capitoli che lo citano, con il numero di occorrenze tra parentesi.

Criteri: token = span inline in backtick fuori dai blocchi di codice; identificatore plausibile = PascalCase, CONSTANT_CASE di almeno 4 caratteri, o camelCase seguito da `()` (stesso criterio di `tools/check-refs.py`); dichiarazione = `export (abstract )?(class|interface|const|function|enum|type) NOME` in file `.ts` tracciati di `core-libs/`, `feature-libs/`, `integration-libs/`, esclusi `*.spec.ts`. I simboli citati solo dentro blocchi di codice, o dichiarati solo in Angular/dipendenze, non compaiono.

Conteggi

- Simboli indicizzati: **2217**
- Occorrenze in backtick conteggiate: **8545**
- Capitoli scansionati: **51**
- Simboli per tipo di dichiarazione (prima dichiarazione): **class 1556, const 283, enum 73, function 26, interface 242, type 37**

Come rigenerare

```bash
cd docs/spartacus-book
python3 tools/extract3/a12_indice_simboli.py
# oppure entrambe le appendici: bash tools/extract3/run_all.sh
```

Indice per lettera

A · B · C · D · E · F · G · H · I · J · K · L · M · N · O · P · Q · R · S · T · U · V · W

A

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
`AbstractBrowserTabNotificationService`	class	`core-libs/core/src/util/browser-tab-notification/abstract-browser-tab-notification.service.ts:L49-L135`	46 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
AbstractOrderContext	class	feature-libs/cart/base/components/abstract-order-context/abstract-order-context.model.ts:L16-L18	30 (1)
AbstractOrderKey	type	feature-libs/cart/base/root/models/cart.model.ts:L133-L137	25 (2)
AbstractStoreItemComponent	class	feature-libs/storefinder/components/abstract-store-item/abstract-store-item.component.ts:L12-L24	32 (2)
AccountSummaryFacade	class	feature-libs/organization/account-summary/root/facade/account-summary.facade.ts:L30-L60	28 (3)
AccountSummaryModule	class	feature-libs/organization/account-summary/account-summary.module.ts:L21-L21	28 (1)
AccountSummaryPageMetaResolver	class	feature-libs/organization/account-summary/core/account-summary-page-meta.resolver.ts:L31-L104	28 (1)
AccountSummaryRootModule	class	feature-libs/organization/account-summary/root/account-summary-root.module.ts:L59-L59	28 (1)
AccountSummaryService	class	feature-libs/organization/account-summary/core/facade/account-summary.service.ts:L21-L83	28 (2)
AccountSummaryUnitListService	class	feature-libs/organization/account-summary/components/services/account-summary-unit-list.service.ts:L15-L17	28 (1)
ActionToEventMapping	interface	core-libs/core/src/state/event/action-to-event-mapping.ts:L9-L24	19 (2)
ActivatedRouterStateSnapshot	interface	core-libs/core/src/routing/store/routing-state.ts:L18-L25	14 (1), 20 (1)
ActivatedRoutesService	class	core-libs/core/src/routing/services/activated-routes.service.ts:L16-L40	14 (1)
ActiveCartFacade	class	feature-libs/cart/base/root/facade/active-cart.facade.ts:L46-L182	12 (1), 20 (2), 25 (5), 50 (1), 53 (1)
ActiveCartOrderEntriesContext	class	feature-libs/cart/base/components/page-context/active-cart-order-entries.context.ts:L24-L58	25 (2)
ActiveCartOrderEntriesContextToken	const	feature-libs/cart/base/root/tokens/context.ts:L9-L11	14 (1), 25 (3)
ActiveCartService	class	feature-libs/cart/base/core/facade/active-cart.service.ts:L50-L737	25 (7)
ActiveCartStatePersistenceService	class	feature-libs/cart/base/core/services/active-cart-state-persistence.service.ts:L23-L114	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ActiveLinkCellComponent	class	feature-libs/organization/administration/components/shared/table/active-link/active-link-cell.component.ts:L19-L23	28 (1)
ADD_CONTAINER_ROW_SUCCESS	const	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L31-L32	30 (1)
ADD_MESSAGE	const	core-libs/core/src/global-message/store/actions/global-message.actions.ts:L13-L13	41 (2)
ADD_TO_CART_FEATURE	const	feature-libs/cart/base/root/feature-name.ts:L11-L11	32 (2), 47 (2)
ADD_TO_WISHLIST_FEATURE	const	feature-libs/cart/wish-list/root/feature-name.ts:L10-L10	47 (1)
AddedToCartDialogEventListener	class	feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog-event.listener.ts:L22-L91	18 (1), 25 (1)
AddedToCartDialogModule	class	feature-libs/cart/base/components/added-to-cart-dialog/added-to-cart-dialog.module.ts:L50-L54	25 (1)
AdditionalFeatureConfiguration	interface	core-libs/schematics/src/shared/utils/feature-utils.ts:L71-L80	43 (1)
AdditionalProviders	interface	core-libs/schematics/src/shared/utils/feature-utils.ts:L63-L66	43 (1)
AddLocation	const	feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L49-L52	32 (1)
AddMessage	class	core-libs/core/src/global-message/store/actions/global-message.actions.ts:L18-L21	41 (2)
AddNextOwner	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L107-L110	30 (1)
AddOrderEntriesContext	interface	feature-libs/cart/base/root/context/add-order-entries.context.ts:L14-L31	25 (1)
Address	interface	core-libs/core/src/model/address.model.ts:L38-L67 (+1)	46 (5), 52 (1), 54 (1)
ADDRESS_NORMALIZER	const	core-libs/core/src/user/connectors/address/converters.ts:L12-L14	23 (1)
ADDRESS_SERIALIZER	const	core-libs/core/src/user/connectors/address/converters.ts:L20-L22	23 (1), 52 (6)
ADDRESS_VALIDATION_NORMALIZER	const	core-libs/core/src/user/connectors/address/converters.ts:L24-L26	23 (1)
AddressBookComponentService	class	feature-libs/user/profile/components/address-book/address-book.component.service.ts:L14-L48	23 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
AddressValidation	interface	core-libs/core/src/model/address.model.ts:L69-L74 (+1)	46 (2)
AddToCart	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L91-L96 (+1)	30 (1)
AddToCartComponent	class	feature-libs/cart/base/components/add-to-cart/add-to-cart.component.ts:L77-L348	17 (1), 25 (2), 32 (2), 36 (1), 53 (9)
AddToCartModule	class	feature-libs/cart/base/components/add-to-cart/add-to-cart.module.ts:L48-L48	53 (1)
AddToHomeScreenComponent	class	core-libs/storefront/cms-structure/pwa/components/add-to-home-screen.component.ts:L12-L23	40 (3)
AddToHomeScreenService	class	core-libs/storefront/cms-structure/pwa/services/add-to-home-screen.service.ts:L19-L72	40 (5)
AddUserAddress	class	core-libs/core/src/user/store/actions/user-addresses.action.ts:L56-L62	23 (1)
AddUserAddressEvent	class	core-libs/core/src/user/events/user.events.ts:L25-L28	23 (2)
AdminGuard	class	feature-libs/organization/administration/core/guards/admin.guard.ts:L20-L50	28 (8)
AdministrationComponentsModule	class	feature-libs/organization/administration/components/administration-components.module.ts:L25-L25	28 (1)
AdministrationModule	class	feature-libs/organization/administration/administration.module.ts:L19-L19	28 (6), 34 (1)
AdministrationOccModule	class	feature-libs/organization/administration/occ/administration-occ.module.ts:L168-L168	28 (2)
AdministrationRootModule	class	feature-libs/organization/administration/root/administration-root.module.ts:L64-L64	28 (3)
AepCollectorConfig	interface	feature-libs/tracking/tms/aep/config/default-aep.config.ts:L10-L12	33 (2)
AepCollectorService	class	feature-libs/tracking/tms/aep/services/aep-collector.service.ts:L20-L45	33 (5)
AepModule	class	feature-libs/tracking/tms/aep/aep.module.ts:L14-L14	33 (3), 47 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
AmountCellComponent	class	feature-libs/organization/administration/components/shared/table/amount/amount-cell.component.ts:L19-L34	28 (1)
ANONYMOUS_CONSENT_NORMALIZER	const	core-libs/core/src/anonymous-consents/connectors/converters.ts:L11-L13	23 (1)
ANONYMOUS_CONSENT_STATUS	enum	core-libs/core/src/model/consent.model.ts:L27-L30	46 (1)
ANONYMOUS_CONSENTS_HEADER	const	core-libs/core/src/model/consent.model.ts:L32-L32	46 (1)
AnonymousConsentCheckUpdatedVersions	class	core-libs/core/src/anonymous-consents/store/actions/anonymous-consents.action.ts:L121-L127	23 (1)
AnonymousConsentManagementBannerComponent	class	core-libs/storefront/cms-components/anonymous-consent-management/banner/anonymous-consent-management-banner.component.ts:L20-L60	23 (1)
AnonymousConsentsConfig	class	core-libs/core/src/anonymous-consents/config/anonymous-consents-config.ts:L15-L43	23 (2)
AnonymousConsentsInterceptor	class	core-libs/core/src/anonymous-consents/http-interceptors/anonymous-consents-interceptor.ts:L28-L136	06 (7), 23 (1)
AnonymousConsentsService	class	core-libs/core/src/anonymous-consents/facade/anonymous-consents.service.ts:L22-L377	23 (3)
app()	function	core-libs/schematics/src/migrations/2211_36/ssr/ssr_spec.ts:L48-L55 (+9)	47 (2)
AppComponent	class	core-libs/schematics/src/add-spartacus/add-imports-to-component-decorator_spec.ts:L28-L36 (+4)	02 (4), 10 (1), 43 (1), 47 (1)
Applicable	interface	core-libs/core/src/util/applicable.ts:L32-L45	37 (2), 40 (2), 41 (2), 46 (3)
AppModule	class	core-libs/schematics/src/add-spartacus/create-app-module.ts:L64-L65 (+10)	01 (2), 02 (18), 04 (4), 05 (1), 06 (1), 10 (2), 19 (2), 41 (1), 47 (5), 50 (5), 51 (1), 52 (2)
AppRoutingModule	class	core-libs/storefront/router/app-routing.module.ts:L30-L30 (+1)	01 (1), 02 (10), 05 (6), 06 (1), 10 (7), 14 (4), 43 (2), 47 (2)
ApproverGuard	class	feature-libs/organization/order-approval/core/guards/approver.guard.ts:L22-L54	28 (1)
AppServerModule	class	core-libs/setup/ssr/providers/server-request-origin.ts:L52-L52 (+1)	02 (3), 07 (2), 41 (1), 43 (1), 47 (2)
ASM_CUSTOMER_360_NORMALIZER	const	feature-libs/asm/customer-360/core/connectors/converters.ts:L11-L13	29 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ASM_FEATURE	const	feature-libs/asm/root/feature-name.ts:L7-L7 (+2)	29 (1)
AsmAdapter	class	feature-libs/asm/core/connectors/asm.adapter.ts:L18-L47	29 (2)
AsmAuthHttpHeaderService	class	feature-libs/asm/root/services/asm-auth-http-header.service.ts:L33-L168	06 (1), 22 (1), 29 (5)
AsmAuthService	class	feature-libs/asm/root/services/asm-auth.service.ts:L41-L215	22 (1), 29 (8)
AsmAuthStorageService	class	feature-libs/asm/root/services/asm-auth-storage.service.ts:L37-L139	22 (1), 29 (5), 34 (1)
AsmBindCartComponent	class	feature-libs/asm/components/asm-bind-cart/asm-bind-cart.component.ts:L76-L361	29 (2), 54 (1)
AsmBindCartFacade	class	feature-libs/asm/root/facade/asm-bind-cart.facade.ts:L21-L26	29 (2), 54 (2)
AsmBindCartService	class	feature-libs/asm/core/facade/asm-bind-cart.service.ts:L16-L46	29 (2), 54 (2)
AsmComponentService	class	feature-libs/asm/components/services/asm-component.service.ts:L25-L119	29 (1)
AsmComponentsModule	class	feature-libs/asm/components/asm-components.module.ts:L107-L107	29 (2)
AsmConfig	class	feature-libs/asm/root/config/asm-config.ts:L15-L55	29 (5)
AsmConnector	class	feature-libs/asm/core/connectors/asm.connector.ts:L23-L47	29 (1)
AsmCoreModule	class	feature-libs/asm/core/asm-core.module.ts:L35-L35	01 (2), 12 (1), 29 (2)
AsmCreateCustomerFacade	class	feature-libs/asm/root/facade/asm-create-customer.facade.ts:L23-L25	29 (2)
AsmCreateCustomerFormComponent	class	feature-libs/asm/components/asm-create-customer-form/asm-create-customer-form.component.ts:L60-L191	29 (1)
AsmCustomer360Adapter	class	feature-libs/asm/customer-360/core/connectors/asm-customer-360.adapter.ts:L13-L20	29 (1)
AsmCustomer360Component	class	feature-libs/asm/customer-360/components/asm-customer-360/asm-customer-360.component.ts:L76-L329	29 (2)
AsmCustomer360Config	class	feature-libs/asm/customer-360/components/config/asm-customer-360-config.ts:L15-L17	29 (1)
AsmCustomer360CoreModule	class	feature-libs/asm/customer-360/core/asm-customer-360-core.module.ts:L16-L16	29 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
AsmCustomer360Module	class	feature-libs/asm/customer-360/asm-customer-360.module.ts:L24-L24	29 (1)
AsmCustomer360Query	interface	feature-libs/asm/customer-360/root/model/asm-customer-360.model.ts:L250-L253	29 (1)
AsmCustomer360RootModule	class	feature-libs/asm/customer-360/root/asm-customer-360-root.module.ts:L36-L36	06 (1), 29 (2)
AsmCustomer360SectionComponent	class	feature-libs/asm/customer-360/components/sections/asm-customer-360-section/asm-customer-360-section.component.ts:L35-L75	29 (1)
AsmCustomer360SectionContext	class	feature-libs/asm/customer-360/components/sections/asm-customer-360-section-context.model.ts:L15-L29	29 (4)
AsmCustomer360SectionContextSource	class	feature-libs/asm/customer-360/components/sections/asm-customer-360-section-context-source.model.ts:L17-L33	29 (2)
AsmCustomer360Service	class	feature-libs/asm/customer-360/core/services/asm-customer-360.service.ts:L22-L74	29 (3)
AsmCustomer360TabsConfig	class	feature-libs/asm/customer-360/root/model/asm-customer-360-tabs-config.ts:L9-L13	29 (1)
AsmCustomer360Type	enum	feature-libs/asm/customer-360/root/model/asm-customer-360.model.ts:L77-L90	29 (1)
AsmCustomerListFacade	class	feature-libs/asm/root/facade/asm-customer-list.facade.ts:L31-L39	29 (2)
AsmCustomerListService	class	feature-libs/asm/core/facade/asm-customer-list.service.ts:L23-L84	29 (1)
AsmDeepLinkService	class	feature-libs/asm/root/services/asm-deep-link.service.ts:L15-L80	29 (2)
AsmEnablerService	class	feature-libs/asm/root/services/asm-enabler.service.ts:L21-L89	29 (3)
AsmLoaderModule	class	feature-libs/asm/root/asm-loader.module.ts:L27-L27	29 (3)
AsmMainUiComponent	class	feature-libs/asm/components/asm-main-ui/asm-main-ui.component.ts:L97-L460	29 (8)
AsmModule	class	feature-libs/asm/asm.module.ts:L15-L15	29 (4)
AsmOccModule	class	feature-libs/asm/occ/asm-occ.module.ts:L24-L24	29 (1)
AsmProtectedRoutesService	class	feature-libs/asm/root/services/asm-protected-routes.service.ts:L24-L44	29 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
AsmRootModule	class	feature-libs/asm/root/asm-root.module.ts:L95-L95	06 (2), 29 (5), 50 (2)
AsmService	class	feature-libs/asm/core/facade/asm.service.ts:L23-L79	29 (2)
AsmSessionCreationAction	class	feature-libs/asm/core/store/actions/asm-session-actions.ts:L16-L21	29 (1)
AsmSessionEffects	class	feature-libs/asm/core/store/effects/asm-session.effect.ts:L15-L41	29 (2)
AsmSessionTimerComponent	class	feature-libs/asm/components/asm-session-timer/asm-session-timer.component.ts:L20-L104	29 (1)
AsmStatePersistenceService	class	feature-libs/asm/core/services/asm-state-persistence.service.ts:L31-L110	29 (1)
AsmStoreModule	class	feature-libs/asm/core/store/asm-store.module.ts:L25-L25	29 (2)
AsmUiUpdate	class	feature-libs/asm/core/store/actions/asm-ui.action.ts:L12-L15	29 (1)
AssignBudget	class	feature-libs/organization/administration/core/store/actions/cost-center.action.ts:L242-L254	28 (1)
Attribute	interface	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L13-L54 (+2)	30 (1)
AttributesDirective	class	integration-libs/cds/src/merchandising/cms-components/directives/attributes/attributes.directive.ts:L16-L48	34 (1)
AuthConfig	class	core-libs/core/src/auth/user-auth/config/auth-config.ts:L36-L143	22 (4)
AuthConfigInitializer	class	core-libs/core/src/auth/user-auth/config/auth-config-initializer.ts:L19-L167	10 (3), 11 (1), 22 (4)
AuthConfigService	class	core-libs/core/src/auth/user-auth/services/auth-config.service.ts:L20-L165	22 (4)
AuthFlowRoutesService	class	core-libs/core/src/auth/user-auth/services/auth-flow-routes.service.ts:L12-L47	05 (1), 14 (1)
AuthGuard	class	core-libs/core/src/auth/user-auth/guards/auth.guard.ts:L22-L43	05 (1), 14 (3), 22 (5), 23 (1), 25 (1), 27 (2), 31 (4), 34 (1), 36 (1)
AuthHttpHeaderService	class	core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L90-L397	06 (2), 22 (6), 29 (4), 36 (10), 50 (10)
AuthInterceptor	class	core-libs/core/src/auth/user-auth/http-interceptors/auth.interceptor.ts:L25-L111	06 (10), 22 (12), 29 (3), 41 (1), 50 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
AuthLibConfig	type	core-libs/core/src/auth/user-auth/config/auth-config.ts:L20-L30	22 (3)
AuthModule	class	core-libs/core/src/auth/auth.module.ts:L18-L24	06 (5), 22 (2)
AuthNotificationService	class	core-libs/core/src/auth/user-auth/facade/auth-notification.service.ts:L27-L38	22 (2), 46 (1)
AuthNotificationType	enum	core-libs/core/src/auth/user-auth/models/auth-notification.model.ts:L7-L9	46 (1)
AuthRedirectService	class	core-libs/core/src/auth/user-auth/services/auth-redirect.service.ts:L27-L162	22 (3)
AuthService	class	core-libs/core/src/auth/user-auth/facade/auth.service.ts:L47-L336	19 (1), 22 (3), 29 (6), 50 (1)
AuthStatePersistenceService	class	core-libs/core/src/auth/user-auth/services/auth-state-persistence.service.ts:L31-L126	07 (1), 19 (1), 22 (2)
AuthStorageService	class	core-libs/core/src/auth/user-auth/services/auth-storage.service.ts:L18-L119	22 (5), 29 (4), 34 (2), 36 (1), 50 (1)
AuthToken	interface	core-libs/core/src/auth/user-auth/models/auth-token.model.ts:L10-L35	22 (2), 46 (1), 54 (1)
AutoFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L72-L96	18 (1)
AutoFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/autofocus/auto-focus.service.ts:L18-L66	18 (3), 54 (1)
AVOID_STACKED_OUTLETS	const	core-libs/storefront/cms-structure/outlet/outlet.model.ts:L15-L15	17 (2)

B

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
B2BApprovalProcess	interface	core-libs/core/src/model/org-unit.model.ts:L41-L44 (+1)	46 (1)
B2BCheckoutDeliveryAddressComponent	class	feature-libs/checkout/b2b/components/checkout-delivery-address/checkout-delivery-address.component.ts:L62-L198	26 (1)
B2BCheckoutReviewSubmitComponent	class	feature-libs/checkout/b2b/components/checkout-review-submit/checkout-review-submit.component.ts:L69-L169	36 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
B2BUnit	interface	core-libs/core/src/model/org-unit.model.ts:L29-L34 (+1)	28 (1), 46 (6), 54 (1)
B2BUser	interface	core-libs/core/src/model/org-unit.model.ts:L36-L39 (+1)	28 (1), 46 (3), 54 (1)
B2BUserRight	enum	core-libs/core/src/model/org-unit.model.ts:L25-L27	46 (1)
B2BUserRole	enum	core-libs/core/src/model/org-unit.model.ts:L18-L23	46 (2), 54 (1)
B2BUserService	class	feature-libs/organization/administration/core/services/b2b-user.service.ts:L36-L417	34 (2)
BackendConfig	interface	core-libs/core/src/occ/config/occ-config.ts:L13-L40	06 (2), 21 (7), 35 (1), 52 (1), 54 (1)
BackendHttpTimeoutConfig	class	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L35-L37	21 (2)
BadCartRequestHandler	class	feature-libs/cart/base/core/http-interceptors/handlers/bad-cart-request.handler.ts:L21-L75	12 (2), 25 (4), 41 (4), 50 (1)
BadCostCenterRequestHandler	class	feature-libs/checkout/b2b/core/http-interceptors/bad-request/bad-cost-center-request.handler.ts:L22-L71	26 (3)
BadGatewayHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/bad-gateway/bad-gateway.handler.ts:L16-L29	41 (1)
BadRequestHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/bad-request/bad-request.handler.ts:L22-L164	30 (1), 41 (6)
BadVoucherRequestHandler	class	feature-libs/cart/base/core/http-interceptors/handlers/bad-voucher-request.handler.ts:L25-L77	25 (4)
BannerCarouselComponent	class	core-libs/storefront/cms-components/content/banner-carousel/banner-carousel.component.ts:L29-L62	45 (1)
BannerCarouselModule	class	core-libs/storefront/cms-components/content/banner-carousel/banner-carousel.module.ts:L33-L33	45 (1)
BannerComponent	class	core-libs/storefront/cms-components/content/banner/banner.component.ts:L44-L190	45 (5)
BannerModule	class	core-libs/storefront/cms-components/content/banner/banner.module.ts:L47-L47	45 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
BASE_SITE_CHANGE	const	core-libs/core/src/site-context/store/actions/base-site.action.ts:L20-L20	13 (1)
BASE_SITE_CONTEXT_ID	const	core-libs/core/src/site-context/providers/context-ids.ts:L9-L9	13 (1), 19 (1), 22 (1)
BASE_SITE_NORMALIZER	const	core-libs/core/src/site-context/connectors/converters.ts:L36-L38	13 (3), 21 (2)
BaseCoreModule	class	core-libs/core/src/base-core.module.ts:L47-L53	01 (1), 02 (11), 05 (1), 06 (3), 07 (1), 10 (7), 11 (3), 12 (2), 13 (1), 15 (1), 17 (1), 19 (1), 21 (3), 22 (1), 37 (1), 38 (1), 40 (1), 41 (4), 42 (1), 47 (2), 50 (1), 54 (1)
BaseFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/base/base-focus.service.ts:L12-L12	18 (1)
BaseItem	type	feature-libs/organization/administration/components/shared/organization.model.ts:L37-L50	28 (1)
BaseMessageComponent	class	feature-libs/organization/administration/components/shared/message/base-message.component.ts:L21-L76	28 (1)
BaseOccModule	class	core-libs/core/src/occ/base-occ.module.ts:L20-L35	06 (1), 13 (1), 15 (1), 21 (2)
BaseOccUrlProperties	interface	core-libs/core/src/occ/services/occ-endpoints.service.ts:L19-L23	21 (1)
BasePageMetaResolver	class	core-libs/core/src/cms/page/base-page-meta.resolver.ts:L28-L108	40 (11), 52 (5)
BaseSite	interface	core-libs/core/src/model/misc.model.ts:L116-L133 (+1)	13 (2), 33 (1), 46 (6)
BaseSiteChange	class	core-libs/core/src/site-context/store/actions/base-site.action.ts:L66-L68	13 (1)
BaseSiteInitializer	class	core-libs/core/src/site-context/services/base-site-initializer.ts:L18-L66	10 (4), 13 (1)
BaseSiteNormalizer	class	core-libs/core/src/occ/adapters/site-context/converters/base-site-normalizer.ts:L13-L29	13 (3), 21 (2)
BaseSiteService	class	core-libs/core/src/site-context/facade/base-site.service.ts:L22-L109	13 (6), 21 (2), 46 (1)
BaseSiteState	interface	core-libs/core/src/site-context/store/state.ts:L37-L41	13 (1)
BaseStore	interface	core-libs/core/src/model/misc.model.ts:L108-L114 (+1)	36 (2), 46 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
BaseStorefrontModule	class	core-libs/storefront/base-storefront.module.ts:L49-L49	02 (14), 06 (3), 10 (8), 14 (1), 16 (1), 17 (2), 18 (1), 20 (1), 38 (2), 40 (2), 41 (2), 43 (1), 45 (2), 47 (4), 51 (1)
BaseTmsModule	class	feature-libs/tracking/tms/core/base-tms.module.ts:L19-L33	33 (4)
BEFORE_CMS_PAGE_GUARD	const	core-libs/storefront/cms-structure/guards/before-cms-page-guard.token.ts:L22-L24	01 (1), 02 (1), 05 (7), 10 (1), 14 (7), 22 (1), 35 (5), 36 (3)
BeforeCmsPageGuardService	class	core-libs/storefront/cms-structure/guards/before-cms-page-guard.service.ts:L18-L33	05 (3), 14 (2)
BlobErrorInterceptor	class	feature-libs/organization/account-summary/root/http-interceptors/blob-error.interceptor.ts:L23-L57	06 (3), 28 (3)
BlockFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L27-L29	18 (1)
BrandPageVisitedEvent	class	integration-libs/cds/src/profiletag/model/profile-tag.model.ts:L112-L118	34 (1)
Breadcrumb	interface	core-libs/core/src/model/product-search.model.ts:L26-L33 (+1)	46 (1)
BreadcrumbComponent	class	core-libs/storefront/cms-components/navigation/breadcrumb/breadcrumb.component.ts:L32-L67	45 (1), 51 (1)
BreadcrumbMeta	interface	core-libs/core/src/cms/model/page.model.ts:L80-L83	15 (1), 40 (1), 52 (1)
BreadcrumbSchemaBuilder	class	core-libs/storefront/cms-structure/seo/structured-data/builders/breadcrumb/breadcrumb-schema.builder.ts:L16-L57	40 (2)
BREAKPOINT	enum	core-libs/storefront/layout/config/layout-config.ts:L22-L28	16 (1), 18 (6), 54 (2)
BreakPoint	interface	core-libs/storefront/layout/config/layout-config.ts:L63-L73	18 (2)
BreakpointService	class	core-libs/storefront/layout/breakpoint/breakpoint.service.ts:L37-L207	16 (1), 18 (6), 38 (2), 46 (1), 51 (1)
BtnLikeLinkDirective	class	core-libs/storefront/layout/a11y/btn-like-link/btn-like-link.directive.ts:L31-L36	18 (1)
BUDGET_ENTITIES	const	feature-libs/organization/administration/core/store/organization-state.ts:L25-L25	28 (1)
BUDGET_NORMALIZER	const	feature-libs/organization/administration/core/connectors/budget/converters.ts:L11-L13	28 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
BUDGET_SERIALIZER	const	feature-libs/organization/administration/core/connectors/budget/converters.ts:L18-L20	28 (2)
BudgetAdapter	class	feature-libs/organization/administration/core/connectors/budget/budget.adapter.ts:L11-L34	28 (2)
BudgetConnector	class	feature-libs/organization/administration/core/connectors/budget/budget.connector.ts:L16-L41	28 (1)
BudgetDetailsComponent	class	feature-libs/organization/administration/components/budget/details/budget-details.component.ts:L48-L60	28 (2)
BudgetFormComponent	class	feature-libs/organization/administration/components/budget/form/budget-form.component.ts:L66-L103	28 (3)
BudgetItemService	class	feature-libs/organization/administration/components/budget/services/budget-item.service.ts:L22-L63	28 (2)
BudgetListService	class	feature-libs/organization/administration/components/budget/services/budget-list.service.ts:L30-L69	28 (2)
BudgetRoutePageMetaResolver	class	feature-libs/organization/administration/components/budget/services/budget-route-page-meta.resolver.ts:L17-L28	28 (1)
BudgetService	class	feature-libs/organization/administration/core/services/budget.service.ts:L29-L150	28 (2)
BulkPricingService	class	feature-libs/product/bulk-pricing/core/services/bulk-pricing.service.ts:L16-L81	24 (2)

C

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CacheEntrySizeCalculator	interface	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.model.ts:L26-L28	39 (3)
CanActivate	type	core-libs/storefront/cms-structure/services/guards-composer.ts:L19-L19	05 (1), 16 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CanActivateObservable	type	core-libs/storefront/cms-structure/services/guards-composer.ts:L24-L24	16 (1)
CANCEL_REPLENISHMENT_ORDER_SUCCESS	const	feature-libs/order/core/store/actions/replenishment-order-details.action.ts:L24-L25	27 (1)
CancelOrder	class	feature-libs/order/core/store/actions/order-details.action.ts:L60-L71	27 (1)
CancelServiceOrderComponent	class	integration-libs/s4-service/order/components/cancel-service-order/cancel-service-order.component.ts:L46-L115	36 (1)
CancelServiceOrderFacade	class	integration-libs/s4-service/root/facade/cancel-service-order.facade.ts:L15-L30	36 (3)
CancelServiceOrderGuard	class	integration-libs/s4-service/order/components/guards/cancel-service-order.guard.ts:L16-L38	36 (1)
CanonicalPageResolver	interface	core-libs/core/src/cms/page/page.resolvers.ts:L92-L101	40 (1)
CanonicalUrlOptions	interface	core-libs/core/src/cms/page/config/page-meta.config.ts:L43-L71	15 (1), 40 (3)
CaptchaComponent	class	core-libs/storefront/shared/components/captcha/captcha.component.ts:L29-L75	45 (1)
CaptchaConfig	interface	core-libs/core/src/model/misc.model.ts:L135-L138	46 (1)
CaptchaModule	class	core-libs/storefront/shared/components/captcha/captcha.module.ts:L25-L25	23 (1), 45 (1)
Card	interface	core-libs/storefront/shared/components/card/card.component.ts:L36-L49	27 (1)
CardComponent	class	core-libs/storefront/shared/components/card/card.component.ts:L69-L193 (+1)	28 (3)
CardType	interface	core-libs/core/src/model/payment.model.ts:L9-L12 (+1)	46 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CarouselComponent	class	core-libs/storefront/shared/components/carousel/carousel.component.ts:L78-L300	07 (1)
CarouselService	class	core-libs/storefront/shared/components/carousel/carousel.service.ts:L15-L61	45 (1)
Cart	interface	core-libs/core/src/occ/occ-models/occ.models.ts:L1040-L1112 (+1)	25 (4), 32 (3), 52 (5), 54 (1)
CART_ACCESS_CODE_NORMALIZER	const	feature-libs/cart/base/core/connectors/access-code/converters.ts:L10-L12	25 (1)
CART_ADD_ENTRY_SUCCESS	const	feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L12-L12	25 (2), 53 (1)
CART_ADD_VOUCHER	const	feature-libs/cart/base/core/store/actions/cart-voucher.action.ts:L10-L10	25 (1)
CART_ADD_VOUCHER_SUCCESS	const	feature-libs/cart/base/core/store/actions/cart-voucher.action.ts:L12-L13	25 (1)
CART_BASE_CORE_FEATURE	const	feature-libs/cart/base/root/feature-name.ts:L8-L8	20 (1)
CART_BASE_FEATURE	const	feature-libs/cart/base/root/feature-name.ts:L7-L7	12 (1), 26 (1), 27 (1), 32 (2), 47 (2), 50 (1)
CART_BASE_MODULE	const	core-libs/schematics/src/shared/lib-configs/cart-schematics-config.ts:L42-L42	36 (1)
CART_GUEST_USER_NORMALIZER	const	feature-libs/cart/base/core/connectors/guest-user/converters.ts:L11-L13	25 (1)
CART_MODIFICATION_NORMALIZER	const	feature-libs/cart/base/root/tokens/converters.ts:L25-L27	25 (2), 30 (1)
CART_NORMALIZER	const	feature-libs/cart/base/root/tokens/converters.ts:L17-L19	25 (4), 35 (2), 52 (10)
CART_VALIDATION_NORMALIZER	const	feature-libs/cart/base/core/connectors/validation/converters.ts:L11-L13	25 (3)
CART_VOUCHER_NORMALIZER	const	feature-libs/cart/base/root/tokens/converters.ts:L33-L35	25 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CART_WISH_LIST_FEATURE	const	feature-libs/cart/wish-list/root/feature-name.ts:L7-L7	47 (1)
CartAccessCodeAdapter	class	feature-libs/cart/base/core/connectors/access-code/cart-access-code.adapter.ts:L9-L21	25 (1)
CartAccessCodeConnector	class	feature-libs/cart/base/core/connectors/access-code/cart-access-code.connector.ts:L12-L21	25 (1)
CartAccessCodeFacade	class	feature-libs/cart/base/root/facade/cart-access-code.facade.ts:L21-L29	25 (3)
CartAccessCodeService	class	feature-libs/cart/base/core/facade/cart-access-code.service.ts:L14-L35	25 (2)
CartAdapter	class	feature-libs/cart/base/core/connectors/cart/cart.adapter.ts:L10-L69	25 (3), 52 (7)
CartAddEntry	class	feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L22-L36	25 (2)
CartAddEntryFail	class	feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L59-L79	30 (1)
CartAddEntryFailEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L78-L86	25 (2)
CartAddEntrySuccess	class	feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L38-L57	25 (1), 30 (2)
CartAddEntrySuccessEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L66-L76	25 (1), 31 (1), 50 (1), 53 (5)
CartAddVoucher	class	feature-libs/cart/base/core/store/actions/cart-voucher.action.ts:L23-L31	25 (4)
CartAddVoucherFail	class	feature-libs/cart/base/core/store/actions/cart-voucher.action.ts:L33-L49	25 (1)
CartBaseComponentsModule	class	feature-libs/cart/base/components/cart-base-components.module.ts:L53-L53	16 (1), 17 (2), 25 (2), 52 (1), 53 (2)
CartBaseCoreModule	class	feature-libs/cart/base/core/cart-base-core.module.ts:L49-L49	12 (1), 25 (4), 41 (1), 50 (2), 52 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CartBaseEventModule	class	feature-libs/cart/base/root/events/cart-base-event.module.ts:L11-L15	25 (1)
CartBaseModule	class	feature-libs/cart/base/cart-base.module.ts:L15-L15	17 (1), 25 (2), 32 (1), 36 (3), 47 (2), 52 (3), 53 (3)
CartBaseOccModule	class	feature-libs/cart/base/occ/cart-base-occ.module.ts:L72-L72	25 (2), 32 (1), 52 (5)
CartBaseRootModule	class	feature-libs/cart/base/root/cart-base-root.module.ts:L83-L83	12 (2), 25 (1), 47 (2)
CartConfig	class	feature-libs/cart/base/root/config/cart-config.ts:L16-L28	25 (3), 35 (1), 50 (1)
CartConfigService	class	feature-libs/cart/base/core/services/cart-config.service.ts:L13-L23	25 (1)
CartConnector	class	feature-libs/cart/base/core/connectors/cart/cart.connector.ts:L15-L54	25 (2), 52 (2)
CartEffects	class	feature-libs/cart/base/core/store/effects/cart.effect.ts:L43-L472	25 (3)
CartEntryAdapter	class	feature-libs/cart/base/core/connectors/entry/cart-entry.adapter.ts:L10-L57	25 (1)
CartEntryEffects	class	feature-libs/cart/base/core/store/effects/cart-entry.effect.ts:L24-L165	25 (4)
CartEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L14-L33	20 (2), 25 (2), 53 (4)
CartEventBuilder	class	feature-libs/cart/base/core/event/cart-event.builder.ts:L47-L250	20 (1), 25 (2), 50 (1), 53 (2)
CartEventModule	class	feature-libs/cart/base/core/event/cart-event.module.ts:L11-L15	25 (2), 50 (1)
CartGuestUserAdapter	class	feature-libs/cart/base/core/connectors/guest-user/cart-guest-user.adapter.ts:L10-L38	25 (1)
CartGuestUserConnector	class	feature-libs/cart/base/core/connectors/guest-user/cart-guest-user.connector.ts:L13-L31	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CartGuestUserFacade	class	feature-libs/cart/base/root/facade/cart-guest-user.facade.ts:L22-L40	25 (3)
CartGuestUserService	class	feature-libs/cart/base/core/facade/cart-guest-user.service.ts:L14-L72	25 (2)
CartItemContext	class	feature-libs/cart/base/root/models/cart-item-context.model.ts:L17-L29	25 (4), 36 (1), 53 (5)
CartItemListComponentService	class	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.service.ts:L12-L16	36 (7)
CartItemListRowComponent	class	feature-libs/cart/base/components/cart-shared/cart-item-list-row/cart-item-list-row.component.ts:L46-L53	36 (1)
CartItemQuantityService	class	feature-libs/cart/base/root/services/cart-item-quantity.service.ts:L20-L43	25 (1)
CartItemValidationService	class	feature-libs/cart/base/core/services/cart-item-validation.service.ts:L21-L102	25 (1)
CartModification	interface	core-libs/core/src/occ/occ-models/occ.models.ts:L1135-L1147 (+1)	25 (1)
CartNotEmptyGuard	class	feature-libs/checkout/base/components/guards/cart-not-empty.guard.ts:L17-L40	26 (6), 53 (2)
CartOccEndpoints	interface	feature-libs/cart/base/occ/model/occ-cart-endpoints.model.ts:L9-L95	25 (2), 52 (2)
CartOutlets	enum	feature-libs/cart/base/root/models/cart-outlets.model.ts:L10-L31	17 (2), 25 (2), 51 (1), 53 (4)
CartPageEvent	class	feature-libs/cart/base/root/events/cart-page.events.ts:L12-L15	25 (3), 50 (1)
CartPageEventBuilder	class	feature-libs/cart/base/core/event/cart-page-event.builder.ts:L17-L36	25 (1), 50 (1)
CartPageEventModule	class	feature-libs/cart/base/core/event/cart-page-event.module.ts:L11-L15	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CartPageLayoutHandler	class	feature-libs/cart/base/components/cart-page-layout-handler.ts:L21-L68	16 (1), 25 (3), 51 (1)
CartPersistenceModule	class	feature-libs/cart/base/core/cart-persistence.module.ts:L74-L74	12 (2), 19 (1), 25 (2)
CartPickupOptionsContainerComponent	class	feature-libs/pickup-in-store/components/container/cart-pickup-options-container/cart-pickup-options-container.component.ts:L83-L311	32 (1)
CartProceedToCheckoutComponent	class	feature-libs/cart/base/components/cart-proceed-to-checkout/cart-proceed-to-checkout.component.ts:L55-L121	25 (1)
CartProcessesDecrement	class	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L37-L42	25 (2)
CartQuickOrderFormComponent	class	feature-libs/cart/quick-order/components/cart-quick-order-form/cart-quick-order-form.component.ts:L57-L180	25 (1)
CartRemoveEntry	class	feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L81-L89	25 (1)
CartRemoveEntrySuccessEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L96-L102	31 (1)
CartRemoveVoucher	class	feature-libs/cart/base/core/store/actions/cart-voucher.action.ts:L73-L81	25 (1)
CartType	enum	feature-libs/cart/base/root/models/cart.model.ts:L197-L202	25 (2)
CartUiEventAddToCart	class	feature-libs/cart/base/root/events/cart.events.ts:L122-L136	25 (4), 53 (9)
CartUpdateEntry	class	feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L121-L136	25 (1)
CartUpdateEntrySuccess	class	feature-libs/cart/base/core/store/actions/cart-entry.action.ts:L138-L153	30 (1)
CartUpdateEntrySuccessEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L104-L111	31 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CartValidationAdapter	class	feature-libs/cart/base/core/connectors/validation/cart-validation.adapter.ts:L10-L15	25 (1)
CartValidationFacade	class	feature-libs/cart/base/root/facade/cart-validation.facade.ts:L22-L32	25 (3)
CartValidationGuard	class	feature-libs/cart/base/core/guards/cart-validation.guard.ts:L29-L103	25 (3), 26 (1)
CartValidationService	class	feature-libs/cart/base/core/facade/cart-validation.service.ts:L26-L67	25 (2)
CartValidationStateService	class	feature-libs/cart/base/core/services/cart-validation-state.service.ts:L16-L80	25 (3)
CartValidationStatusCode	enum	feature-libs/cart/base/root/models/cart.model.ts:L208-L216	25 (2)
CartVoucherAdapter	class	feature-libs/cart/base/core/connectors/voucher/cart-voucher.adapter.ts:L9-L35	25 (1)
CartVoucherEffects	class	feature-libs/cart/base/core/store/effects/cart-voucher.effect.ts:L21-L121	25 (3)
CartVoucherFacade	class	feature-libs/cart/base/root/facade/cart-voucher.facade.ts:L29-L53	25 (3)
CartVoucherService	class	feature-libs/cart/base/core/facade/cart-voucher.service.ts:L21-L99	25 (2)
Catalog	interface	core-libs/core/src/model/unused.model.ts:L25-L31 (+1)	46 (1)
CategoryNavigationComponent	class	core-libs/storefront/cms-components/navigation/category-navigation/category-navigation.component.ts:L22-L33	45 (4)
CategoryPageMetaResolver	class	core-libs/core/src/product/services/category-page-meta.resolver.ts:L36-L143	24 (1), 40 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CategoryPageResultsEvent	class	core-libs/storefront/events/product/product-page.events.ts:L25-L31	20 (3)
CDC_FEATURE	const	integration-libs/cdc/root/feature-name.ts:L7-L7	34 (2)
CDC_PREFERENCE_SERIALIZER	const	integration-libs/cdc/root/consent-management/converters/converter.ts:L11-L13	34 (4), 52 (1)
CdcAdministrationModule	class	integration-libs/cdc/organization/administration/cdc-administration.module.ts:L62-L62	28 (1), 34 (8)
CdcAuthFacade	class	integration-libs/cdc/root/facade/cdc-auth.facade.ts:L21-L33	34 (4)
CdcAuthModule	class	integration-libs/cdc/core/auth/cdc-auth.module.ts:L16-L16	34 (1)
CdcAuthService	class	integration-libs/cdc/core/auth/facade/cdc-auth.service.ts:L29-L146	34 (3)
CDCB2BRegisterComponentService	class	integration-libs/cdc/organization/user-registration/cdc-b2b-register-component.service.ts:L32-L138	34 (1)
CDCB2BRegisterModule	class	integration-libs/cdc/organization/user-registration/cdc-b2b-register.module.ts:L40-L40	34 (3)
CdcComponentsModule	class	integration-libs/cdc/components/cdc-components.module.ts:L15-L15	34 (2)
CdcConfig	class	integration-libs/cdc/root/config/cdc-config.ts:L14-L20	34 (2)
CdcConsentManagementComponentService	class	integration-libs/cdc/root/consent-management/services/cdc-consent-management-component.service.ts:L16-L53	34 (1)
CdcConsentManagementModule	class	integration-libs/cdc/root/consent-management/cdc-consent.module.ts:L31-L31	34 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CdcConsentsLocalStorageService	class	integration-libs/cdc/root/consent-management/services/cdc-consents-local-storage.service.ts:L20-L82	34 (1)
CdcCoreModule	class	integration-libs/cdc/core/cdc-core.module.ts:L17-L17	34 (5)
CdcEventBuilder	class	integration-libs/cdc/core/events/cdc-event.builder.ts:L15-L39	34 (1)
CdcEventModule	class	integration-libs/cdc/core/events/cdc-event.module.ts:L11-L15	34 (1)
CDCForgotPasswordModule	class	integration-libs/cdc/user-profile/forgot-password/cdc-forgot-password.module.ts:L50-L50	34 (1)
CdcJsService	class	integration-libs/cdc/root/service/cdc-js.service.ts:L48-L860	34 (6)
CdcLoadUserTokenFailEvent	class	integration-libs/cdc/root/events/cdc-event.ts:L13-L18	34 (4)
CdcLoginAsGuestGuard	class	integration-libs/cdc/root/guards/cdc-login-as-guest.guard.ts:L13-L17	34 (3)
CdcLoginFormComponentService	class	integration-libs/cdc/user-account/login-form/cdc-login-form-component.service.ts:L19-L75	34 (2)
CDCLoginFormModule	class	integration-libs/cdc/user-account/login-form/cdc-login-form.module.ts:L71-L71	34 (3)
CdcLoginGuard	class	integration-libs/cdc/root/guards/cdc-login.guard.ts:L20-L32	34 (3)
CdcLogoutGuard	class	integration-libs/cdc/root/guards/cdc-logout.guard.ts:L26-L56	34 (3)
CdcModule	class	integration-libs/cdc/cdc.module.ts:L14-L14	34 (6)
CdcPreferenceSerializer	class	integration-libs/cdc/root/consent-management/converters/cdc-preference.serializer.ts:L12-L39	34 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CdcReconsentDialogEventListener	class	integration-libs/cdc/user-account/login-form/reconsent/cdc-reconsent-dialogue-event.listener.ts:L16-L56	34 (1)
CdcReConsentEvent	class	integration-libs/cdc/root/events/cdc-event.ts:L20-L31	34 (3)
CdcReconsentModule	class	integration-libs/cdc/user-account/login-form/reconsent/cdc-reconsent.module.ts:L35-L41	34 (1)
CDCRegisterModule	class	integration-libs/cdc/user-profile/register/cdc-register.module.ts:L53-L53	34 (2)
CdcRootModule	class	integration-libs/cdc/root/cdc-root.module.ts:L69-L69	34 (7)
CdcStoreModule	class	integration-libs/cdc/core/store/cdc-store.module.ts:L18-L18	34 (2)
CDCUpdateEmailModule	class	integration-libs/cdc/user-profile/update-email/cdc-update-email.module.ts:L52-L52	34 (1)
CDCUpdatePasswordModule	class	integration-libs/cdc/user-profile/update-password/cdc-update-password.module.ts:L52-L52	34 (1)
CDCUpdateProfileModule	class	integration-libs/cdc/user-profile/update-profile/cdc-update-profile.module.ts:L54-L54	34 (1)
CDCUserAccountModule	class	integration-libs/cdc/user-account/cdc-user-account.module.ts:L13-L13	34 (2)
CdcUserAddressesEffects	class	integration-libs/cdc/core/store/effects/cdc-user-addresses.effect.ts:L24-L121	34 (2)
CdcUserConsentAdapter	class	integration-libs/cdc/root/consent-management/cdc-user-consent.adapter.ts:L21-L90	34 (1), 52 (1)
CDCUserProfileModule	class	integration-libs/cdc/user-profile/cdc-user-profile.module.ts:L23-L23	34 (1)
CdcUserTokenEffects	class	integration-libs/cdc/core/store/effects/cdc-user-token.effect.ts:L22-L67	34 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CdpCustomerTicketingModule	class	integration-libs/cdp/customer-ticketing/cdp-customer-ticketing.module.ts:L35-L35	31 (1), 34 (7), 47 (1)
CdsBackendNotificationAdapter	class	integration-libs/cds/src/profiletag/adapters/cds-backend-notification-adapter.ts:L9-L11	34 (1)
CdsConfig	class	integration-libs/cds/src/config/cds-config.ts:L17-L27	34 (2)
CdsEndpoints	interface	integration-libs/cds/src/cds-models/cds-endpoints.model.ts:L7-L15	34 (1)
CdsEndpointsService	class	integration-libs/cds/src/services/cds-endpoints.service.ts:L15-L90	34 (1)
CdsModule	class	integration-libs/cds/src/cds.module.ts:L35-L47	34 (2)
CellComponent	class	feature-libs/organization/administration/components/shared/table/cell.component.ts:L31-L111	28 (3)
CHANGE_NEXT_PAGE_CONTEXT	const	core-libs/core/src/routing/store/actions/router.action.ts:L10-L10	14 (1)
ChangeGroup	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L349-L365	30 (2)
ChangeNextPageContext	class	core-libs/core/src/routing/store/actions/router.action.ts:L12-L15	05 (1), 14 (2)
CheckConflictDialoge	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L504-L510	30 (1)
checkOAuthParamsInUrl()	function	core-libs/core/src/auth/user-auth/user-auth.module.ts:L39-L56	29 (2)
CHECKOUT_B2B_CMS_COMPONENTS	const	feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L23-L31	36 (2)
CHECKOUT_B2B_MODULE	const	core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L77-L77	36 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CHECKOUT_B2B_SCHEMATICS_CONFIG	const	core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L83-L125	43 (1)
CHECKOUT_BASE_CMS_COMPONENTS	const	feature-libs/checkout/base/root/checkout-root.module.ts:L20-L35	26 (2), 36 (1), 53 (3)
CHECKOUT_BASE_SCHEMATICS_CONFIG	const	core-libs/schematics/src/shared/lib-configs/checkout-schematics-config.ts:L40-L75	43 (1)
CHECKOUT_CORE_FEATURE	const	feature-libs/checkout/base/root/feature-name.ts:L8-L8	35 (1), 36 (2)
CHECKOUT_FEATURE	const	feature-libs/checkout/base/root/feature-name.ts:L7-L7	36 (1), 43 (1), 53 (2)
CHECKOUT_NORMALIZER	const	feature-libs/checkout/base/core/connectors/checkout/converters.ts:L11-L13	26 (1)
CheckoutAdapter	class	feature-libs/checkout/base/core/connectors/checkout/checkout.adapter.ts:L10-L21	26 (1)
CheckoutAuthGuard	class	feature-libs/checkout/base/components/guards/checkout-auth.guard.ts:L25-L78	26 (7), 29 (1), 35 (5), 53 (2)
CheckoutB2BAuthGuard	class	feature-libs/checkout/b2b/components/guards/checkout-b2b-auth.guard.ts:L30-L89	26 (2)
CheckoutB2BComponentsModule	class	feature-libs/checkout/b2b/components/checkout-b2b-components.module.ts:L39-L39	26 (1)
CheckoutB2BModule	class	feature-libs/checkout/b2b/checkout-b2b.module.ts:L19-L19	26 (1), 36 (2), 43 (2), 47 (1)
CheckoutB2BRootModule	class	feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L52-L52	26 (3), 47 (1)
CheckoutB2BStepsSetGuard	class	feature-libs/checkout/b2b/components/guards/checkout-b2b-steps-set.guard.ts:L32-L178	26 (2), 35 (4), 36 (1), 53 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CheckoutBillingAddressAdapter	class	feature-libs/checkout/base/core/connectors/checkout-billing-address/checkout-billing-address.adapter.ts:L10-L23	26 (1)
CheckoutBillingAddressService	class	feature-libs/checkout/base/core/facade/checkout-billing-address.service.ts:L25-L81	26 (1)
CheckoutCartInterceptor	class	feature-libs/checkout/base/root/http-interceptors/checkout-cart.interceptor.ts:L27-L95	06 (3), 26 (3), 54 (1)
CheckoutConfig	class	feature-libs/checkout/base/root/config/checkout-config.ts:L22-L51	26 (2), 36 (11), 53 (2)
CheckoutConfigService	class	feature-libs/checkout/base/components/services/checkout-config.service.ts:L18-L113	26 (1)
CheckoutCoreModule	class	feature-libs/checkout/base/core/checkout-core.module.ts:L33-L33	26 (1), 52 (1)
CheckoutCostCenterComponent	class	feature-libs/checkout/b2b/components/checkout-cost-center/checkout-cost-center.component.ts:L33-L93	26 (3)
CheckoutDeliveryAddressAdapter	class	feature-libs/checkout/base/core/connectors/checkout-delivery-address/checkout-delivery-address.adapter.ts:L10-L47	26 (1)
CheckoutDeliveryAddressClearedEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L86-L91	26 (2)
CheckoutDeliveryAddressCreatedEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L58-L67	20 (1), 26 (3)
CheckoutDeliveryAddressService	class	feature-libs/checkout/base/core/facade/checkout-delivery-address.service.ts:L31-L173	20 (1), 35 (1)
CheckoutDeliveryAddressSetEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L72-L81	26 (4)
CheckoutDeliveryModeClearedErrorEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L125-L130	26 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CheckoutDeliveryModeClearedEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L115-L120	26 (3)
CheckoutDeliveryModeComponent	class	feature-libs/checkout/base/components/checkout-delivery-mode/checkout-delivery-mode.component.ts:L61-L194	36 (1)
CheckoutDeliveryModesAdapter	class	feature-libs/checkout/base/core/connectors/checkout-delivery-modes/checkout-delivery-modes.adapter.ts:L10-L45	26 (1)
CheckoutDeliveryModeSetEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L101-L110	26 (3)
CheckoutDeliveryModesFacade	class	feature-libs/checkout/base/root/facade/checkout-delivery-modes.facade.ts:L30-L55	53 (3)
CheckoutEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L32-L48	26 (1)
CheckoutEventModule	class	feature-libs/checkout/base/root/events/checkout-event.module.ts:L16-L27	26 (3)
CheckoutFlow	interface	feature-libs/checkout/base/root/model/checkout-flow.model.ts:L10-L31	26 (4), 35 (1), 53 (1)
CheckoutFlowOrchestratorService	class	feature-libs/checkout/base/components/services/checkout-flow-orchestrator.service.ts:L16-L45	26 (2), 35 (2)
CheckoutGuard	class	feature-libs/checkout/base/components/guards/checkout.guard.ts:L21-L71	26 (5)
CheckoutLegacyStoreEventListener	class	feature-libs/checkout/base/root/events/checkout-legacy-store-event.listener.ts:L25-L75	23 (1)
CheckoutModule	class	feature-libs/checkout/base/checkout.module.ts:L15-L15	26 (3), 36 (1), 43 (3), 47 (1), 53 (4)
CheckoutOccModule	class	feature-libs/checkout/base/occ/checkout-occ.module.ts:L50-L50	26 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CheckoutOrchestratorComponent	class	feature-libs/checkout/base/components/checkout-orchestrator/checkout-orchestrator.component.ts:L14-L18	26 (1)
CheckoutPageMetaResolver	class	feature-libs/checkout/base/core/services/checkout-page-meta.resolver.ts:L32-L77	26 (2), 40 (1)
CheckoutPaymentAdapter	class	feature-libs/checkout/base/core/connectors/checkout-payment/checkout-payment.adapter.ts:L10-L52	26 (1)
CheckoutPaymentCardTypesQueryReloadEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L188-L193	26 (1)
CheckoutPaymentDetailsCreatedEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L160-L169	26 (3)
CheckoutPaymentDetailsSetEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L174-L183	26 (3)
CheckoutPaymentMethodModule	class	feature-libs/checkout/base/components/checkout-payment-method/checkout-payment-method.module.ts:L45-L45	35 (1)
CheckoutPaymentService	class	feature-libs/checkout/base/core/facade/checkout-payment.service.ts:L37-L185	35 (1)
CheckoutPaymentTypeComponent	class	feature-libs/checkout/b2b/components/checkout-payment-type/checkout-payment-type.component.ts:L48-L214	26 (1)
CheckoutPaymentTypeFacade	class	feature-libs/checkout/b2b/root/facade/checkout-payment-type.facade.ts:L29-L67	35 (7)
CheckoutPaymentTypeSetEvent	class	feature-libs/checkout/b2b/root/events/checkout-b2b.events.ts:L57-L72	35 (1)
CheckoutPlaceOrderEventListener	class	feature-libs/checkout/base/root/events/checkout-place-order-event.listener.ts:L17-L46	27 (1)
CheckoutQueryEventListener	class	feature-libs/checkout/base/root/events/checkout-query-event.listener.ts:L30-L67	20 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CheckoutQueryFacade	class	feature-libs/checkout/base/root/facade/checkout-query.facade.ts:L22-L29	26 (1)
CheckoutQueryReloadEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L12-L17	26 (4)
CheckoutQueryResetEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L22-L27	26 (6), 27 (1), 36 (3)
CheckoutQueryService	class	feature-libs/checkout/base/core/facade/checkout-query.service.ts:L28-L89	20 (2), 26 (1)
CheckoutReplenishmentFormService	class	feature-libs/checkout/scheduled-replenishment/components/services/checkout-replenishment-form.service.ts:L33-L126	26 (2)
CheckoutReviewOverviewComponent	class	feature-libs/checkout/base/components/checkout-review/checkout-review-overview/checkout-review-overview.component.ts:L25-L77	26 (1)
CheckoutReviewSubmitComponent	class	feature-libs/checkout/base/components/checkout-review-submit/checkout-review-submit.component.ts:L73-L217	26 (1), 35 (1)
CheckoutRootModule	class	feature-libs/checkout/base/root/checkout-root.module.ts:L60-L60	06 (1), 26 (4), 43 (1)
CheckoutScheduledReplenishmentModule	class	feature-libs/checkout/scheduled-replenishment/checkout-scheduled-replenishment.module.ts:L13-L13	26 (1), 47 (1)
CheckoutScheduledReplenishmentPlaceOrderComponent	class	feature-libs/checkout/scheduled-replenishment/components/checkout-place-order/checkout-place-order.component.ts:L60-L176	26 (1)
CheckoutScheduledReplenishmentRootModule	class	feature-libs/checkout/scheduled-replenishment/root/checkout-scheduled-replenishment-root.module.ts:L37-L37	26 (2), 47 (1)
CheckoutServiceDetailsAdapter	class	integration-libs/s4-service/checkout/core/connector/checkout-service-details.adapter.ts:L10-L23	36 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CheckoutServiceDetailsComponent	class	integration-libs/s4-service/checkout/components/checkout-service-details/checkout-service-details.component.ts:L57-L169	36 (1)
CheckoutServiceDetailsEventListener	class	integration-libs/s4-service/root/events/checkout-service-details-event.listener.ts:L16-L34	36 (1)
CheckoutServiceDetailsEventModule	class	integration-libs/s4-service/root/events/checkout-service-details-event.module.ts:L11-L17	36 (2)
CheckoutServiceDetailsFacade	class	integration-libs/s4-service/root/facade/checkout-service-details.facade.ts:L28-L57	36 (2)
CheckoutServiceDetailsSetEvent	class	integration-libs/s4-service/root/events/checkout-service-details.events.ts:L18-L28	36 (4)
CheckoutServiceOrderStepsSetGuard	class	integration-libs/s4-service/checkout/components/guards/checkout-service-order-steps-set.guard.ts:L21-L100	36 (3)
CheckoutServiceSchedulePickerService	class	integration-libs/s4-service/root/facade/checkout-service-schedule-picker.service.ts:L16-L113	36 (3)
CheckoutState	interface	feature-libs/checkout/base/root/model/checkout-state.model.ts:L10-L14	26 (4), 36 (2)
CheckoutStep	interface	feature-libs/checkout/base/root/model/checkout-step.model.ts:L20-L27	26 (2), 53 (4)
CheckoutStepService	class	feature-libs/checkout/base/components/services/checkout-step.service.ts:L22-L204	26 (4), 53 (6)
CheckoutStepsSetGuard	class	feature-libs/checkout/base/components/guards/checkout-steps-set.guard.ts:L36-L187	26 (8), 36 (6), 53 (6)
CheckoutStepState	enum	feature-libs/checkout/base/root/model/checkout-step.model.ts:L14-L18	26 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CheckoutStepType	enum	feature-libs/checkout/base/root/model/checkout-step.model.ts:L7-L12	26 (4), 36 (1), 53 (7)
CheckoutSupportedDeliveryModesQueryReloadEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L135-L140	26 (1), 32 (1)
CheckoutSupportedDeliveryModesQueryResetEvent	class	feature-libs/checkout/base/root/events/checkout.events.ts:L145-L150	26 (2)
CLEAR_CART_STATE	const	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L17-L17 (+1)	25 (1)
CLEAR_ORGANIZATION_DATA	const	feature-libs/organization/administration/core/store/actions/organization.action.ts:L9-L9	28 (2)
CLEAR_STOCK_DATA	const	feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L21-L21	32 (1)
CLEAR_STORE_FINDER_DATA	const	feature-libs/storefinder/core/store/actions/view-all-stores.action.ts:L14-L14	32 (2)
ClearCartState	class	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L55-L60	25 (4)
ClearCmsComponent	class	core-libs/core/src/cms/store/actions/components.action.ts:L82-L88	15 (2), 33 (1)
ClearMiscsDataEffect	class	core-libs/core/src/user/store/effects/clear-miscs-data.effect.ts:L16-L30	23 (1)
ClearProductPrice	class	core-libs/core/src/product/store/actions/product.action.ts:L66-L72	24 (1)
ClearProductSearchByCodeState	class	core-libs/core/src/product/store/actions/product-search-by-code.action.ts:L61-L66	24 (1)
ClearSearch	interface	core-libs/core/src/model/product-search.model.ts:L85-L88	46 (2)
ClearStockData	class	feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L67-L73	32 (1)
ClearStoreFinderData	class	feature-libs/storefinder/core/store/actions/view-all-stores.action.ts:L43-L45	32 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ClearUserMiscsData	class	core-libs/core/src/user/store/actions/user-misc.action.ts:L10-L12	23 (2)
ClientAuthenticationTokenService	class	core-libs/core/src/auth/client-auth/services/client-authentication-token.service.ts:L20-L49	22 (1)
ClientAuthModule	class	core-libs/core/src/auth/client-auth/client-auth.module.ts:L22-L29	06 (2), 22 (4)
ClientTokenEffect	class	core-libs/core/src/auth/client-auth/store/effects/client-token.effect.ts:L18-L48	22 (2)
ClientTokenInterceptor	class	core-libs/core/src/auth/client-auth/http-interceptors/client-token.interceptor.ts:L32-L112	06 (12), 21 (1), 22 (10)
CloneSavedCartSuccess	class	feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L301-L313	25 (1)
CMS_COMPONENT_NORMALIZER	const	core-libs/core/src/cms/connectors/component/converters.ts:L11-L13	15 (4)
CMS_FEATURE	const	core-libs/core/src/cms/store/cms-state.ts:L12-L12	15 (3)
CMS_GET_COMPONENT_FROM_PAGE	const	core-libs/core/src/cms/store/actions/components.action.ts:L16-L16 (+1)	15 (1)
CMS_PAGE_NORMALIZER	const	core-libs/core/src/cms/connectors/page/converters.ts:L11-L13	15 (5), 21 (1)
CmsActivatedRouteSnapshot	interface	core-libs/core/src/routing/models/cms-route.ts:L19-L21	14 (1)
CmsBreadcrumbsComponent	interface	core-libs/core/src/model/cms.model.ts:L168-L168	46 (1)
CmsComponent	interface	core-libs/core/src/model/cms.model.ts:L7-L32	15 (1), 46 (4), 51 (2)
CmsComponentAdapter	class	core-libs/core/src/cms/connectors/component/cms-component.adapter.ts:L11-L35	15 (3), 21 (2)
CmsComponentChildRoutesConfig	interface	core-libs/core/src/cms/config/cms-config.ts:L48-L58	15 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CMSComponentConfig	interface	core-libs/core/src/cms/config/cms-config.ts:L103-L107	15 (3), 51 (1)
CmsComponentConnector	class	core-libs/core/src/cms/connectors/component/cms-component.connector.ts:L19-L87	15 (2)
CmsComponentData	class	core-libs/storefront/cms-structure/page/model/cms-component-data.ts:L10-L13	01 (1), 12 (1), 16 (4), 45 (7), 51 (13)
CmsComponentMapping	interface	core-libs/core/src/cms/config/cms-config.ts:L60-L93	15 (3), 28 (1), 45 (1), 51 (2), 53 (2)
CmsComponentsService	class	core-libs/storefront/cms-structure/services/cms-components.service.ts:L39-L277	03 (1), 11 (2), 12 (2), 16 (3), 33 (1), 50 (1), 51 (1), 53 (1)
CmsConfig	class	core-libs/core/src/cms/config/cms-config.ts:L128-L134	12 (4), 15 (4), 23 (3), 26 (1), 27 (3), 50 (1), 51 (5), 53 (5)
CmsFeaturesService	class	core-libs/storefront/cms-structure/services/cms-features.service.ts:L33-L206	03 (2), 11 (1), 12 (9), 16 (4), 24 (3), 33 (2), 47 (3), 53 (1)
CmsGetComponentFromPage	class	core-libs/core/src/cms/store/actions/components.action.ts:L65-L80	15 (4)
CmsGuardsService	class	core-libs/storefront/cms-structure/services/cms-guards.service.ts:L22-L49	05 (1), 12 (1), 16 (3), 26 (3)
CmsI18nService	class	core-libs/storefront/cms-structure/services/cms-i18n.service.ts:L14-L29	16 (1), 37 (2)
CmsInjectorService	class	core-libs/storefront/cms-structure/page/component/services/cms-injector.service.ts:L22-L51	03 (1), 12 (1), 16 (4)
CmsLcpModule	class	core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.module.ts:L12-L19	02 (2), 16 (1)
CmsLcpService	class	core-libs/storefront/cms-structure/cms-lcp-context/cms-lcp.service.ts:L14-L66	16 (4)
CmsModule	class	core-libs/core/src/cms/cms.module.ts:L17-L24	15 (3), 40 (1)
CmsOccModule	class	core-libs/core/src/occ/adapters/cms/cms-occ.module.ts:L34-L34	15 (2), 21 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CmsPageAction	type	core-libs/core/src/cms/store/actions/page.action.ts:L72-L76	15 (1)
CmsPageAdapter	class	core-libs/core/src/cms/connectors/page/cms-page.adapter.ts:L15-L24	15 (4), 21 (3)
CmsPageConfig	interface	core-libs/core/src/cms/config/cms-structure.config.ts:L19-L49	15 (3), 51 (1)
CmsPageConnector	class	core-libs/core/src/cms/connectors/page/cms-page.connector.ts:L19-L71	15 (4)
CmsPageGuard	class	core-libs/storefront/cms-structure/guards/cms-page.guard.ts:L26-L92	05 (16), 10 (2), 12 (2), 14 (14), 15 (3), 16 (2), 23 (1), 24 (1), 25 (2), 27 (1), 28 (1), 29 (1), 30 (1), 31 (3), 36 (1), 51 (8), 53 (1)
CmsPageGuardService	class	core-libs/storefront/cms-structure/guards/cms-page-guard.service.ts:L38-L144	05 (1), 14 (1), 34 (1), 37 (2), 41 (1)
CmsPageNotFoundOutboundHttpError	class	core-libs/core/src/error-handling/http-error-handler/outbound-http-error.ts:L20-L25	06 (2), 39 (3), 41 (7), 47 (1)
CmsPageSlotConfig	interface	core-libs/core/src/cms/config/cms-structure.config.ts:L63-L66	15 (2), 51 (1)
CmsPageSlotsConfig	interface	core-libs/core/src/cms/config/cms-structure.config.ts:L54-L56	15 (3)
CmsParagraphComponent	interface	core-libs/core/src/model/cms.model.ts:L88-L92	51 (2)
CmsParagraphModule	class	core-libs/storefront/cms-components/content/paragraph/paragraph.module.ts:L35-L35	45 (1), 51 (1)
CmsRoute	interface	core-libs/core/src/routing/models/cms-route.ts:L15-L17	14 (1), 51 (4)
CmsRouteData	interface	core-libs/core/src/routing/models/cms-route.ts:L10-L13	14 (1), 51 (1)
CmsRouteModule	class	core-libs/storefront/cms-structure/routing/cms-route/cms-route.module.ts:L20-L20	05 (1), 10 (1), 14 (3)
CmsRoutesImplService	class	core-libs/storefront/cms-structure/services/cms-routes-impl.service.ts:L33-L183	03 (2), 05 (1), 12 (1), 14 (2), 16 (5), 51 (2), 54 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CmsRoutesService	class	core-libs/storefront/cms-structure/services/cms-routes.service.ts:L17-L24	05 (2), 14 (1), 16 (4), 51 (1)
CmsService	class	core-libs/core/src/cms/facade/cms.service.ts:L36-L310	14 (1), 15 (9), 16 (1), 40 (1)
CmsSetPageFailIndex	class	core-libs/core/src/cms/store/actions/page.action.ts:L53-L69	15 (2)
CmsSetPageSuccessIndex	class	core-libs/core/src/cms/store/actions/page.action.ts:L45-L51	15 (3)
CmsStoreModule	class	core-libs/core/src/cms/store/cms-store.module.ts:L45-L45	15 (3)
CmsStructureConfig	class	core-libs/core/src/cms/config/cms-structure.config.ts:L79-L85	15 (3), 51 (5)
CmsStructureConfigService	class	core-libs/core/src/cms/services/cms-structure-config.service.ts:L30-L193	15 (6)
CmsStructureModel	interface	core-libs/core/src/cms/model/page.model.ts:L31-L34	15 (1)
CmsStructureOptions	interface	core-libs/storefront/cms-structure/utils/cms-structure.model.ts:L15-L36	16 (1)
CMSTabParagraphContainer	interface	core-libs/core/src/model/cms.model.ts:L94-L97	45 (1)
CmsTicketInterceptor	class	feature-libs/smartedit/root/http-interceptors/cms-ticket.interceptor.ts:L20-L73	06 (4), 33 (4)
CombinedInjector	class	core-libs/core/src/util/combined-injector.ts:L27-L72	01 (5), 03 (9), 12 (6), 46 (9), 54 (1)
Command	class	core-libs/core/src/util/command-query/command.service.ts:L27-L29	20 (2), 23 (7), 26 (1), 28 (1), 29 (2), 31 (3), 32 (1), 34 (1), 35 (2)
CommandService	class	core-libs/core/src/util/command-query/command.service.ts:L43-L140	20 (6), 24 (2), 25 (2)
CommandStrategy	enum	core-libs/core/src/util/command-query/command.service.ts:L31-L38	20 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CommonConfiguratorComponentsModule	class	feature-libs/product-configurator/common/components/common-configurator-components.module.ts:L26-L26	30 (2)
CommonConfiguratorModule	class	feature-libs/product-configurator/common/common-configurator.module.ts:L14-L14	30 (3)
CommonConfiguratorOccModule	class	feature-libs/product-configurator/common/occ/common-configurator-occ.module.ts:L14-L14	30 (1)
CommonConfiguratorUISettingsConfig	class	feature-libs/product-configurator/common/components/config/common-configurator-ui-settings.config.ts:L23-L25	30 (2)
CommonConfiguratorUtilsService	class	feature-libs/product-configurator/common/shared/utils/common-configurator-utils.service.ts:L36-L195	30 (1)
CompactAddToCartComponent	class	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/compact-add-to-cart/compact-add-to-cart.component.ts:L29-L29	36 (1)
COMPONENT_ENTITY	const	core-libs/core/src/cms/store/cms-state.ts:L14-L14	15 (1)
ComponentCreateEvent	class	core-libs/storefront/cms-structure/page/component/events/component.event.ts:L29-L38	16 (5), 51 (1)
ComponentData	interface	core-libs/schematics/src/shared/utils/file-utils.ts:L70-L79	43 (1), 54 (1)
ComponentDataProvider	class	core-libs/storefront/cms-structure/page/component/services/component-data.provider.ts:L22-L60	16 (2), 51 (1)
ComponentDecorator	class	core-libs/core/src/cms/decorators/component-decorator.ts:L11-L23	12 (1), 15 (3), 33 (3)
ComponentDestroyEvent	class	core-libs/storefront/cms-structure/page/component/events/component.event.ts:L43-L48	16 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ComponentEvent	class	core-libs/storefront/cms-structure/page/component/events/component.event.ts:L15-L24	16 (1)
ComponentHandler	class	core-libs/storefront/cms-structure/page/component/handlers/component-handler.ts:L21-L45	16 (5)
ComponentHandlerService	class	core-libs/storefront/cms-structure/page/component/services/component-handler.service.ts:L33-L89	16 (2), 46 (2)
ComponentsContext	interface	core-libs/core/src/cms/store/cms-state.ts:L22-L62	15 (3)
ComponentsEffects	class	core-libs/core/src/cms/store/effects/components.effect.ts:L25-L120	15 (2)
ComponentWrapperDirective	class	core-libs/storefront/cms-structure/page/component/component-wrapper.directive.ts:L43-L151	01 (1), 04 (1), 12 (1), 15 (2), 16 (6), 51 (1)
Config	class	core-libs/core/src/config/config-tokens.ts:L21-L21 (+1)	01 (8), 02 (1), 07 (1), 10 (7), 11 (27), 12 (8), 13 (1), 14 (1), 15 (4), 18 (2), 19 (2), 21 (4), 22 (2), 25 (3), 26 (2), 27 (2), 29 (2), 30 (2), 31 (4), 32 (1), 33 (2), 35 (1), 36 (2), 37 (2), 40 (2), 41 (2), 42 (16), 46 (2), 47 (1), 50 (18), 51 (1), 52 (3), 53 (1), 54 (11)
CONFIG_INITIALIZER	const	core-libs/core/src/config/config-initializer/config-initializer.ts:L33-L35	01 (11), 02 (1), 06 (1), 10 (12), 11 (12), 13 (4), 14 (1), 37 (2), 50 (1)
CONFIG_INITIALIZER_FORROOT_GUARD	const	core-libs/core/src/config/config-initializer/config-initializer.ts:L37-L39	01 (2), 10 (2), 11 (4), 50 (1)
ConfigChunk	const	core-libs/core/src/config/config-tokens.ts:L53-L53	01 (9), 03 (2), 10 (1), 11 (18), 12 (4), 16 (1), 31 (1), 42 (6), 47 (4), 50 (9), 51 (3), 52 (1), 53 (4), 54 (2)
ConfigDeprecation	interface	core-libs/schematics/src/shared/utils/file-utils.ts:L107-L110	43 (1), 54 (1)
ConfigFactory	type	core-libs/core/src/config/config-factory.ts:L9-L9	11 (1), 50 (1)
configFactory()	function	core-libs/core/src/config/config-tokens.ts:L10-L12	01 (3), 10 (3), 11 (1)
ConfigInitializer	interface	core-libs/core/src/config/config-initializer/config-initializer.ts:L18-L28	01 (2), 10 (2), 11 (4), 13 (2), 34 (1), 54 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ConfigInitializerModule	class	core-libs/core/src/config/config-initializer/config-initializer.module.ts:L42-L68	05 (2), 10 (2), 11 (3), 13 (1)
ConfigInitializerService	class	core-libs/core/src/config/config-initializer/config-initializer.service.ts:L24-L176	01 (2), 10 (3), 11 (8), 13 (1), 37 (1), 50 (2)
ConfigModule	class	core-libs/core/src/config/config.module.ts:L14-L67	01 (3), 02 (5), 10 (3), 11 (7), 50 (1)
ConfigurableRoutesService	class	core-libs/core/src/routing/configurable-routes/configurable-routes.service.ts:L22-L186	03 (1), 05 (4), 10 (2), 14 (3)
ConfigurationService	class	core-libs/core/src/config/services/configuration.service.ts:L24-L114	01 (4), 02 (3), 10 (5), 11 (9), 12 (2), 29 (1), 31 (1), 42 (2), 50 (3), 54 (2)
ConfiguratorActions	type	feature-libs/product-configurator/textfield/core/state/actions/configurator-textfield.action.ts:L183-L195	30 (1)
ConfiguratorAttributeBaseComponent	class	feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-base.component.ts:L20-L615	30 (1)
ConfiguratorAttributeCompositionConfig	class	feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.config.ts:L22-L24	30 (1)
ConfiguratorAttributeCompositionContext	class	feature-libs/product-configurator/rulebased/components/attribute/composition/configurator-attribute-composition.model.ts:L12-L21	30 (3)
ConfiguratorAttributePriceChangeService	class	feature-libs/product-configurator/rulebased/components/attribute/price-change/configurator-attribute-price-change.service.ts:L31-L113	30 (1)
ConfiguratorAttributeSelectionBaseComponent	class	feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-selection-base.component.ts:L24-L61	30 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ConfiguratorAttributeSingleSelectionBaseComponent	class	feature-libs/product-configurator/rulebased/components/attribute/types/base/configurator-attribute-single-selection-base.component.ts:L23-L252	30 (1)
ConfiguratorBadRequestHandler	class	feature-libs/product-configurator/rulebased/root/http-interceptors/configurator-bad-request.handler.ts:L22-L126	30 (4)
ConfiguratorBasicEffectService	class	feature-libs/product-configurator/rulebased/core/state/effects/configurator-basic-effect.service.ts:L17-L190	30 (1)
ConfiguratorCartEntryBundleInfoComponent	class	feature-libs/product-configurator/common/components/configurator-cart-entry-bundle-info/configurator-cart-entry-bundle-info.component.ts:L66-L420	30 (1)
ConfiguratorCartEntryInfoComponent	class	feature-libs/product-configurator/common/components/configurator-cart-entry-info/configurator-cart-entry-info.component.ts:L26-L79	30 (1)
ConfiguratorCartService	class	feature-libs/product-configurator/rulebased/core/facade/configurator-cart.service.ts:L25-L293	30 (3)
ConfiguratorCommonsService	class	feature-libs/product-configurator/rulebased/core/facade/configurator-commons.service.ts:L25-L539	30 (2)
ConfiguratorConflictSolverDialogLauncherService	class	feature-libs/product-configurator/rulebased/components/conflict-solver-dialog/configurator-conflict-solver-dialog-launcher.service.ts:L34-L114	30 (1)
ConfiguratorCoreConfig	class	feature-libs/product-configurator/rulebased/core/config/configurator-core.config.ts:L18-L20	30 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ConfiguratorExpertModeService	class	feature-libs/product-configurator/rulebased/core/services/configurator-expert-mode.service.ts:L18-L57	30 (4)
ConfiguratorFormComponent	class	feature-libs/product-configurator/rulebased/components/form/configurator-form.component.ts:L50-L213	30 (2)
ConfiguratorFormModule	class	feature-libs/product-configurator/rulebased/components/form/configurator-form.module.ts:L33-L33	30 (2)
ConfiguratorGroupComponent	class	feature-libs/product-configurator/rulebased/components/group/configurator-group.component.ts:L35-L140	30 (1)
ConfiguratorGroupModule	class	feature-libs/product-configurator/rulebased/components/group/configurator-group.module.ts:L72-L72	30 (3)
ConfiguratorGroupsService	class	feature-libs/product-configurator/rulebased/core/facade/configurator-groups.service.ts:L24-L485	30 (2)
ConfiguratorIssuesNotificationComponent	class	feature-libs/product-configurator/common/components/configurator-issues-notification/configurator-issues-notification.component.ts:L38-L97	30 (1)
ConfiguratorMessageConfig	class	feature-libs/product-configurator/rulebased/components/config/configurator-message.config.ts:L13-L15	30 (1)
ConfiguratorModelUtils	class	feature-libs/product-configurator/common/shared/utils/configurator-model-utils.ts:L13-L69	30 (1)
ConfiguratorOverviewFormComponent	class	feature-libs/product-configurator/rulebased/components/overview-form/configurator-overview-form.component.ts:L72-L558	04 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ConfiguratorQuantityService	class	feature-libs/product-configurator/rulebased/core/services/configurator-quantity.service.ts:L16-L36	30 (3)
ConfiguratorRouterExtractorService	class	feature-libs/product-configurator/common/components/service/configurator-router-extractor.service.ts:L20-L114	30 (4)
ConfiguratorRouterListener	class	feature-libs/product-configurator/rulebased/core/facade/routing/configurator-router.listener.ts:L14-L43	30 (1)
ConfiguratorRouterModule	class	feature-libs/product-configurator/rulebased/core/facade/routing/configurator-router.module.ts:L11-L15	30 (1)
ConfiguratorTextfieldAdapter	class	feature-libs/product-configurator/textfield/core/connectors/configurator-textfield.adapter.ts:L12-L66	30 (1)
ConfiguratorTextfieldConnector	class	feature-libs/product-configurator/textfield/core/connectors/configurator-textfield.connector.ts:L15-L72	30 (1)
ConfiguratorType	enum	feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L60-L64	30 (1)
ConfiguratorUISettingsConfig	class	feature-libs/product-configurator/rulebased/components/config/configurator-ui-settings.config.ts:L55-L57	30 (2)
ConfiguratorUpdateMessageModule	class	feature-libs/product-configurator/rulebased/components/update-message/configurator-update-message.module.ts:L35-L35	30 (1)
ConfigureCartEntryComponent	class	feature-libs/product-configurator/common/components/configure-cart-entry/configure-cart-entry.component.ts:L50-L347	30 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ConfigureProductComponent	class	feature-libs/product-configurator/common/components/configure-product/configure-product.component.ts:L38-L160	30 (2)
ConfigureProductModule	class	feature-libs/product-configurator/common/components/configure-product/configure-product.module.ts:L46-L46	30 (1)
ConfigValidator	type	core-libs/core/src/config/config-validator/config-validator.ts:L20-L20	11 (5), 50 (2)
ConfigValidatorModule	class	core-libs/core/src/config/config-validator/config-validator.module.ts:L46-L63	01 (1), 10 (1), 11 (2)
ConfigValidatorToken	const	core-libs/core/src/config/config-validator/config-validator.ts:L11-L13	01 (2), 11 (3), 50 (1)
ConfirmActionDialogMappingConfig	type	feature-libs/quote/components/config/quote-ui.config.ts:L41-L45	31 (1)
ConfirmationMessageComponent	class	feature-libs/organization/administration/components/shared/message/confirmation/confirmation-message.component.ts:L31-L64	28 (1)
ConflictHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/conflict/conflict.handler.ts:L16-L29	41 (1), 50 (2)
ConflictHeaderId	const	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L387-L387	30 (1)
ConflictIdPrefix	const	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L386-L386	30 (1)
ConsentManagementComponent	class	core-libs/storefront/cms-components/myaccount/consent-management/components/consent-management.component.ts:L64-L403	04 (1), 23 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ConsentManagementComponentService	class	core-libs/storefront/cms-components/myaccount/consent-management/consent-management-component.service.ts:L11-L25	34 (3)
ConsentManagementModule	class	core-libs/storefront/cms-components/myaccount/consent-management/consent-management.module.ts:L63-L63	34 (2)
ConsentReferenceInterceptor	class	integration-libs/cds/src/profiletag/http-interceptors/consent-reference-interceptor.ts:L19-L46	06 (3), 34 (2)
ConsentService	class	core-libs/core/src/user/facade/consent.service.ts:L18-L126	23 (3)
Consignment	interface	feature-libs/order/root/model/order.model.ts:L74-L82 (+1)	32 (1)
CONSIGNMENT_TRACKING_NORMALIZER	const	feature-libs/order/root/tokens/converters.ts:L40-L42	27 (1)
ConstructorDeprecation	interface	core-libs/schematics/src/shared/utils/file-utils.ts:L81-L91	43 (1), 54 (1)
ContainerRowGroupIdPrefix	const	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L388-L388	30 (1)
ContentPageMetaResolver	class	core-libs/core/src/cms/page/content-page-meta.resolver.ts:L32-L65	40 (5), 51 (2), 52 (2)
ContentSlotComponentData	interface	core-libs/core/src/cms/model/content-slot-component-data.model.ts:L7-L12	15 (3), 16 (1)
ContentSlotData	interface	core-libs/core/src/cms/model/content-slot-data.model.ts:L9-L12	15 (3)
ContextService	class	core-libs/storefront/context/context.service.ts:L17-L38	14 (4)
ContextServiceMap	class	core-libs/core/src/site-context/providers/context-service-map.ts:L18-L20	10 (1), 13 (10)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ContextToken	type	core-libs/storefront/context/context.model.ts:L7-L7	54 (1)
Converter	interface	core-libs/core/src/util/converter.service.ts:L21-L30	01 (1), 21 (1), 46 (3), 52 (5)
ConverterService	class	core-libs/core/src/util/converter.service.ts:L35-L142	01 (5), 12 (1), 21 (8), 24 (1), 34 (1), 44 (2), 46 (10), 52 (5)
COST_CENTER_NORMALIZER	const	core-libs/core/src/cost-center/connectors/cost-center/converters.ts:L12-L14	28 (1)
COST_CENTER_SERIALIZER	const	core-libs/core/src/cost-center/connectors/cost-center/converters.ts:L20-L22	52 (2)
CostCenter	interface	core-libs/core/src/model/org-unit.model.ts:L11-L16 (+1)	28 (1), 46 (3), 54 (1)
CostCenterOccModule	class	core-libs/core/src/occ/adapters/cost-center/cost-center-occ.module.ts:L40-L40	21 (1), 28 (3), 43 (1), 47 (1)
Country	interface	core-libs/core/src/model/address.model.ts:L9-L12 (+1)	46 (1)
CountryType	enum	core-libs/core/src/model/address.model.ts:L14-L17	46 (2)
CouponClaimComponent	class	core-libs/storefront/cms-components/myaccount/my-coupons/coupon-claim/coupon-claim.component.ts:L20-L59	23 (2)
CouponSearchPageResolver	class	core-libs/core/src/product/services/coupon-search-page-meta.resolver.ts:L33-L96	24 (3), 40 (6), 54 (1)
CPQ_CONFIGURATOR_NORMALIZER	const	feature-libs/product-configurator/rulebased/cpq/common/converters/cpq-configurator.converters.ts:L12-L14	30 (1)
CPQ_QUOTE_MODULE	const	core-libs/schematics/src/shared/lib-configs/integration-libs/cpq-quote-schematics-config.ts:L23-L23	36 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CpqConfiguratorCommonModule	class	feature-libs/product-configurator/rulebased/cpq/common/cpq-configurator-common.module.ts:L45-L45	30 (1)
CpqConfiguratorOccAdapter	class	feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.adapter.ts:L23-L161	30 (2)
CpqConfiguratorOccModule	class	feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.module.ts:L41-L41	30 (2)
CpqConfiguratorOccService	class	feature-libs/product-configurator/rulebased/cpq/occ/cpq-configurator-occ.service.ts:L31-L455	30 (2)
CpqConfiguratorPageLayoutHandler	class	feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-page-layout-handler.ts:L31-L161	30 (3)
CpqConfiguratorRootModule	class	feature-libs/product-configurator/rulebased/root/cpq/cpq-configurator-root.module.ts:L21-L21	30 (5)
CpqDiscountModule	class	integration-libs/cpq-quote/cpq-quote-discount/cpq-discount.module.ts:L59-L59	36 (2)
CpqQuoteDiscountComponent	class	integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-discount-tbody/cpq-quote.component.ts:L23-L72	36 (1)
CpqQuoteHeadingComponent	class	integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote-heading/cpq-quote-heading.component.ts:L27-L80	36 (2)
CpqQuoteModule	class	integration-libs/cpq-quote/cpq-quote.module.ts:L13-L13	36 (5), 47 (2)
CpqQuoteOfferComponent	class	integration-libs/cpq-quote/cpq-quote-discount/components/cpq-quote/cpq-quote-offer.component.ts:L24-L75	36 (1)
CpqQuoteRootModule	class	integration-libs/cpq-quote/root/cpq-quote-root.module.ts:L10-L10	36 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CpqQuoteService	class	integration-libs/cpq-quote/cpq-quote-discount/cpq-qute.service.ts:L13-L23	36 (3)
CpqQuoteSharedService	class	integration-libs/cpq-quote/cpq-quote-discount/cpq-qute-shared.service.ts:L16-L24	36 (6)
CREATE_CART	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L12-L12	25 (1)
CREATE_CART_SUCCESS	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L14-L14	25 (2)
CreateB2BUser	class	feature-libs/organization/administration/core/store/actions/b2b-user.action.ts:L128-L134	28 (1)
CreateB2BUserSuccess	class	feature-libs/organization/administration/core/store/actions/b2b-user.action.ts:L147-L153	28 (2)
CreateBudgetSuccess	class	feature-libs/organization/administration/core/store/actions/budget.action.ts:L124-L130	28 (1)
CreateCart	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L52-L57	25 (4)
CreateCartEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L35-L40	25 (2), 35 (1)
CreateCartSuccess	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L78-L83	25 (3)
CreateConfiguration	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L91-L103 (+1)	30 (2)
CreateConfigurationSuccess	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L121-L127 (+1)	30 (1)
createStylePreprocessorOptions()	function	core-libs/schematics/src/add-spartacus/index.ts:L371-L439	43 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CreateWishList	class	feature-libs/cart/wish-list/core/store/actions/wish-list.action.ts:L20-L30	25 (1)
CsAgentAuthService	class	feature-libs/asm/root/services/csagent-auth.service.ts:L31-L210	29 (3)
CSAgentLoginFormComponent	class	feature-libs/asm/components/csagent-login-form/csagent-login-form.component.ts:L37-L65	29 (1)
CsrfStateService	class	core-libs/core/src/auth/user-auth/facade/csrf-state.service.ts:L16-L26	22 (2)
Currency	interface	core-libs/core/src/model/misc.model.ts:L10-L15 (+1)	46 (2)
CURRENCY_CHANGE	const	core-libs/core/src/site-context/store/actions/currencies.action.ts:L15-L15	13 (1), 23 (1), 24 (1), 25 (1), 27 (1), 32 (1)
CURRENCY_CONTEXT_ID	const	core-libs/core/src/site-context/providers/context-ids.ts:L8-L8	13 (1)
CurrencyChange	class	core-libs/core/src/site-context/store/actions/currencies.action.ts:L39-L44	13 (2)
CurrencyInitializer	class	core-libs/core/src/site-context/services/currency-initializer.ts:L19-L69	10 (2)
CurrencyService	class	core-libs/core/src/site-context/facade/currency.service.ts:L26-L104	06 (1), 13 (4)
CurrencySetEvent	class	core-libs/core/src/site-context/events/site-context.events.ts:L23-L29	13 (5), 23 (1), 26 (3)
CurrencyStatePersistenceService	class	core-libs/core/src/site-context/services/currency-state-persistence.service.ts:L15-L43	13 (1)
CURRENT_CONTEXT_KEY	const	core-libs/core/src/cms/utils/cms-utils.ts:L10-L10	15 (1)
CurrentBudgetService	class	feature-libs/organization/administration/components/budget/services/current-budget.service.ts:L20-L43	28 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CurrentItemService	class	feature-libs/organization/administration/components/shared/current-item.service.ts:L19-L70	28 (3)
CurrentLocationService	class	feature-libs/pickup-in-store/components/services/current-location.service.ts:L16-L39	32 (1)
CurrentProductService	class	core-libs/storefront/cms-components/product/current-product.service.ts:L21-L58	24 (2), 25 (1), 30 (1)
CUSTOMER_LISTS_NORMALIZER	const	feature-libs/asm/core/connectors/converters.ts:L15-L17	29 (1)
CUSTOMER_SEARCH_PAGE_NORMALIZER	const	feature-libs/asm/core/connectors/converters.ts:L11-L13 (+1)	29 (1)
CustomerCouponAdapter	class	core-libs/core/src/user/connectors/customer-coupon/customer-coupon.adapter.ts:L14-L46	23 (1)
CustomerCouponSearchResult	interface	core-libs/core/src/model/customer-coupon.model.ts:L27-L31 (+1)	46 (1)
CustomerEffects	class	feature-libs/asm/core/store/effects/customer.effect.ts:L17-L74	29 (1)
CustomerEmulationComponent	class	feature-libs/asm/components/customer-emulation/customer-emulation.component.ts:L29-L99	29 (1)
CustomerListColumnActionType	enum	feature-libs/asm/root/model/customer-list.model.ts:L28-L33	29 (1)
CustomerListComponent	class	feature-libs/asm/components/customer-list/customer-list.component.ts:L78-L440	29 (1)
CustomerSearch	class	feature-libs/asm/core/store/actions/customer.action.ts:L28-L33	29 (1)
CustomerSearchFail	class	feature-libs/asm/core/store/actions/customer.action.ts:L35-L43	29 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CustomerSelectionComponent	class	feature-libs/asm/components/customer-selection/customer-selection.component.ts:L61-L434	29 (1)
CustomerTicketingCloseComponentService	class	feature-libs/customer-ticketing/components/details/customer-ticketing-close/customer-ticketing-close-component.service.ts:L15-L31	34 (1)
CustomerTicketingConfig	class	feature-libs/customer-ticketing/root/config/customer-ticketing-config.ts:L14-L27	31 (2)
CustomerTicketingDialogComponent	class	feature-libs/customer-ticketing/components/shared/customer-ticketing-dialog/customer-ticketing-dialog.component.ts:L27-L123	31 (2)
CustomerTicketingEventModule	class	feature-libs/customer-ticketing/root/events/customer-ticketing-event.module.ts:L11-L15	31 (2)
CustomerTicketingFacade	class	feature-libs/customer-ticketing/root/facade/customer-ticketing.facade.ts:L41-L85	31 (1)
CustomerTicketingMessagesComponentService	class	feature-libs/customer-ticketing/components/details/customer-ticketing-messages/customer-ticketing-messages-component.service.ts:L11-L15	34 (1)
CustomerTicketingModule	class	feature-libs/customer-ticketing/customer-ticketing.module.ts:L19-L19	31 (1), 34 (4)
CustomerTicketingPageMetaResolver	class	feature-libs/customer-ticketing/core/services/customer-ticketing-page-meta.resolver.ts:L30-L102	31 (2), 40 (2)
CustomerTicketingReopenComponentService	class	feature-libs/customer-ticketing/components/details/customer-ticketing-reopen/customer-ticketing-reopen-component.service.ts:L15-L32	34 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
CustomerTicketingService	class	feature-libs/customer-ticketing/core/facade/customer-ticketing.service.ts:L56-L373	31 (3)
CustomFormValidators	class	core-libs/storefront/shared/utils/validators/custom-form-validators.ts:L26-L433	45 (2)
CustomLoginGuard	class	core-libs/core/src/auth/user-auth/guards/custom-login.guard.ts:L50-L138	22 (7), 23 (1), 34 (2)
CustomSerializer	class	core-libs/core/src/routing/store/reducers/router.reducer.ts:L109-L240	05 (11), 14 (5), 20 (1), 51 (3)
CX_KEY	const	core-libs/core/src/state/reducers/transfer-state.reducer.ts:L16-L16	07 (4), 13 (1), 19 (3), 54 (1)
CxApiService	class	core-libs/storefront/cms-structure/page/component/services/cx-api.service.ts:L36-L76	04 (1), 16 (3), 42 (1)
CxCommonEngine	class	core-libs/setup/ssr/engine/cx-common-engine.ts:L20-L67	04 (1), 07 (8), 39 (5), 47 (2)
CxDatePipe	class	core-libs/core/src/i18n/date.pipe.ts:L14-L56	36 (2), 37 (6)
CxErrorHandler	class	core-libs/core/src/error-handling/cx-error-handler.ts:L19-L31	02 (1), 39 (3), 41 (15)
CxErrorHandlerEffect	class	core-libs/core/src/error-handling/effects-error-handler/cx-error-handler.effect.ts:L19-L33	32 (3), 41 (6)
CxEvent	class	core-libs/core/src/event/cx-event.ts:L12-L17	12 (2), 20 (12), 24 (1), 25 (1), 26 (2), 31 (1), 33 (2), 50 (8), 53 (1)
CxNumericPipe	class	core-libs/core/src/i18n/numeric.pipe.ts:L13-L53	37 (6)
CxRovingTabindexDirective	class	core-libs/storefront/shared/directives/roving-tabindex/roving-tabindex.directive.ts:L59-L430	45 (2)

D

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
Data	interface	feature-libs/product-configurator/common/components/service/configurator-router-data.ts:L13-L29	30 (1)
DatePickerComponent	class	core-libs/storefront/shared/components/form/date-picker/date-picker.component.ts:L38-L69	37 (1)
DatePickerService	class	core-libs/storefront/shared/components/form/date-picker/date-picker.service.ts:L24-L77	37 (2)
DateRangeModalComponent	class	feature-libs/subscription-billing/components/list/billing/date-range-modal/date-range-modal.component.ts:L36-L95	31 (3)
DateValidationService	class	feature-libs/requested-delivery-date/root/components/shared/date-validation.service.ts:L12-L52	32 (4)
DebugInterceptor	class	integration-libs/cds/src/profiletag/http-interceptors/debug-interceptor.ts:L19-L42	06 (3), 34 (3)
DEFAULT_AUTH_HTTP_HEADER_SERVICE	const	core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L55-L71	22 (1), 36 (2)
DEFAULT_AUTHORIZATION_ERROR_RETRIES_COUNT	const	core-libs/core/src/util/occ-http-error-handlers.ts:L31-L31	35 (1), 46 (2)
DEFAULT_LCP_PRESENCE	const	core-libs/storefront/shared/lcp-context/lcp-presence.token.ts:L11-L13	16 (1)
DEFAULT_SCOPE	const	core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L7-L7	21 (3), 24 (1), 52 (2)
DEFAULT_SERVER_ERROR_RETRIES_COUNT	const	core-libs/core/src/util/occ-http-error-handlers.ts:L30-L30	46 (1)
DEFAULT_URL_MATCHER	const	core-libs/core/src/routing/url-matcher/default-url-matcher.ts:L35-L45	05 (3), 14 (3), 16 (1)
DefaultCacheEntrySizeCalculator	class	core-libs/setup/ssr/optimized-engine/rendering-cache/default-cache-entry-size-calculator.ts:L24-L75	39 (4)
defaultCheckoutComponentsConfig()	function	feature-libs/checkout/b2b/root/checkout-b2b-root.module.ts:L33-L42 (+2)	26 (2)
DefaultComponentHandler	class	core-libs/storefront/cms-structure/page/component/handlers/default-component.handler.ts:L27-L86	03 (3), 12 (1), 16 (8), 51 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
DefaultConfig	const	core-libs/core/src/config/config-tokens.ts:L32-L35	01 (6), 10 (2), 11 (9), 12 (1), 42 (3), 50 (2), 51 (2), 53 (1)
DefaultConfigChunk	const	core-libs/core/src/config/config-tokens.ts:L60-L62	01 (10), 03 (1), 10 (1), 11 (14), 12 (4), 16 (1), 29 (1), 30 (1), 31 (1), 32 (1), 42 (4), 47 (1), 50 (4), 51 (3), 53 (2)
DefaultExpressServerLogger	class	core-libs/setup/ssr/logger/loggers/default-express-server-logger.ts:L28-L133	39 (6)
DefaultFeatureToggles	const	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L33-L43	42 (4), 44 (1)
DefaultFeatureTogglesChunk	const	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L75-L78	42 (4)
defaultOrderComponentsConfig()	function	feature-libs/order/root/order-root.module.ts:L68-L80	27 (1)
DefaultRoutePageMetaResolver	class	core-libs/core/src/cms/page/routing/default-route-page-meta.resolver.ts:L22-L79	40 (4)
DeferLoaderService	class	core-libs/storefront/layout/loading/defer-loader.service.ts:L22-L65	07 (1), 16 (1), 17 (4), 18 (5)
DeferLoadingStrategy	enum	core-libs/core/src/cms/config/cms-config.ts:L96-L101	15 (1), 16 (2), 17 (1), 18 (1), 51 (3)
DELEGATED_AUTH_HTTP_HEADER_SERVICE	const	core-libs/core/src/auth/user-auth/services/auth-http-header.service.ts:L79-L82	22 (1), 29 (1), 36 (2)
DELETE_CART	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L35-L35	20 (1)
DELETE_CART_SUCCESS	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L36-L36	25 (1)
DeleteCart	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L270-L273	25 (1)
DeleteCartEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L180-L185	19 (1), 26 (1)
DeleteCartFail	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L282-L288	25 (2)
DeleteCartFailEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L194-L199	25 (1)
DeleteCartSuccess	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L275-L280	25 (1)
DeleteCartSuccessEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L187-L192	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
DeleteItemComponent	class	feature-libs/organization/administration/components/shared/detail/delete-item-action/delete-item.component.ts:L28-L122	28 (1)
DeleteUserAddress	class	core-libs/core/src/user/store/actions/user-addresses.action.ts:L114-L120	23 (1)
DeleteUserAddressEvent	class	core-libs/core/src/user/events/user.events.ts:L20-L23	23 (2), 26 (1)
DELIVERY_MODE_NORMALIZER	const	feature-libs/checkout/base/core/connectors/checkout-delivery-modes/converters.ts:L11-L13	26 (1)
DeliveryMode	interface	core-libs/core/src/occ/occ-models/occ.models.ts:L516-L524 (+1)	25 (1), 36 (1), 54 (1)
DeliveryModeDatePickerComponent	class	feature-libs/requested-delivery-date/root/components/delivery-mode-date-picker/delivery-mode-date-picker.component.ts:L53-L181	32 (2)
DeliveryModePreferences	enum	feature-libs/checkout/base/root/config/checkout-config.ts:L12-L16	26 (1)
DeliveryPointsService	class	feature-libs/pickup-in-store/components/services/delivery-points.service.ts:L25-L161	32 (2)
DeprecatedNode	interface	core-libs/schematics/src/shared/utils/file-utils.ts:L101-L105	43 (1), 54 (1)
DIALOG_TYPE	enum	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L81-L88	18 (3)
DigitalPaymentsModule	class	integration-libs/digital-payments/src/digital-payments.module.ts:L13-L13	35 (5), 47 (3)
Direction	interface	core-libs/storefront/layout/direction/config/direction.model.ts:L29-L56	38 (1)
DirectionConfig	class	core-libs/storefront/layout/direction/config/direction.config.ts:L29-L31	38 (2)
DirectionMode	enum	core-libs/storefront/layout/direction/config/direction.model.ts:L14-L23	38 (3)
DirectionModule	class	core-libs/storefront/layout/direction/direction.module.ts:L35-L35	18 (1), 38 (1)
DirectionService	class	core-libs/storefront/layout/direction/direction.service.ts:L31-L118	11 (1), 38 (7)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
DirectiveStateTransferService	class	core-libs/storefront/utils/directive-state-transfer.service.ts:L18-L42	07 (4), 39 (1), 46 (1)
DissmissConflictDialoge	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L496-L502	30 (1)
DOCUMENT	const	core-libs/schematics/src/shared/constants.ts:L1004-L1004	02 (1), 07 (1), 46 (3)
DomChangeDirective	class	core-libs/storefront/layout/a11y/on-dom-change/dom-change.directive.ts:L29-L66	18 (2)
DownloadOrderInvoicesEvent	class	feature-libs/order/root/events/order.events.ts:L63-L72	27 (2)
DP_DETAILS_NORMALIZER	const	integration-libs/digital-payments/src/checkout/adapters/converters.ts:L11-L13	35 (1)
DP_REQUEST_NORMALIZER	const	integration-libs/digital-payments/src/checkout/adapters/converters.ts:L15-L17	35 (1)
DpCheckoutModule	class	integration-libs/digital-payments/src/checkout/dp-checkout.module.ts:L42-L42	35 (4)
DpCheckoutPaymentService	class	integration-libs/digital-payments/src/checkout/facade/dp-checkout-payment.service.ts:L27-L115	35 (2)
DpPaymentFormComponent	class	integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-form/dp-payment-form.component.ts:L23-L58	35 (1)
DpPaymentMethodComponent	class	integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.component.ts:L46-L94	35 (3)
DpPaymentMethodModule	class	integration-libs/digital-payments/src/checkout/cms-components/dp-payment-method/dp-payment-method.module.ts:L45-L45	35 (2)
DynamicAttributeService	class	core-libs/core/src/cms/services/dynamic-attribute.service.ts:L23-L65	12 (1), 15 (3), 33 (2)
DynamicTemplate	class	integration-libs/cds/src/utils/dynamic-template.ts:L7-L18	34 (1)

E

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
EditSavedCart	class	feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L238-L251	25 (4)
EffectsErrorHandlerModule	class	core-libs/core/src/error-handling/effects-error-handler/effects-error-handler.module.ts:L15-L22	41 (2)
EMAIL_PATTERN	const	core-libs/core/src/util/regex-pattern.ts:L8-L9	46 (1)
ENDS_WITH_LEGAL_CHARACTER_PATTERN	const	core-libs/core/src/util/regex-pattern.ts:L35-L36	46 (1)
EntitiesModel	interface	core-libs/core/src/model/misc.model.ts:L67-L71	46 (1)
EntityFailAction	class	core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L79-L88	41 (1)
EntityLoadAction	class	core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L71-L77	19 (1), 23 (1), 25 (1), 27 (1)
EntityLoaderResetAction	class	core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L102-L108	15 (1), 30 (1)
EntityLoaderState	type	core-libs/core/src/state/utils/entity-loader/entity-loader-state.ts:L10-L10	19 (2)
EntityMeta	interface	core-libs/core/src/state/utils/entity/entity.action.ts:L14-L18	19 (1)
EntityProcessesDecrementAction	class	core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader.action.ts:L80-L88	25 (1), 30 (1)
EntityProcessesIncrementAction	class	core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader.action.ts:L70-L78	19 (1), 25 (1), 30 (1)
EntityProcessesLoaderResetAction	class	core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader.action.ts:L60-L68	19 (1)
EntityProcessesLoaderState	type	core-libs/core/src/state/utils/entity-processes-loader/entity-processes-loader-state.ts:L10-L12	30 (1)
EntityRemoveAction	class	core-libs/core/src/state/utils/entity/entity.action.ts:L48-L54	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
EntityRemoveAllAction	class	core-libs/core/src/state/utils/entity/entity.action.ts:L56-L62	25 (1)
EntityScopedResetAction	class	core-libs/core/src/state/utils/scoped-loader/entity-scoped-loader.actions.ts:L117-L124	24 (1)
EntityState	interface	core-libs/core/src/state/utils/entity/entity-state.ts:L7-L11	19 (1)
EntitySuccessAction	class	core-libs/core/src/state/utils/entity-loader/entity-loader.action.ts:L90-L100	25 (2)
EpdVisualizationApiModule	class	integration-libs/epd-visualization/epd-visualization-api/epd-visualization-api.module.ts:L24-L24	36 (1)
EpdVisualizationConfig	class	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L19-L24	36 (2)
EpdVisualizationRootModule	class	integration-libs/epd-visualization/root/epd-visualization-root.module.ts:L38-L38	36 (1)
ErrorAction	interface	core-libs/core/src/error-handling/effects-error-handler/error-action.ts:L9-L11	19 (1), 41 (3)
ErrorActionService	class	core-libs/core/src/error-handling/effects-error-handler/error-action.service.ts:L20-L57	41 (9)
ErrorHandlingModule	class	core-libs/core/src/error-handling/error-handling.module.ts:L19-L32	02 (2), 06 (3), 41 (4)
ErrorModel	interface	core-libs/core/src/model/misc.model.ts:L92-L98 (+1)	41 (2), 46 (2)
EscapeFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L64-L70	18 (1)
EscapeFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/escape/escape-focus.service.ts:L15-L43	18 (1)
EstimatedDeliveryDateComponent	class	feature-libs/estimated-delivery-date/show-estimated-delivery-date/components/estimated-delivery-date.component.ts:L25-L74	32 (2), 51 (1)
EstimatedDeliveryDateModule	class	feature-libs/estimated-delivery-date/estimated-delivery-date.module.ts:L13-L13	32 (2), 47 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
EstimatedDeliveryDateRootModule	class	feature-libs/estimated-delivery-date/root/estimated-delivery-date-root.module.ts:L10-L10	32 (1)
EventService	class	core-libs/core/src/event/event.service.ts:L47-L223	12 (1), 13 (1), 16 (1), 19 (1), 20 (13), 22 (2), 32 (1), 33 (1), 45 (1), 50 (5), 51 (1), 53 (3), 54 (1)
EXPIRED_REFRESH_TOKEN_HANDLERS	const	core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts:L21-L23	22 (5), 36 (6)
ExpiredRefreshTokenHandler	interface	core-libs/core/src/auth/user-auth/services/auth-http-header-handler.ts:L17-L19	22 (5)
EXPRESS_SERVER_LOGGER	const	core-libs/setup/ssr/logger/loggers/express-server-logger.ts:L46-L48	39 (7)
ExpressCheckoutService	class	feature-libs/checkout/base/components/services/express-checkout.service.ts:L30-L176	26 (2)
ExpressLoggerService	class	core-libs/setup/ssr/logger/services/express-logger.service.ts:L25-L62	07 (1), 39 (7), 41 (1)
ExpressServerLogger	interface	core-libs/setup/ssr/logger/loggers/express-server-logger.ts:L30-L36	39 (5)
ExpressServerLoggerContext	interface	core-libs/setup/ssr/logger/loggers/express-server-logger.ts:L14-L17	39 (1)
ExtendSubscriptionFrequencyDropdownOptionsConfig	class	feature-libs/subscription-billing/root/config/extend-subscription-dropdown-options-config.ts:L14-L19	31 (2)
ExternalRoutesConfig	class	core-libs/core/src/routing/external-routes/external-routes-config.ts:L18-L20	14 (2)
ExternalRoutesGuard	class	core-libs/core/src/routing/external-routes/external-routes.guard.ts:L13-L42	05 (1), 14 (1)
ExternalRoutesModule	class	core-libs/core/src/routing/external-routes/external-routes.module.ts:L15-L29	05 (1), 47 (1)
ExternalRoutesService	class	core-libs/core/src/routing/external-routes/external-routes.service.ts:L19-L71	14 (1)

F

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
FacadeDescriptor	interface	core-libs/core/src/lazy-loading/facade-factory/facade-descriptor.ts:L10-L37	12 (2), 53 (2)
facadeFactory()	function	core-libs/core/src/lazy-loading/facade-factory/facade-factory.ts:L20-L24	12 (1), 53 (1)
FacadeFactoryService	class	core-libs/core/src/lazy-loading/facade-factory/facade-factory.service.ts:L32-L141	03 (1), 12 (2), 20 (2), 32 (1)
Facet	interface	core-libs/core/src/model/product-search.model.ts:L35-L61 (+1)	46 (4), 54 (1)
FacetChangedEvent	class	core-libs/core/src/product/event/product.events.ts:L12-L37	24 (3)
FacetService	class	core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/facet.service.ts:L27-L160	24 (3)
FEATURE_TOGGLES_COPIED_FROM_CORE_LIB_PATH	const	core-libs/schematics/src/shared/constants.ts:L1176-L1177	42 (1)
FeatureConfigService	class	core-libs/core/src/features-config/services/feature-config.service.ts:L21-L34	42 (14), 44 (1), 54 (7)
FeatureDirective	class	core-libs/core/src/features-config/directives/feature.directive.ts:L12-L30	42 (3), 44 (1)
FeatureLevelDirective	class	core-libs/core/src/features-config/directives/feature-level.directive.ts:L14-L32	42 (4), 54 (1)
FeatureModuleConfig	interface	core-libs/core/src/cms/config/cms-config.ts:L109-L122	03 (2), 12 (2), 15 (2), 33 (1), 47 (2), 51 (1), 53 (2)
FeatureModulesService	class	core-libs/core/src/lazy-loading/feature-modules.service.ts:L16-L124	03 (4), 12 (8), 16 (1), 33 (1), 47 (2), 50 (1)
FeaturesConfig	class	core-libs/core/src/features-config/config/features-config.ts:L31-L118	11 (1), 42 (11)
FeaturesConfigModule	class	core-libs/core/src/features-config/features-config.module.ts:L27-L53	42 (3)
FeatureStylesService	class	core-libs/core/src/features-config/services/feature-styles.service.ts:L26-L163	18 (1), 38 (7), 42 (8)
FeatureToggleExpression	type	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L28-L28	42 (3)
FeatureToggleKey	type	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L27-L27	42 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
FeatureToggles	class	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L24-L24 (+1)	42 (24), 44 (3), 47 (3), 54 (7)
FeatureTogglesChunk	const	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L64-L67	42 (6), 44 (1), 47 (2)
FeatureTogglesFactory	type	core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L14-L14	42 (2)
FeatureTogglesInterface	interface	core-libs/core/src/features-config/feature-toggles/config/feature-toggles.ts:L11-L1011	42 (6)
FederatedLoginConfig	class	core-libs/core/src/federated-login/config/federated-login-config.ts:L14-L47	22 (2)
FederatedLoginContextSerializerService	class	core-libs/core/src/federated-login/services/federated-login-context-serializer.service.ts:L13-L72	46 (1)
FederatedLoginGuard	class	core-libs/core/src/auth/user-auth/guards/federated-login.guard.ts:L32-L105	05 (1), 10 (1), 14 (2), 22 (4)
FederatedLoginModule	class	core-libs/core/src/federated-login/federated-login.module.ts:L17-L17	02 (1), 10 (1), 22 (1)
FederatedLoginService	class	core-libs/core/src/federated-login/services/federated-login.service.ts:L17-L106	22 (1)
FileDownloadService	class	core-libs/storefront/shared/services/file/file-download.service.ts:L12-L30	45 (1)
FilesFormValidators	class	core-libs/storefront/shared/services/file/files-form-validators.ts:L13-L86	45 (1)
FIND_STORE_BY_ID_SUCCESS	const	feature-libs/storefinder/core/store/actions/find-stores.action.ts:L22-L23	32 (1)
FIND_STORES_SUCCESS	const	feature-libs/storefinder/core/store/actions/find-stores.action.ts:L18-L18	32 (1)
FindStores	class	feature-libs/storefinder/core/store/actions/find-stores.action.ts:L33-L48	32 (3)
FindStoresOnHold	class	feature-libs/storefinder/core/store/actions/find-stores.action.ts:L25-L31	32 (1)
FocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L149-L149	18 (3)
FocusDirective	class	core-libs/storefront/layout/a11y/keyboard-focus/focus.directive.ts:L13-L25	18 (6)
FocusFirstInvalidFieldDirective	class	core-libs/storefront/layout/a11y/focus-first-invalid-field/focus-first-invalid-field.directive.ts:L34-L53	18 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
FooterNavigationComponent	class	core-libs/storefront/cms-components/navigation/footer-navigation/footer-navigation.component.ts:L23-L36	45 (2)
ForbiddenHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/forbidden/forbidden.handler.ts:L20-L50	41 (6)
ForgotPasswordComponent	class	feature-libs/user/profile/components/forgot-password/forgot-password.component.ts:L45-L62	34 (1)
FormComponent	class	feature-libs/organization/administration/components/shared/form/form.component.ts:L50-L143	28 (6)
FormConfig	class	core-libs/storefront/shared/config/form-config.ts:L14-L18	45 (3)
FormErrorsComponent	class	core-libs/storefront/shared/components/form/form-errors/form-errors.component.ts:L39-L143	45 (1)
FormModule	class	feature-libs/organization/administration/components/shared/form/form.module.ts:L32-L32	28 (1)
FormService	class	feature-libs/organization/administration/components/shared/form/form.service.ts:L12-L58	28 (2)
FutureStockFacade	class	feature-libs/product/future-stock/root/facade/future-stock.facade.ts:L24-L29	24 (1)
FutureStockService	class	feature-libs/product/future-stock/core/services/future-stock.service.ts:L20-L48	24 (1)

G

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
GatewayTimeoutHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/gateway/gateway-timeout.handler.ts:L16-L29	41 (1), 54 (1)
generateMappings()	function	core-libs/schematics/src/shared/schematics-config-mappings.ts:L262-L267	43 (1)
GenericLinkComponentService	class	core-libs/storefront/shared/components/generic-link/generic-link-component.service.ts:L12-L39	45 (4)
GeoPoint	interface	core-libs/core/src/model/misc.model.ts:L23-L26 (+1)	46 (1)
getAddressesLoading()	const	core-libs/core/src/user/store/selectors/user-addresses.selectors.ts:L24-L27	23 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
GetConfigurationOverview	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L383-L389	30 (1)
getCostCenterState()	const	feature-libs/organization/administration/core/store/selectors/cost-center.selector.ts:L92-L102	26 (1)
getNextPageContext()	const	core-libs/core/src/routing/store/selectors/routing.selector.ts:L34-L41	14 (2), 15 (1)
getOrderDetails()	const	feature-libs/order/core/store/selectors/order-details.selectors.ts:L18-L21 (+1)	27 (2)
GetOrderEntriesContext	interface	feature-libs/cart/base/root/context/get-order-entries.context.ts:L13-L20	25 (1)
getReducers()	function	feature-libs/asm/core/store/reducers/index.ts:L24-L34 (+16)	15 (1), 27 (1)
getRouterState()	const	core-libs/core/src/routing/store/selectors/routing.selector.ts:L18-L19	28 (1)
getServerRequestProviders()	function	core-libs/setup/ssr/providers/ssr-providers.ts:L53-L66	07 (4), 39 (1), 46 (2), 47 (1)
GetSubscriptionByCodeReloadEvent	class	feature-libs/subscription-billing/root/events/subscription-billing.events.ts:L9-L11	31 (2)
GetTicketQueryReloadEvent	class	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L13-L15	31 (2)
GigyaRaasComponent	class	integration-libs/cdc/components/gigya-raas/gigya-raas.component.ts:L35-L148	34 (6)
GigyaRaasComponentData	interface	integration-libs/cdc/core/models/cms.model.ts:L9-L42	34 (1)
GigyaRaasGuard	class	integration-libs/cdc/components/gigya-raas/gigya-raas.guard.ts:L32-L123	34 (2)
GigyaRaasModule	class	integration-libs/cdc/components/gigya-raas/gigya-raas.module.ts:L35-L35	34 (1)
GIVE_USER_CONSENT	const	core-libs/core/src/user/store/actions/user-consents.action.ts:L22-L22	23 (1)
GiveUserConsent	class	core-libs/core/src/user/store/actions/user-consents.action.ts:L71-L83	19 (1)
GiveUserConsentFail	class	core-libs/core/src/user/store/actions/user-consents.action.ts:L85-L94	19 (1)
GiveUserConsentSuccess	class	core-libs/core/src/user/store/actions/user-consents.action.ts:L96-L102	19 (1)
GLOBAL_MESSAGE_FEATURE	const	core-libs/core/src/global-message/store/global-message-state.ts:L9-L9	41 (2)
GlobalMessage	interface	core-libs/core/src/global-message/models/global-message.model.ts:L17-L21	41 (1), 50 (2)
GlobalMessageComponent	class	core-libs/storefront/cms-components/misc/global-message/global-message.component.ts:L33-L51	18 (1), 41 (2), 45 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
GlobalMessageComponentModule	class	core-libs/storefront/cms-components/misc/global-message/global-message.module.ts:L17-L17	41 (2), 45 (1)
GlobalMessageConfig	class	core-libs/core/src/global-message/config/global-message-config.ts:L19-L27	41 (3)
GlobalMessageEffect	class	core-libs/core/src/global-message/store/effects/global-message.effect.ts:L30-L114	41 (2)
GlobalMessageModule	class	core-libs/core/src/global-message/global-message.module.ts:L26-L33	06 (1), 41 (2)
GlobalMessageService	class	core-libs/core/src/global-message/facade/global-message.service.ts:L24-L73	25 (1), 32 (1), 35 (7), 41 (2)
GlobalMessageStoreModule	class	core-libs/core/src/global-message/store/global-message-store.module.ts:L21-L21	41 (1)
GlobalMessageType	enum	core-libs/core/src/global-message/models/global-message.model.ts:L9-L15	41 (1), 45 (1), 50 (1)
GlobalMessageTypeConfig	type	core-libs/core/src/global-message/config/global-message-config.ts:L11-L13	41 (2)
GlobService	class	core-libs/core/src/util/glob.service.ts:L11-L33	05 (1), 46 (2)
GOOGLE_MAPS_DEVELOPMENT_KEY_CONFIG	const	feature-libs/storefinder/root/config/constants.ts:L7-L7	32 (2), 47 (1)
GoogleMapRendererService	class	feature-libs/storefinder/core/service/google-map-renderer.service.ts:L19-L292	32 (3)
GroupType	enum	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L295-L302 (+1)	30 (1)
GtmCollectorConfig	interface	feature-libs/tracking/tms/gtm/config/default-gtm.config.ts:L10-L12	33 (2)
GtmCollectorService	class	feature-libs/tracking/tms/gtm/services/gtm-collector.service.ts:L20-L63	33 (6)
GtmModule	class	feature-libs/tracking/tms/gtm/gtm.module.ts:L14-L14	33 (5), 47 (1)
GuardsComposer	class	core-libs/storefront/cms-structure/services/guards-composer.ts:L31-L94	05 (5), 14 (3), 16 (3), 24 (1), 28 (1), 46 (1)

H

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
HamburgerMenuComponent	class	core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.component.ts:L19-L29	10 (1), 15 (2), 18 (3)
HamburgerMenuModule	class	core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.module.ts:L25-L25	18 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
HamburgerMenuService	class	core-libs/storefront/layout/header/hamburger-menu/hamburger-menu.service.ts:L15-L36	18 (2)
HierarchicalAddressConfig	class	core-libs/core/src/user/config/hierarchical-address-config.ts:L14-L23	23 (2)
HighlightPipe	class	core-libs/storefront/cms-components/navigation/search-box/highlight.pipe.ts:L10-L31	45 (2)
HOME_PAGE_CONTEXT	const	core-libs/core/src/routing/models/page-context.model.ts:L15-L15	05 (1), 14 (2), 15 (1)
HomePageEvent	class	core-libs/storefront/events/home/home-page.events.ts:L12-L15	20 (8)
HomePageEventBuilder	class	core-libs/storefront/events/home/home-page-event.builder.ts:L17-L36	20 (4)
HomePageEventModule	class	core-libs/storefront/events/home/home-page-event.module.ts:L11-L15	20 (2)
HTTP_TIMEOUT_CONFIG	const	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L16-L18	06 (8), 21 (5), 50 (1)
HttpErrorHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/http-error.handler.ts:L15-L74	06 (4), 25 (3), 26 (1), 28 (5), 30 (3), 31 (5), 32 (6), 41 (7), 46 (1), 50 (11), 54 (4)
HttpErrorHandlerInterceptor	class	core-libs/core/src/error-handling/http-error-handler/http-error-handler.interceptor.ts:L41-L99	04 (1), 06 (11), 39 (1), 41 (20)
HttpErrorInterceptor	class	core-libs/core/src/global-message/http-interceptors/http-error.interceptor.ts:L23-L66	06 (8), 12 (2), 31 (1), 41 (11), 46 (3), 50 (1), 54 (3)
HttpErrorModel	class	core-libs/core/src/model/misc.model.ts:L100-L106	21 (3), 31 (1), 41 (5), 46 (9)
HttpModule	class	core-libs/core/src/http/http.module.ts:L13-L19	06 (1), 21 (1)
HttpParamsURIEncoder	class	core-libs/core/src/util/http-params-uri.encoder.ts:L16-L32	06 (6), 21 (2), 35 (1), 46 (2), 52 (1)
HttpResponseStatus	enum	core-libs/core/src/global-message/models/response-status.model.ts:L7-L17	41 (3), 50 (1)
HttpTimeoutConfig	interface	core-libs/core/src/http/http-timeout/http-timeout.config.ts:L23-L33	21 (1)
HttpTimeoutInterceptor	class	core-libs/core/src/http/http-timeout/http-timeout.interceptor.ts:L27-L125	06 (9), 21 (5), 41 (4)
HttpTimeoutModule	class	core-libs/core/src/http/http-timeout/http-timeout.module.ts:L23-L23	06 (1), 21 (1)

I

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
I18nConfig	class	core-libs/core/src/i18n/config/i18n-config.ts:L18-L92	11 (1), 37 (4), 53 (3)
I18nConfigInitializer	class	core-libs/core/src/i18n/config/i18n-config-initializer.ts:L15-L34	11 (4), 37 (4)
I18NEXT_HTTP_BACKEND_CLIENT	const	core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend-client.ts:L16-L44	06 (4), 37 (4)
I18NEXT_INSTANCE	const	core-libs/core/src/i18n/i18next/i18next-instance.ts:L19-L22	37 (5)
I18NEXT_LOGGER_PLUGIN	const	core-libs/core/src/i18n/i18next/i18next-plugins/i18next-logger-plugin.ts:L17-L40	37 (3), 54 (1)
I18nextBackendInitializer	class	core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.initializer.ts:L15-L27	37 (2)
I18nextBackendService	class	core-libs/core/src/i18n/i18next/i18next-backend/i18next-backend.service.ts:L21-L39	37 (2), 46 (2)
I18nextHttpBackendInitializer	class	core-libs/core/src/i18n/i18next/i18next-backend/i18next-http-backend.initializer.ts:L24-L92	37 (4)
I18nextInitializer	class	core-libs/core/src/i18n/i18next/i18next-initializer.ts:L21-L111	37 (3)
I18nextResourcesToBackendInitializer	class	core-libs/core/src/i18n/i18next/i18next-backend/i18next-resources-to-backend.initializer.ts:L15-L38	37 (4)
I18nextTranslationService	class	core-libs/core/src/i18n/i18next/i18next-translation.service.ts:L17-L124	37 (3)
I18nModule	class	core-libs/core/src/i18n/i18n.module.ts:L40-L57	37 (3), 40 (1)
I18nTestingModule	class	core-libs/core/src/i18n/testing/i18n-testing.module.ts:L20-L20	37 (4), 54 (1)
ICON_TYPE	enum	core-libs/storefront/cms-components/misc/icon/icon.model.ts:L11-L77	45 (5), 54 (2)
IconComponent	class	core-libs/storefront/cms-components/misc/icon/icon.component.ts:L47-L131	38 (1), 45 (2)
IconConfig	class	core-libs/storefront/cms-components/misc/icon/icon.model.ts:L83-L85	45 (3)
IconLoaderService	class	core-libs/storefront/cms-components/misc/icon/icon-loader.service.ts:L22-L171	45 (3)
IconModule	class	core-libs/storefront/cms-components/misc/icon/icon.module.ts:L23-L23	45 (3)
IconOptions	interface	core-libs/storefront/cms-components/misc/icon/icon.model.ts:L91-L121	45 (1)
IconResourceType	enum	core-libs/storefront/cms-components/misc/icon/icon.model.ts:L133-L149	45 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
IconTestingModule	class	core-libs/storefront/cms-components/misc/icon/testing/icon-testing.module.ts:L50-L50	45 (3)
Image	interface	core-libs/core/src/model/image.model.ts:L11-L34 (+3)	45 (1), 46 (1)
ImageGroup	interface	core-libs/core/src/model/image.model.ts:L41-L43	45 (1), 46 (2)
ImageLoadingStrategy	enum	core-libs/storefront/shared/components/media/media.model.ts:L94-L107	45 (1)
Images	interface	core-libs/core/src/model/image.model.ts:L7-L9	46 (2)
ImageType	enum	core-libs/core/src/model/image.model.ts:L36-L39 (+3)	46 (1)
ImportCsvFileService	class	core-libs/storefront/shared/services/file/csv/import-csv-file.service.ts:L17-L121	45 (1)
ImportExportConfig	class	feature-libs/cart/import-export/core/config/import-export-config.ts:L16-L22	25 (1)
ImportExportRootModule	class	feature-libs/cart/import-export/root/import-export-root.module.ts:L26-L26	25 (1)
ImportOrderEntriesComponent	class	feature-libs/cart/import-export/components/import-to-cart/import-entries/import-order-entries.component.ts:L32-L51	25 (1)
ImportProductsFromCsvService	class	feature-libs/cart/import-export/components/import-to-cart/import-products-from-csv.service.ts:L13-L29	25 (1)
IndexType	type	core-libs/core/src/cms/store/cms-state.ts:L64-L69	15 (1)
INJECTOR	const	core-libs/schematics/src/shared/constants.ts:L47-L47	01 (1), 03 (1)
InlineRootRenderStrategy	class	core-libs/storefront/layout/launch-dialog/services/inline-root-render.strategy.ts:L22-L70	18 (1)
InnerComponentsHostDirective	class	core-libs/storefront/cms-structure/page/component/inner-components-host.directive.ts:L29-L87	16 (3)
installPackageJsonDependencies()	function	core-libs/schematics/src/shared/utils/lib-utils.ts:L736-L741	08 (3), 43 (2)
IntendedPickupLocationFacade	class	feature-libs/pickup-in-store/root/facade/intended-pickup-location.facade.ts:L35-L74	32 (2)
InteractionState	interface	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L177-L187	30 (1)
InterceptorUtil	class	core-libs/core/src/occ/utils/interceptor-util.ts:L13-L47	06 (5), 21 (2), 22 (1)
InternalServerErrorHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/internal-server/internal-server.handler.ts:L16-L29	41 (1), 50 (2)
IntersectionOptions	interface	core-libs/storefront/layout/loading/intersection.model.ts:L9-L21	17 (1), 18 (1)
IntersectionService	class	core-libs/storefront/layout/loading/intersection.service.ts:L26-L132	17 (3), 18 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
InvalidTraceparentFormatError	class	core-libs/setup/ssr/logger/loggers/w3c-trace-context/errors/invalid-traceparent-format-error.ts:L10-L14	39 (1)
InvalidTraceparentLengthError	class	core-libs/setup/ssr/logger/loggers/w3c-trace-context/errors/invalid-traceparent-length-error.ts:L11-L17	39 (1)
InvoicesListComponent	class	feature-libs/pdf-invoices/components/invoices-list/invoices-list.component.ts:L63-L214	27 (2)
IS_GUEST_USER_CHECKOUT_KEY	const	core-libs/storefront/cms-components/user/utils/user-constants.ts:L7-L7	26 (1)
isSsr()	function	core-libs/core/src/state/utils/browser-storage.ts:L73-L75	41 (2), 50 (1)
ItemActiveDirective	class	feature-libs/organization/administration/components/shared/item-active.directive.ts:L16-L52	28 (1)
ItemExistsDirective	class	feature-libs/organization/administration/components/shared/item-exists.directive.ts:L16-L42	28 (1)
ItemService	class	feature-libs/organization/administration/components/shared/item.service.ts:L24-L132	28 (8), 34 (1)

J

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
JavaRegExpConverter	class	core-libs/core/src/util/java-reg-exp-converter/java-reg-exp-converter.ts:L11-L59	46 (4)
JSONLD_PRODUCT_BUILDER	const	core-libs/storefront/cms-structure/seo/structured-data/builders/tokens.ts:L21-L23	40 (3)
JsonLdBaseProductBuilder	class	core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-base-product.builder.ts:L19-L61	40 (1)
JsonLdBuilder	interface	core-libs/storefront/cms-structure/seo/structured-data/builders/schema.interface.ts:L13-L15	40 (1)
JsonLdBuilderModule	class	core-libs/storefront/cms-structure/seo/structured-data/builders/json-ld-builder.module.ts:L52-L52	40 (6), 47 (1)
JsonLdDirective	class	core-libs/storefront/cms-structure/seo/structured-data/json-ld.directive.ts:L22-L47	40 (4)
JsonLdProductOfferBuilder	class	core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-offer.builder.ts:L19-L37	40 (1)
JsonLdProductReviewBuilder	class	core-libs/storefront/cms-structure/seo/structured-data/builders/product/jsonld-product-review.builder.ts:L21-L83	40 (1)
JsonLdScriptFactory	class	core-libs/storefront/cms-structure/seo/structured-data/json-ld-script.factory.ts:L22-L89	40 (3)
JspIncludeCmsComponentConfig	interface	core-libs/core/src/cms/config/cms-config.ts:L33-L40	15 (1)

K

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
KeyboardFocusModule	class	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.module.ts:L29-L29	18 (3)
KeyboardFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/services/keyboard-focus.service.ts:L13-L13	18 (4)
KeyboardFocusTestingModule	class	core-libs/storefront/layout/a11y/keyboard-focus/focus-testing.module.ts:L19-L19	18 (3), 54 (1)

L

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
Language	interface	core-libs/core/src/model/misc.model.ts:L28-L33 (+1)	46 (2)
LANGUAGE_CHANGE	const	core-libs/core/src/site-context/store/actions/languages.action.ts:L15-L15	13 (4), 14 (2), 15 (3), 23 (3), 24 (1), 25 (1), 27 (1), 32 (2)
LANGUAGE_CONTEXT_ID	const	core-libs/core/src/site-context/providers/context-ids.ts:L7-L7	13 (1)
LANGUAGE_NORMALIZER	const	core-libs/core/src/site-context/connectors/converters.ts:L12-L14	13 (1)
LanguageChange	class	core-libs/core/src/site-context/store/actions/languages.action.ts:L42-L48	13 (4)
LanguageCurrencyComponent	class	core-libs/storefront/cms-components/misc/site-context-selector/language-currency.component.ts:L20-L22	45 (2)
LanguageInitializer	class	core-libs/core/src/site-context/services/language-initializer.ts:L20-L80	10 (1)
LanguageService	class	core-libs/core/src/site-context/facade/language.service.ts:L25-L96	06 (1), 13 (4), 37 (1)
LanguageSetEvent	class	core-libs/core/src/site-context/events/site-context.events.ts:L12-L18	13 (7), 23 (1), 26 (3), 30 (2)
LanguageStatePersistenceService	class	core-libs/core/src/site-context/services/language-state-persistence.service.ts:L15-L48	13 (1)
LAUNCH_CALLER	enum	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L93-L103	18 (6), 25 (3), 27 (1), 29 (2), 30 (2), 31 (4), 32 (1), 34 (1), 54 (6)
LaunchConfig	interface	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L9-L11	18 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
LaunchDialog	interface	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L22-L32	18 (1), 34 (1)
LaunchDialogModule	class	core-libs/storefront/layout/launch-dialog/launch-dialog.module.ts:L42-L49	18 (1)
LaunchDialogService	class	core-libs/storefront/layout/launch-dialog/services/launch-dialog.service.ts:L25-L186	18 (6), 27 (2), 29 (2), 30 (1), 32 (1), 34 (1), 46 (2), 54 (1)
LaunchInlineDialog	interface	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L52-L54	18 (1)
LaunchInlineRootDialog	interface	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L60-L62	18 (1)
LaunchOptions	type	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L13-L17	18 (2)
LaunchOutletDialog	interface	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L38-L47	17 (3), 18 (2)
LaunchRenderStrategy	class	core-libs/storefront/layout/launch-dialog/services/launch-render.strategy.ts:L24-L158	18 (3)
LaunchRoute	interface	core-libs/storefront/layout/launch-dialog/config/launch-config.ts:L67-L76	18 (1)
LayoutBreakPoints	type	core-libs/storefront/layout/config/layout-config.ts:L80-L82	16 (1), 18 (3), 51 (1)
LayoutConfig	class	core-libs/storefront/layout/config/layout-config.ts:L94-L125	10 (1), 16 (5), 17 (2), 18 (7), 26 (1), 27 (3), 30 (1), 32 (1), 50 (1), 51 (4)
layoutConfigFactory()	function	core-libs/storefront/recipes/config/layout-config.ts:L190-L198	10 (3), 18 (3), 50 (1), 51 (1)
LayoutModule	class	core-libs/storefront/layout/layout.module.ts:L25-L25	02 (1), 13 (1), 16 (1), 18 (3), 38 (3)
LayoutSections	type	core-libs/storefront/layout/config/layout-config.ts:L30-L34	18 (2)
LayoutSlotConfig	type	core-libs/storefront/layout/config/layout-config.ts:L59-L61	16 (1), 18 (3), 50 (1), 51 (2)
LazyComponentHandler	class	core-libs/storefront/cms-structure/page/component/handlers/lazy-component.handler.ts:L30-L82	03 (3), 04 (4), 07 (4), 12 (1), 16 (8), 51 (3)
LazyLoadingModule	class	core-libs/core/src/lazy-loading/lazy-loading.module.ts:L39-L53	10 (1), 12 (5), 54 (1)
LazyModulesService	class	core-libs/core/src/lazy-loading/lazy-modules.service.ts:L41-L246	01 (2), 02 (1), 03 (6), 12 (10), 17 (1), 20 (1), 46 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
LCP_PRESENCE	const	core-libs/storefront/shared/lcp-context/lcp-presence.token.ts:L26-L32	16 (6)
LcpCmsComponentsConfig	class	core-libs/storefront/cms-structure/cms-lcp-context/config/lcp-cms-components.config.ts:L18-L48	16 (2)
LcpContextDirective	class	core-libs/storefront/shared/lcp-context/lcp-context.directive.ts:L46-L84	16 (1)
LcpPresence	enum	core-libs/storefront/shared/lcp-context/lcp-presence.model.ts:L10-L13	16 (1)
LcpPresenceMappingService	class	core-libs/storefront/shared/lcp-context/lcp-presence-mapping.service.ts:L15-L22	16 (2)
LegacyOccCmsComponentAdapter	class	core-libs/core/src/occ/adapters/cms/legacy-occ-cms-component.adapter.ts:L29-L58	15 (3), 21 (4)
LibraryOptions	interface	core-libs/schematics/src/shared/utils/lib-utils.ts:L85-L110	08 (3), 43 (3)
LinkComponent	class	core-libs/storefront/cms-components/content/link/link.component.ts:L26-L54	45 (2)
LinkModule	class	core-libs/storefront/cms-components/content/link/link.module.ts:L25-L25	45 (1)
ListComponent	class	feature-libs/organization/administration/components/shared/list/list.component.ts:L75-L251	28 (9), 34 (1)
Listener	interface	integration-libs/epd-visualization/root/util/event-listener-utils.ts:L12-L16	04 (2)
ListModel	interface	core-libs/core/src/model/misc.model.ts:L60-L64	46 (1)
ListModule	class	feature-libs/organization/administration/components/shared/list/list.module.ts:L44-L44	28 (1)
ListService	class	feature-libs/organization/administration/components/shared/list/list.service.ts:L43-L243	28 (11), 34 (1)
LOAD_BASE_SITES_SUCCESS	const	core-libs/core/src/site-context/store/actions/base-site.action.ts:L17-L17	13 (1)
LOAD_CART	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L16-L16	25 (1)
LOAD_CART_SUCCESS	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L18-L18	25 (3)
LOAD_CARTS_SUCCESS	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L20-L20	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
LOAD_CDC_USER_TOKEN_FAIL	const	integration-libs/cdc/core/store/actions/cdc-user-token.action.ts:L11-L11	34 (1)
LOAD_PRODUCT	const	core-libs/core/src/product/store/actions/product.action.ts:L15-L15	19 (1), 21 (1), 24 (1)
LOAD_USER_ADDRESSES_FAIL	const	core-libs/core/src/user/store/actions/user-addresses.action.ts:L13-L13	23 (1)
LOAD_USER_ADDRESSES_SUCCESS	const	core-libs/core/src/user/store/actions/user-addresses.action.ts:L14-L14	23 (1)
LoadAnonymousConsentTemplates	class	core-libs/core/src/anonymous-consents/store/actions/anonymous-consents.action.ts:L42-L48	23 (1)
LoadAnonymousConsentTemplatesSuccess	class	core-libs/core/src/anonymous-consents/store/actions/anonymous-consents.action.ts:L50-L56	23 (1)
LoadBaseSite	class	core-libs/core/src/site-context/store/actions/base-site.action.ts:L22-L24	13 (1)
LoadBaseSites	class	core-libs/core/src/site-context/store/actions/base-site.action.ts:L41-L43	13 (1), 35 (1)
LoadBudgets	class	feature-libs/organization/administration/core/store/actions/budget.action.ts:L64-L75	28 (1)
LoadBudgetSuccess	class	feature-libs/organization/administration/core/store/actions/budget.action.ts:L51-L62	28 (1)
LoadCart	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L130-L135	25 (15)
LoadCartEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L162-L167	25 (3), 26 (3)
LoadCartFail	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L141-L149	25 (2)
LoadCartSuccess	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L155-L160	25 (3)
LoadCdcUserToken	class	integration-libs/cdc/core/store/actions/cdc-user-token.action.ts:L35-L39	34 (1)
LoadCdcUserTokenFail	class	integration-libs/cdc/core/store/actions/cdc-user-token.action.ts:L26-L33	34 (1)
LoadClientToken	class	core-libs/core/src/auth/client-auth/store/actions/client-token.action.ts:L16-L22	22 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
LoadCmsComponent	class	core-libs/core/src/cms/store/actions/components.action.ts:L19-L30	15 (4), 33 (1)
LoadCmsComponentFail	class	core-libs/core/src/cms/store/actions/components.action.ts:L32-L47	15 (2)
LoadCmsComponentSuccess	class	core-libs/core/src/cms/store/actions/components.action.ts:L49-L63	15 (2)
LoadCmsNavigationItems	class	core-libs/core/src/cms/store/actions/navigation-entry-item.action.ts:L17-L23	45 (1)
LoadCmsPageData	class	core-libs/core/src/cms/store/actions/page.action.ts:L18-L24	15 (4), 40 (1)
LoadCmsPageDataFail	class	core-libs/core/src/cms/store/actions/page.action.ts:L26-L35	15 (2)
LoadCmsPageDataSuccess	class	core-libs/core/src/cms/store/actions/page.action.ts:L37-L43	15 (2)
LoadDefaultPointOfService	const	feature-libs/pickup-in-store/core/store/actions/default-point-of-service-name.action.ts:L17-L19	32 (1)
LoadDeliveryCountriesFail	class	core-libs/core/src/user/store/actions/delivery-countries.action.ts:L25-L32	41 (1)
LoaderFailAction	class	core-libs/core/src/state/utils/loader/loader.action.ts:L74-L83	19 (1), 41 (1)
LoaderLoadAction	class	core-libs/core/src/state/utils/loader/loader.action.ts:L65-L72	30 (1)
LoaderResetAction	class	core-libs/core/src/state/utils/loader/loader.action.ts:L94-L101	19 (1), 23 (1)
LoaderState	interface	core-libs/core/src/state/utils/loader/loader-state.ts:L7-L12	19 (3), 30 (2), 32 (1)
LoadingScopeConfig	interface	core-libs/core/src/occ/config/loading-scopes-config.ts:L10-L23	21 (2), 24 (1), 52 (3)
LoadingScopes	interface	core-libs/core/src/occ/config/loading-scopes-config.ts:L29-L31	21 (3), 52 (3)
LoadingScopesConfig	interface	core-libs/core/src/occ/config/loading-scopes-config.ts:L25-L27	21 (1)
LoadingScopesService	class	core-libs/core/src/occ/services/loading-scopes.service.ts:L14-L78	21 (2), 24 (1)
LoadOrderById	class	feature-libs/order/core/store/actions/order-by-id.action.ts:L14-L19	27 (1)
LoadOrderByIdSuccess	class	feature-libs/order/core/store/actions/order-by-id.action.ts:L31-L36	27 (1)
LoadOrderDetailsSuccess	class	feature-libs/order/core/store/actions/order-details.action.ts:L46-L51 (+1)	27 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
LoadProduct	class	core-libs/core/src/product/store/actions/product.action.ts:L29-L38	19 (1), 24 (2), 52 (1)
LoadProductFail	class	core-libs/core/src/product/store/actions/product.action.ts:L40-L53	21 (3), 24 (2)
LoadProductReviews	class	core-libs/core/src/product/store/actions/product-reviews.action.ts:L21-L25	24 (1)
LoadProductSuccess	class	core-libs/core/src/product/store/actions/product.action.ts:L55-L64	21 (1)
LoadSavedCarts	class	feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L82-L92	25 (1)
LoadSavedCartsSuccess	class	feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L94-L104	25 (1)
LoadStatus	enum	feature-libs/organization/administration/core/model/organization-item-status.ts:L7-L10	28 (1)
LoadUserAddresses	class	core-libs/core/src/user/store/actions/user-addresses.action.ts:L28-L34	23 (1)
LoadUserAddressesEvent	class	core-libs/core/src/user/events/user.events.ts:L30-L32	23 (1), 26 (2)
LoadUserOrders	class	feature-libs/order/core/store/actions/orders.action.ts:L16-L30	27 (1)
LoadUserPaymentMethods	class	core-libs/core/src/user/store/actions/payment-methods.action.ts:L31-L37	23 (2)
LoadUserPaymentMethodsEvent	class	core-libs/core/src/user/events/user.events.ts:L34-L36	23 (1), 26 (2)
LoadWishListSuccess	class	feature-libs/cart/wish-list/core/store/actions/wish-list.action.ts:L70-L76	25 (1)
Locals	interface	core-libs/setup/ssr/optimized-engine/request-context.ts:L106-L110	39 (1)
LOCATION_INITIALIZED_MULTI	const	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L36-L38	01 (5), 05 (5), 10 (10), 11 (6), 14 (4), 22 (1), 35 (1)
LocationInitializer	type	core-libs/core/src/routing/location-initialized-multi/location-initialized-multi.ts:L22-L22	10 (1), 14 (1)
LockFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L142-L147	18 (1)
LockFocusDirective	class	core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.directive.ts:L33-L222	18 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
LockFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/lock/lock-focus.service.ts:L13-L13	18 (1)
LoggerService	class	core-libs/core/src/logger/logger.service.ts:L18-L39	07 (1), 11 (1), 12 (1), 20 (1), 33 (1), 37 (4), 39 (7), 41 (9), 46 (2), 47 (1)
LoggerTestingModule	class	core-libs/core/src/logger/testing/logger-testing.module.ts:L14-L14	41 (3)
LoggingErrorHandler	class	core-libs/core/src/error-handling/multi-error-handler/logging-error-handler.ts:L20-L26	41 (9)
LOGIN	const	core-libs/core/src/auth/user-auth/store/actions/login-logout.action.ts:L9-L9	14 (2), 15 (3), 24 (1)
Login	class	core-libs/core/src/auth/user-auth/store/actions/login-logout.action.ts:L12-L14	22 (1), 29 (4)
LOGIN_EVENTS	const	integration-libs/cds/src/profiletag/tokens/login-events.token.ts:L20-L22	34 (3)
LOGIN_FORM_SERIALIZER	const	feature-libs/user/account/core/connectors/converters.ts:L27-L29	23 (1)
LoginAsGuestGuard	class	feature-libs/user/account/components/guards/login-as-guest.guard.ts:L20-L46	34 (2)
LoginComponent	class	feature-libs/user/account/components/login/login.component.ts:L34-L68	10 (1), 15 (2), 23 (1)
LoginEvent	class	core-libs/core/src/auth/user-auth/events/user-auth.events.ts:L28-L33	20 (1), 22 (5), 23 (3), 26 (2), 31 (2), 35 (1)
LoginFormComponent	class	feature-libs/user/account/components/login-form/login-form.component.ts:L53-L70	07 (1), 23 (1)
LoginFormComponentService	class	feature-libs/user/account/components/login-form/login-form-component.service.ts:L43-L325	23 (2), 34 (2)
LoginGuard	class	core-libs/storefront/cms-components/user/login-route/login.guard.ts:L27-L66	23 (6), 34 (3)
LoginRouteModule	class	core-libs/storefront/cms-components/user/login-route/login-route.module.ts:L36-L36	02 (1), 10 (1), 23 (3), 43 (1), 47 (1)
LOGOUT	const	core-libs/core/src/auth/user-auth/store/actions/login-logout.action.ts:L10-L10 (+1)	14 (2), 15 (3), 22 (2), 23 (1), 24 (1), 46 (1)
Logout	class	core-libs/core/src/auth/user-auth/store/actions/login-logout.action.ts:L16-L18	29 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
LogoutConfig	class	core-libs/storefront/cms-components/user/logout/config/logout-config.ts:L14-L25	23 (1)
LogoutCustomerSupportAgent	class	feature-libs/asm/core/store/actions/logout-agent.action.ts:L15-L17	29 (1)
LogoutEvent	class	core-libs/core/src/auth/user-auth/events/user-auth.events.ts:L15-L20	20 (1), 22 (8), 23 (4), 26 (2), 30 (2), 50 (1)
LogoutGuard	class	core-libs/storefront/cms-components/user/logout/logout.guard.ts:L31-L111	22 (2), 23 (3), 34 (1)
LogoutModule	class	core-libs/storefront/cms-components/user/logout/logout.module.ts:L33-L33	02 (1), 10 (1), 23 (3), 43 (1), 47 (1)
LOOKUP_VISUALIZATIONS_RESPONSE_NORMALIZER	const	integration-libs/epd-visualization/core/connectors/visualization/converters.ts:L14-L16	36 (1)

M

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
MAX_OCC_INTEGER_VALUE	const	feature-libs/organization/administration/components/constants.ts:L9-L9	28 (1)
Media	interface	core-libs/storefront/shared/components/media/media.model.ts:L9-L47	45 (1)
MediaComponent	class	core-libs/storefront/shared/components/media/media.component.ts:L55-L223	44 (1), 45 (1), 46 (1)
MediaConfig	class	core-libs/storefront/shared/components/media/media.config.ts:L23-L77	10 (1), 45 (1)
MediaModule	class	core-libs/storefront/shared/components/media/media.module.ts:L29-L43	02 (1), 45 (1)
MediaPreconnectService	class	core-libs/storefront/cms-structure/seo/media-preconnect.service.ts:L15-L35	40 (3)
MediaService	class	core-libs/storefront/shared/components/media/media.service.ts:L33-L429	45 (2)
MediaSourcesPipe	class	core-libs/storefront/shared/components/media/media-sources.pipe.ts:L10-L28	45 (1)
MerchandisingCarouselComponent	class	integration-libs/cds/src/merchandising/cms-components/merchandising-carousel/merchandising-carousel.component.ts:L47-L144	34 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
MerchandisingModule	class	integration-libs/cds/src/merchandising/merchandising.module.ts:L21-L21	34 (2)
MerchandisingStrategyAdapter	class	integration-libs/cds/src/merchandising/connectors/strategy/merchandising-strategy.adapter.ts:L11-L21	34 (2)
MergeCart	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L186-L189	25 (2)
MergeCartAbort	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L226-L236	25 (3)
MergeCartAndIncrementProcessesCount	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L195-L200	25 (3)
MergeCartSuccess	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L210-L215	25 (1)
MergeCartSuccessEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L141-L157	26 (1)
MergingSubject	class	core-libs/core/src/event/utils/merging-subject.ts:L24-L141	20 (8), 50 (4), 54 (4)
MessageComponent	class	core-libs/storefront/cms-components/misc/message/message.component.ts:L26-L90 (+1)	28 (1), 45 (2)
MessageComponentModule	class	core-libs/storefront/cms-components/misc/message/message.module.ts:L25-L25	45 (1)
MessageData	class	feature-libs/organization/administration/components/shared/message/message.model.ts:L13-L53	28 (1)
MessageRenderService	class	feature-libs/organization/administration/components/shared/message/services/message-render.service.ts:L19-L39	28 (1)
MessageService	class	feature-libs/organization/administration/components/shared/message/services/message.service.ts:L15-L56	28 (8)
META_REDUCERS	const	core-libs/schematics/src/shared/constants.ts:L609-L609	07 (4), 13 (1), 19 (13), 25 (3)
MetaTagConfigModule	class	core-libs/core/src/occ/config/meta-tag-config.module.ts:L11-L18	21 (1)
MethodPropertyDeprecation	interface	core-libs/schematics/src/shared/utils/file-utils.ts:L93-L99	43 (1), 54 (1)
migrate()	function	core-libs/schematics/src/modernize-app-migrated-from-2211_32-to-2211_36/index.ts:L29-L48 (+35)	02 (1), 43 (3)
MINI_CART_FEATURE	const	feature-libs/cart/base/root/feature-name.ts:L10-L10	47 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
MiniCartComponent	class	feature-libs/cart/base/components/mini-cart/mini-cart.component.ts:L29-L43	12 (1), 25 (2), 47 (1)
MiniCartComponentService	class	feature-libs/cart/base/components/mini-cart/mini-cart-component.service.ts:L40-L177	25 (1)
MockDatePipe	class	core-libs/core/src/i18n/testing/mock-date.pipe.ts:L11-L22	37 (1)
MockFeatureDirective	class	core-libs/storefront/shared/test/mock-feature-directive.ts:L10-L23	42 (1), 44 (6)
MockFeatureDirectivesModule	class	core-libs/storefront/shared/test/mock-feature-directives.module.ts:L14-L14	44 (3)
MockFeatureLevelDirective	class	core-libs/storefront/shared/test/mock-feature-level-directive.ts:L10-L19	44 (3)
MockFeatureTogglesController	class	core-libs/core/src/features-config/feature-toggles/testing/mock-feature-toggles.ts:L48-L79	42 (4), 44 (3)
MockLoggerService	class	core-libs/core/src/logger/testing/mock-logger.service.ts:L12-L22	41 (3)
MockMediaComponent	class	core-libs/storefront/shared/test/mock-media.component.ts:L15-L26	44 (4)
MockOccEndpointsService	class	core-libs/core/src/occ/adapters/user/unit-test.helper.ts:L23-L36	21 (1)
MockTranslatePipe	class	core-libs/core/src/i18n/testing/mock-translate.pipe.ts:L12-L33	37 (1)
MockTranslationService	class	core-libs/core/src/i18n/testing/mock-translation.service.ts:L13-L29	37 (2)
MockUrlPipe	class	core-libs/core/src/routing/configurable-routes/url-translation/testing/mock-url.pipe.ts:L41-L104	14 (2)
MockWinRef	class	core-libs/storefront/shared/test/mock-window-ref.ts:L9-L23	44 (5)
Module	interface	core-libs/schematics/src/shared/utils/lib-utils.ts:L201-L212	43 (1)
MODULE_INITIALIZER	const	core-libs/core/src/lazy-loading/tokens.ts:L19-L20	01 (10), 02 (2), 03 (12), 10 (11), 12 (17), 17 (6), 25 (4), 27 (2), 29 (2), 50 (1), 51 (2), 53 (5)
ModuleInitializedEvent	class	core-libs/core/src/lazy-loading/events/module-initialized-event.ts:L16-L32	03 (5), 11 (1), 12 (8), 20 (2), 53 (1)
MOVE_FOCUS	enum	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L12-L15	18 (2)
MULTI_CART_DATA	const	feature-libs/cart/base/core/store/multi-cart-state.ts:L11-L11	25 (2), 30 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
MULTI_CART_FEATURE	const	feature-libs/cart/base/core/store/multi-cart-state.ts:L10-L10	25 (1)
MULTI_ERROR_HANDLER	const	core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L25-L27	07 (1), 39 (3), 41 (10), 47 (1), 54 (3)
MultiCartEffects	class	feature-libs/cart/base/core/store/effects/multi-cart.effect.ts:L19-L144	25 (3)
MultiCartEffectsService	class	feature-libs/cart/base/core/store/effects/multi-cart-effect.service.ts:L12-L35	25 (5)
MultiCartEventListener	class	feature-libs/cart/base/root/events/multi-cart-event.listener.ts:L16-L48	25 (2)
MultiCartFacade	class	feature-libs/cart/base/root/facade/multi-cart.facade.ts:L43-L255	25 (4)
MultiCartService	class	feature-libs/cart/base/core/facade/multi-cart.service.ts:L34-L410	25 (5)
MultiCartStatePersistenceService	class	feature-libs/cart/base/core/services/multi-cart-state-persistence.service.ts:L24-L76	19 (1)
MultiCartStoreModule	class	feature-libs/cart/base/core/store/multi-cart-store.module.ts:L32-L32	25 (2)
MultiErrorHandler	interface	core-libs/core/src/error-handling/multi-error-handler/multi-error-handler.ts:L16-L18	41 (4)
MyAccountV2ConsignmentTrackingComponent	class	feature-libs/order/components/order-details/my-account-v2/consignment-tracking/my-account-v2-consignment-tracking.component.ts:L19-L35	27 (2)
MyAccountV2DownloadInvoicesComponent	class	feature-libs/order/components/order-details/my-account-v2/download-invoices/my-account-v2-download-invoices.component.ts:L41-L70	27 (1)
MyAccountV2DownloadInvoicesEventListener	class	feature-libs/order/components/order-details/my-account-v2/download-invoices/my-account-v2-download-invoices-event.listener.ts:L17-L45	27 (1)
MyAccountV2OrderConsignmentsService	class	feature-libs/order/components/order-details/my-account-v2-order-consignments.service.ts:L18-L82	27 (1)
MyAccountV2OrderDetailsActionsComponent	class	feature-libs/order/components/order-details/my-account-v2/order-details-actions/my-account-v2-order-details-actions.component.ts:L19-L36	27 (2)
MyAccountV2OrderHistoryComponent	class	feature-libs/order/components/order-history/my-account-v2/my-account-v2-order-history.component.ts:L36-L53	27 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
MyAccountV2OrderHistoryService	class	feature-libs/order/core/facade/my-account-v2-order-history.service.ts:L37-L270	27 (1)
MyPreferredStoreComponent	class	feature-libs/pickup-in-store/components/container/my-preferred-store/my-preferred-store.component.ts:L70-L184	32 (3)

N

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
NativeSelectSpaceDirective	class	core-libs/storefront/layout/a11y/native-select-space/native-select-space.directive.ts:L26-L49	18 (3)
NAVIGATION_DETAIL_ENTITY	const	core-libs/core/src/cms/store/cms-state.ts:L13-L13	15 (1)
NavigationComponent	class	core-libs/storefront/cms-components/navigation/navigation/navigation.component.ts:L23-L40	45 (2), 51 (1)
NavigationEntryItemEffects	class	core-libs/core/src/cms/store/effects/navigation-entry-item.effect.ts:L19-L109	15 (2)
NavigationEvent	class	core-libs/storefront/events/navigation/navigation.event.ts:L13-L34	20 (13), 25 (1), 33 (1), 50 (7)
NavigationEventBuilder	class	core-libs/storefront/events/navigation/navigation-event.builder.ts:L26-L65	20 (3)
NavigationEventModule	class	core-libs/storefront/events/navigation/navigation-event.module.ts:L11-L15	20 (3), 50 (2)
NavigationNode	interface	core-libs/storefront/cms-components/navigation/navigation/navigation-node.model.ts:L7-L33	45 (2)
NavigationService	class	core-libs/storefront/cms-components/navigation/navigation/navigation.service.ts:L21-L224	45 (3)
NavigationUIComponent	class	core-libs/storefront/cms-components/navigation/navigation/navigation-ui.component.ts:L53-L408	45 (1)
NewMessageEvent	class	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L25-L27	31 (1)
NewSavedCartOrderEntriesContext	class	feature-libs/cart/saved-cart/components/page-context/saved-carts-page/new-saved-cart-order-entries.context.ts:L35-L107	25 (1)
NewSavedCartOrderEntriesContextToken	const	feature-libs/cart/saved-cart/root/tokens/context.ts:L13-L15	25 (1)
NgExpressEngine	type	core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L28-L30	39 (2)
NgExpressEngineDecorator	class	core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L36-L49	07 (1), 39 (2), 43 (1), 47 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
NgExpressEngineInstance	type	core-libs/setup/ssr/engine-decorator/ng-express-engine-decorator.ts:L22-L26	39 (2)
NgSelectA11yDirective	class	core-libs/storefront/shared/components/ng-select-a11y/ng-select-a11y.directive.ts:L41-L267	04 (1)
NgSetupOptions	type	core-libs/setup/ssr/engine/ng-express-engine.ts:L24-L28	07 (1)
NodeItem	interface	core-libs/core/src/cms/model/node-item.model.ts:L9-L11	15 (2)
NODES_RESPONSE_NORMALIZER	const	integration-libs/epd-visualization/core/connectors/scene/converters.ts:L14-L16	36 (1)
NotAuthGuard	class	core-libs/core/src/auth/user-auth/guards/not-auth.guard.ts:L21-L41	22 (4), 23 (1), 28 (1), 34 (1)
NotCheckoutAuthGuard	class	feature-libs/checkout/base/components/guards/not-checkout-auth.guard.ts:L21-L45	26 (3)
NotFoundHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/not-found/not-found.handler.ts:L15-L25	41 (2)
NotFoundTicketRequestHandler	class	feature-libs/customer-ticketing/core/http-interceptors/handlers/not-found-ticket-request.handler.ts:L23-L75	31 (3)

0

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OAUTH_REDIRECT_FLOW_KEY	const	core-libs/core/src/auth/user-auth/utils/oauth-constants.ts:L7-L7	22 (1)
OAuthCallbackGuard	class	core-libs/core/src/auth/user-auth/guards/oauth-callback.guard.ts:L25-L53	22 (4)
OAuthFlow	enum	core-libs/core/src/auth/user-auth/models/oauth-flow.ts:L10-L23	22 (1)
OAuthLibWrapperService	class	core-libs/core/src/auth/user-auth/services/oauth-lib-wrapper.service.ts:L32-L308	11 (1), 22 (4)
ObjectComparisonUtils	class	core-libs/core/src/util/object-comparison-utils.ts:L7-L93	46 (1)
OCC_CART_ID_CURRENT	const	core-libs/core/src/occ/utils/occ-constants.ts:L7-L7	52 (1)
OCC_HTTP_TOKEN	const	core-libs/core/src/occ/utils/occ-http-token.ts:L20-L20	06 (7), 21 (3), 29 (1), 50 (1)
OCC_USER_ID_CONSTANTS	const	core-libs/core/src/occ/utils/index.ts:L19-L24	21 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OCC_USER_ID_CURRENT	const	core-libs/core/src/occ/utils/occ-user-ids.ts:L7-L7	30 (1), 36 (1)
OccAddressListNormalizer	class	core-libs/core/src/occ/adapters/user/converters/occ-address-list-normalizer.ts:L20-L39	21 (1)
OccAsmAdapter	class	feature-libs/asm/occ/adapters/occ-asm.adapter.ts:L47-L240	29 (4)
OccAsmCustomer360Adapter	class	feature-libs/asm/customer-360/occ/adapters/occ-asm-customer-360.adapter.ts:L30-L90	29 (1)
OccBackendNotification	class	integration-libs/cds/src/profiletag/adapters/occ-backend-notification-adapter.ts:L15-L36	34 (1)
OccBudgetAdapter	class	feature-libs/organization/administration/occ/adapters/occ-budget.adapter.ts:L26-L87	28 (2)
OccCancelServiceOrderAdapter	class	integration-libs/s4-service/order/occ/adapters/occ-cancel-service-order.adapter.ts:L16-L33	36 (1)
OccCartAccessCodeAdapter	class	feature-libs/cart/base/occ/adapters/occ-cart-access-code.adapter.ts:L24-L58	25 (1), 46 (1)
OccCartAdapter	class	feature-libs/cart/base/occ/adapters/occ-cart.adapter.ts:L28-L146	25 (3), 52 (5)
OccCartEntryAdapter	class	feature-libs/cart/base/occ/adapters/occ-cart-entry.adapter.ts:L18-L126	25 (3)
OccCartGuestUserAdapter	class	feature-libs/cart/base/occ/adapters/occ-cart-guest-user.adapter.ts:L29-L99	25 (1)
OccCartNormalizer	class	feature-libs/cart/base/occ/adapters/converters/occ-cart-normalizer.ts:L20-L87	25 (2), 52 (7), 53 (1)
OccCartValidationAdapter	class	feature-libs/cart/base/occ/adapters/occ-cart-validation.adapter.ts:L24-L45	25 (1)
OccCartVoucherAdapter	class	feature-libs/cart/base/occ/adapters/occ-cart-voucher.adapter.ts:L25-L125	25 (3)
OccCheckoutAdapter	class	feature-libs/checkout/base/occ/adapters/occ-checkout.adapter.ts:L26-L63	26 (1)
OccCheckoutDeliveryAddressAdapter	class	feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-address.adapter.ts:L26-L133	26 (1)
OccCheckoutDeliveryModesAdapter	class	feature-libs/checkout/base/occ/adapters/occ-checkout-delivery-modes.adapter.ts:L27-L117	26 (1)
OccCheckoutServiceDetailsAdapter	class	integration-libs/s4-service/checkout/occ/adapter/occ-checkout-service-details.adapter.ts:L21-L45	36 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OccCmsComponentAdapter	class	core-libs/core/src/occ/adapters/cms/occ-cms-component.adapter.ts:L23-L153	15 (1), 21 (2)
OccCmsPageAdapter	class	core-libs/core/src/occ/adapters/cms/occ-cms-page.adapter.ts:L34-L103	15 (1), 21 (2)
OccCmsPageNormalizer	class	core-libs/core/src/occ/adapters/cms/converters/occ-cms-page-normalizer.ts:L23-L223	15 (4), 21 (2)
OccConfig	class	core-libs/core/src/occ/config/occ-config.ts:L46-L48	15 (1), 21 (9), 24 (1), 32 (1), 52 (6), 53 (2)
OccConfiguratorTextfieldAdapter	class	feature-libs/product-configurator/textfield/occ/occ-configurator-textfield.adapter.ts:L27-L158	30 (1)
OccCostCenterSerializer	class	core-libs/core/src/occ/adapters/cost-center/converters/occ-cost-center-serializer.ts:L15-L26	21 (1), 52 (1)
OccDpRequestNormalizer	class	integration-libs/digital-payments/src/checkout/adapters/occ-digital-payment-request.normalizer.ts:L15-L34	35 (1)
OccEndpoint	interface	core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L9-L12	21 (2), 52 (3)
OccEndpoints	interface	core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L21-L299	10 (2), 21 (6), 23 (1), 25 (2), 26 (1), 27 (1), 31 (2), 32 (1), 36 (2), 52 (7), 54 (2)
OccEndpointsService	class	core-libs/core/src/occ/services/occ-endpoints.service.ts:L34-L249	06 (1), 13 (2), 21 (7), 46 (1), 50 (2), 52 (5)
OccFieldsService	class	core-libs/core/src/occ/services/occ-fields.service.ts:L45-L134	21 (1)
OccHttpContext	interface	core-libs/core/src/occ/utils/occ-http-token.ts:L9-L18	06 (2), 21 (1)
OccHttpErrorReason	enum	core-libs/core/src/util/occ-http-error-constants.ts:L14-L16	21 (2), 46 (2), 54 (1)
OccHttpErrorType	enum	core-libs/core/src/util/occ-http-error-constants.ts:L9-L12	21 (2), 46 (3), 54 (1)
OccOmfOrderHistoryAdapter	class	integration-libs/omf/order/occ-omf-order-history.adapter.ts:L26-L87	36 (4)
OccOpfTokenisationUserPaymentAdapter	class	integration-libs/opf/tokenisation/root/adapters/occ-opf-tokenisation-user-payment.adapter.ts:L27-L128	35 (2)
OccOppsCouponCodesInterceptor	class	integration-libs/opps/root/coupon-codes/http-interceptors/occ-opps-coupon-codes.interceptor.ts:L20-L65	06 (4), 35 (2)
OccOrderAdapter	class	feature-libs/order/occ/adapters/occ-order.adapter.ts:L29-L127	46 (2)
OccOrderHistoryAdapter	class	feature-libs/order/occ/adapters/occ-order-history.adapter.ts:L46-L232	36 (1), 52 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OccOrderNormalizer	class	feature-libs/order/occ/adapters/converters/occ-order-normalizer.ts:L22-L82	27 (2), 52 (1)
OccOrgUnitAdapter	class	feature-libs/organization/administration/occ/adapters/occ-org-unit.adapter.ts:L36-L309	52 (1)
OccPersonalizationIdInterceptor	class	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-id.interceptor.ts:L21-L99	06 (2), 33 (2)
OccPersonalizationTimeInterceptor	class	feature-libs/tracking/personalization/root/http-interceptors/occ-personalization-time.interceptor.ts:L21-L99	06 (2), 33 (2)
OccProductAdapter	class	core-libs/core/src/occ/adapters/product/occ-product.adapter.ts:L21-L60	21 (3), 24 (1), 52 (8)
OccProductAvailabilityAdapter	class	core-libs/core/src/occ/adapters/product/occ-product-availability-adapter.ts:L15-L46	21 (1)
OccProductReferencesAdapter	class	core-libs/core/src/occ/adapters/product/occ-product-references.adapter.ts:L17-L44	21 (1)
OccProductReferencesListNormalizer	class	core-libs/core/src/occ/adapters/product/converters/occ-product-references-list-normalizer.ts:L17-L40	21 (1), 24 (1)
OccProductSearchPageNormalizer	class	core-libs/core/src/occ/adapters/product/converters/occ-product-search-page-normalizer.ts:L20-L103	21 (1), 24 (2)
OccQuote	interface	feature-libs/quote/root/model/quote.model.ts:L11-L39	31 (1)
OccQuoteActionNormalizer	class	feature-libs/quote/occ/converters/occ-quote-action-normalizer.ts:L21-L91	31 (2)
OccQuoteEntryNormalizer	class	feature-libs/quote/occ/converters/occ-quote-entry-normalizer.ts:L16-L36	31 (3), 52 (1)
OccReorderOrderAdapter	class	feature-libs/order/occ/adapters/occ-reorder-order.adapter.ts:L22-L52	27 (2)
OccReorderOrderNormalizer	class	feature-libs/order/occ/adapters/converters/occ-reorder-order-normalizer.ts:L12-L21	27 (1)
OccReplenishmentOrderNormalizer	class	feature-libs/order/occ/adapters/converters/occ-replenishment-order-normalizer.ts:L18-L44	52 (1)
OccRequestedDeliveryDateAdapter	class	feature-libs/requested-delivery-date/occ/adapters/occ-requested-delivery-date.adapter.ts:L17-L45	32 (1)
OccRequestsOptimizerService	class	core-libs/core/src/occ/services/occ-requests-optimizer.service.ts:L22-L80	21 (2), 24 (1), 52 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OccReturnRequestNormalizer	class	feature-libs/order/occ/adapters/converters/occ-return-request-normalizer.ts:L18-L44	52 (1)
OccSavedCartAdapter	class	feature-libs/cart/saved-cart/occ/adapters/occ-saved-cart.adapter.ts:L20-L103	25 (2)
OccSegmentRefsInterceptor	class	integration-libs/segment-refs/root/http-interceptors/occ-segment-refs.interceptor.ts:L19-L87	06 (4), 34 (3), 50 (2)
OccSiteAdapter	class	core-libs/core/src/occ/adapters/site-context/occ-site.adapter.ts:L34-L141	13 (3), 21 (3)
OccUserAccountAdapter	class	feature-libs/user/account/occ/adapters/occ-user-account.adapter.ts:L35-L80	23 (1)
OccUserConsentAdapter	class	core-libs/core/src/occ/adapters/user/occ-user-consent.adapter.ts:L21-L79	34 (1)
OccUserCostCenterAdapter	class	core-libs/core/src/occ/adapters/user/occ-user-cost-centers.adapter.ts:L22-L56	21 (1)
OccUserProfileAdapter	class	feature-libs/user/profile/occ/adapters/occ-user-profile.adapter.ts:L38-L204	23 (1)
OMF_ORDER_MODULE	const	core-libs/schematics/src/shared/lib-configs/integration-libs/omf-schematics-config.ts:L20-L20	36 (1)
OmfConfig	class	integration-libs/omf/order/config/omf-config.ts:L15-L17	36 (5), 50 (1)
OmfConfiguration	class	integration-libs/omf/root/model/omf-configuration.ts:L7-L9	36 (3)
OmfOrderHistoryService	class	integration-libs/omf/order/omf-order-history.service.ts:L13-L24	36 (7)
OmfOrderModule	class	integration-libs/omf/order/omf-order.module.ts:L26-L26	36 (8), 47 (2), 50 (3)
OmfRootModule	class	integration-libs/omf/root/omf-root.module.ts:L10-L10	36 (2), 47 (1)
OneTimePasswordLoginFormComponent	class	feature-libs/user/account/components/otp-login-form/otp-login-form.component.ts:L69-L189	23 (1)
OneTimePasswordLoginFormModeule	class	feature-libs/user/account/components/otp-login-form/otp-login-form.module.ts:L55-L55	23 (1)
OnNavigateConfig	class	core-libs/storefront/router/config/on-navigate-config.ts:L14-L27	05 (1), 14 (2)
OnNavigateService	class	core-libs/storefront/router/on-navigate.service.ts:L30-L211	05 (5), 10 (1), 14 (3)
OpeningSchedule	interface	core-libs/core/src/model/point-of-service.model.ts:L44-L49 (+1)	46 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OPF_ACTIVE_CONFIGURATIONS_NORMALIZER	const	integration-libs/opf/base/core/tokens/tokens.ts:L11-L13	35 (2)
OPF_BASE_FEATURE	const	integration-libs/opf/base/root/feature-name.ts:L7-L7	35 (1)
OPF_CTA_SCRIPTS_NORMALIZER	const	integration-libs/opf/cta/core/tokens/tokens.ts:L11-L13	35 (1)
OPF_PAYMENT_CONFIG_SERIALIZER	const	integration-libs/opf/payment/core/connectors/converters.ts:L11-L13	35 (2)
OPF_QUICK_BUY_FEATURE	const	integration-libs/opf/quick-buy/root/feature-name.ts:L7-L7	35 (1)
OpfActiveConfiguration	interface	integration-libs/opf/base/root/model/opf-base.model.ts:L12-L23	35 (1)
OpfApiBaseAdapter	class	integration-libs/opf/base/opf-api/adapters/opf-api-base.adapter.ts:L36-L119	35 (1)
OpfApiConfig	class	integration-libs/opf/base/root/config/opf-api-config.ts:L15-L17	35 (2)
OpfApiCtaAdapter	class	integration-libs/opf/cta/opf-api/adapters/opf-api-cta.adapter.ts:L35-L89	35 (1)
OpfApiEndpoint	interface	integration-libs/opf/base/root/model/opf-api-endpoint.model.ts:L7-L10	35 (1)
OpfApiEndpoints	interface	integration-libs/opf/base/root/model/opf-api-endpoints.model.ts:L7-L7	35 (3)
OpfApiPaymentAdapter	class	integration-libs/opf/payment/opf-api/adapters/opf-api-payment.adapter.ts:L52-L309	35 (1)
OpfApiQuickBuyAdapter	class	integration-libs/opf/quick-buy/opf-api/adapters/opf-api-quick-buy.adapter.ts:L35-L90	35 (1)
OpfB2bCheckoutPaymentTypeService	class	integration-libs/opf/b2b-checkout/core/facade/opf-b2b-checkout-payment-type.service.ts:L32-L153	35 (2)
OpfB2bCheckoutRootModule	class	integration-libs/opf/b2b-checkout/root/opf-b2b-checkout-root.module.ts:L48-L48	35 (3)
OpfB2bCheckoutStepsSetGuard	class	integration-libs/opf/b2b-checkout/root/guards/opf-b2b-checkout-steps-set.guard.ts:L42-L193	35 (3)
OpfBaseAdapter	class	integration-libs/opf/base/core/connectors/opf-base.adapter.ts:L13-L20	35 (1)
OpfBaseFacade	class	integration-libs/opf/base/root/facade/opf-base.facade.ts:L25-L32	35 (1)
OpfBaseModule	class	integration-libs/opf/base/opf-base.module.ts:L15-L15	35 (1)
OpfBaseRootModule	class	integration-libs/opf/base/root/opf-base-root.module.ts:L32-L32	35 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OpfCheckoutAuthGuard	class	integration-libs/opf/checkout/root/checkout-guard/opf-checkout-auth.guard.ts:L21-L63	35 (2)
OpfCheckoutOutlets	enum	integration-libs/opf/checkout/root/model/opf-checkout-outlets.model.ts:L7-L11	35 (2)
OpfCheckoutPaymentAndReviewComponent	class	integration-libs/opf/checkout/components/opf-checkout-payment-and-review/opf-checkout-payment-and-review.component.ts:L74-L167	35 (3)
OpfCheckoutPaymentsComponent	class	integration-libs/opf/checkout/components/opf-checkout-payments/opf-checkout-payments.component.ts:L75-L394	35 (2)
OpfCheckoutRootModule	class	integration-libs/opf/checkout/root/opf-checkout-root.module.ts:L63-L63	35 (4)
OpfConfig	class	integration-libs/opf/base/root/config/opf-config.ts:L14-L90	35 (2)
OpfCtaAdapter	class	integration-libs/opf/cta/core/connectors/opf-cta.adapter.ts:L13-L20	35 (1)
OpfCtaComponentsModule	class	integration-libs/opf/cta/components/opf-cta-components.module.ts:L25-L25	35 (1)
OpfCtaElementComponent	class	integration-libs/opf/cta/components/opf-cta-element/opf-cta-element.component.ts:L26-L51	35 (1)
OpfCtaFacade	class	integration-libs/opf/cta/root/facade/opf-cta.facade.ts:L26-L35	35 (1)
OpfCtaRootModule	class	integration-libs/opf/cta/root/opf-cta-root.module.ts:L25-L25	35 (2)
OpfCtaScriptsComponent	class	integration-libs/opf/cta/components/opf-cta-scripts/opf-cta-scripts.component.ts:L20-L28	35 (1)
OpfCtaScriptsService	class	integration-libs/opf/cta/components/opf-cta-scripts/opf-cta-scripts.service.ts:L33-L265	35 (1)
OpfDynamicCtaService	class	integration-libs/opf/cta/core/services/opf-dynamic-cta.service.ts:L47-L252	35 (1)
OpfEndpointsService	class	integration-libs/opf/base/core/services/opf-endpoints.service.ts:L21-L115	35 (1)
OpfEventListenerService	class	integration-libs/opf/base/root/events/opf-event.listener.ts:L17-L40	35 (1)
OpfEventModule	class	integration-libs/opf/base/root/events/opf-event.module.ts:L11-L15	35 (1)
OpfGiftCardCartOccNormalizer	class	integration-libs/opf/gift-card/root/normalizers/opf-gift-card-cart-occ-normalizer.ts:L21-L74	35 (1), 52 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OpfGiftCardCoreModule	class	integration-libs/opf/gift-card/core/opf-gift-card-core.module.ts:L14-L14	52 (1)
OpfGiftCardModule	class	integration-libs/opf/gift-card/opf-gift-card.module.ts:L21-L21	06 (2), 35 (1)
OpfGiftCardOrderOccNormalizer	class	integration-libs/opf/gift-card/root/normalizers/opf-gift-card-order-occ-normalizer.ts:L20-L44	35 (1)
OpfGiftCardPaymentApiInterceptor	class	integration-libs/opf/gift-card/root/http-interceptors/opf-gift-card-payment-api.interceptor.ts:L33-L66	06 (4), 35 (1)
OpfGiftCardRootModule	class	integration-libs/opf/gift-card/root/opf-gift-card-root.module.ts:L65-L65	06 (2), 35 (1), 52 (1)
OpfGlobalFunctionsDomain	enum	integration-libs/opf/global-functions/root/model/opf-global-functions.model.ts:L10-L14	35 (1)
OpfGlobalFunctionsFacade	class	integration-libs/opf/global-functions/root/facade/opf-global-functions.facade.ts:L24-L40	35 (1)
OpfGlobalFunctionsRootModule	class	integration-libs/opf/global-functions/root/opf-global-functions-root.module.ts:L14-L14	35 (1)
OpfGlobalMessageService	class	integration-libs/opf/base/root/services/opf-global-message.service.ts:L19-L61	35 (2)
OpfMetadataStoreService	class	integration-libs/opf/base/root/services/opf-metadata-store.service.ts:L12-L39	35 (1)
OpfOccOrderNormalizer	class	integration-libs/opf/order/opf-occ-order-normalizer.ts:L13-L28	35 (2)
OpfOrderModule	class	integration-libs/opf/order/opf-order.module.ts:L20-L20	35 (2), 47 (1)
OpfPaymentAdapter	class	integration-libs/opf/payment/core/connectors/opf-payment.adapter.ts:L20-L77	35 (1)
OpfPaymentErrorHandlerService	class	integration-libs/opf/payment/core/services/opf-payment-error-handler.service.ts:L23-L78	35 (1)
OpfPaymentFacade	class	integration-libs/opf/payment/root/facade/opf-payment.facade.ts:L39-L119	35 (2)
OpfPaymentHostedFieldsService	class	integration-libs/opf/payment/core/services/opf-payment-hosted-fields.service.ts:L48-L258	35 (1)
OpfPaymentProviderType	enum	integration-libs/opf/base/root/model/opf-base.model.ts:L84-L88	35 (1)
OpfPaymentRootModule	class	integration-libs/opf/payment/root/opf-payment-root.module.ts:L49-L49	35 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OpfPaymentService	class	integration-libs/opf/payment/core/facade/opf-payment.service.ts:L27-L172	35 (1)
OpfPaymentVerificationComponent	class	integration-libs/opf/payment/root/components/opf-payment-verification/opf-payment-verification.component.ts:L30-L154	35 (1)
OpfQuickBuyCartAdapter	class	integration-libs/opf/quick-buy/core/connectors/opf-quick-buy-cart.adapter.ts:L11-L39	35 (1)
OpfQuickBuyConfig	class	integration-libs/opf/quick-buy/root/config/opf-quick-buy-config.ts:L18-L24	35 (1)
OpfQuickBuyModule	class	integration-libs/opf/quick-buy/opf-quick-buy.module.ts:L21-L21	35 (1)
OpfQuickBuyPickupInStoreModule	class	integration-libs/opf/quick-buy/pickup-in-store/opf-quick-buy-pickup-in-store.module.ts:L22-L22	35 (1)
OpfQuickBuyRootModule	class	integration-libs/opf/quick-buy/root/opf-quick-buy-root.module.ts:L33-L33	35 (1)
OpfQuickBuySingleProductCartOptionsFacade	class	integration-libs/opf/quick-buy/root/facade/opf-quick-buy-single-product-cart-options.facade.ts:L22-L26	35 (3)
OpfQuickBuyTransactionService	class	integration-libs/opf/quick-buy/core/services/opf-quick-buy-transaction.service.ts:L24-L198	35 (3)
OpfResourceLoaderService	class	integration-libs/opf/base/root/services/opf-resource-loader.service.ts:L33-L437	35 (1)
OpfTokenisationModule	class	integration-libs/opf/tokenisation/opf-tokenisation.module.ts:L18-L18	35 (1)
OpfTokenisationRootModule	class	integration-libs/opf/tokenisation/root/opf-tokenisation-root.module.ts:L58-L58	35 (2)
OppsConfig	class	integration-libs/opps/root/config/opps-config.ts:L14-L25	35 (1)
OppsCouponCodesModule	class	integration-libs/opps/root/coupon-codes/opps-coupon-codes.module.ts:L29-L29	35 (1)
OppsLoginRequiredGuard	class	integration-libs/opps/root/login-required/opps-login-required.guard.ts:L22-L62	35 (4)
OppsLoginRequiredModule	class	integration-libs/opps/root/login-required/opps-login-required.module.ts:L20-L20	35 (1)
OppsRootModule	class	integration-libs/opps/root/opps-root.module.ts:L17-L17	35 (3)
OptimizedSsrEngine	class	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L37-L475	07 (6), 39 (8), 47 (5)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
Order	interface	feature-libs/order/root/model/order.model.ts:L103-L146 (+1)	27 (2), 32 (3), 36 (3), 43 (1)
ORDER_CMS_COMPONENTS	const	feature-libs/order/root/order-root.module.ts:L28-L65	27 (2)
ORDER_CORE_FEATURE	const	feature-libs/order/root/feature-name.ts:L8-L8	50 (1)
ORDER_ENTRIES_CONTEXT	const	feature-libs/cart/base/root/context/order-entires.context.ts:L10-L10	14 (2), 25 (4), 28 (1)
ORDER_ENTRY_PROMOTIONS_NORMALIZER	const	feature-libs/cart/base/root/tokens/converters.ts:L21-L23	25 (3), 27 (1)
ORDER_FEATURE	const	feature-libs/order/root/feature-name.ts:L7-L7 (+1)	28 (1), 50 (2)
ORDER_HISTORY_NORMALIZER	const	feature-libs/order/root/tokens/converters.ts:L36-L38	27 (1)
ORDER_MODULE	const	core-libs/schematics/src/shared/lib-configs/order-schematics-config.ts:L24-L24	36 (2)
ORDER_NORMALIZER	const	feature-libs/order/root/tokens/converters.ts:L24-L26	27 (3), 35 (4), 36 (1), 52 (1)
ORDER_RETURN_REQUEST_INPUT_SERIALIZER	const	feature-libs/order/root/tokens/converters.ts:L48-L50	27 (2)
ORDER_RETURN_REQUEST_NORMALIZER	const	feature-libs/order/root/tokens/converters.ts:L44-L46	27 (1)
ORDER_RETURNS_NORMALIZER	const	feature-libs/order/root/tokens/converters.ts:L52-L54	27 (1)
ORDER_TYPE	enum	feature-libs/order/root/model/scheduled-replenishment.model.ts:L32-L35	27 (1)
OrderAdapter	class	feature-libs/order/core/connectors/order.adapter.ts:L10-L36	27 (1)
OrderAmendService	class	feature-libs/order/components/amend-order/amend-order.service.ts:L35-L153	27 (3)
OrderApprovalDetailService	class	feature-libs/organization/order-approval/components/details/order-approval-detail.service.ts:L18-L66	28 (2)
OrderApprovalService	class	feature-libs/organization/order-approval/core/services/order-approval.service.ts:L31-L181	28 (2)
OrderAttachmentsDialogComponent	class	feature-libs/order/components/order-details/order-attachments/attachments-dialog/order-attachments-dialog.component.ts:L55-L188	04 (1)
OrderAttachmentsFacade	class	feature-libs/order/root/facade/order-attachments.facade.ts:L22-L35	27 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OrderAttachmentsService	class	feature-libs/order/core/facade/order-attachments.service.ts:L17-L45	27 (1)
OrderCancellationGuard	class	feature-libs/order/components/amend-order/cancellations/order-cancellation.guard.ts:L17-L40	27 (1)
OrderCancellationService	class	feature-libs/order/components/amend-order/cancellations/order-cancellation.service.ts:L27-L94	27 (1)
OrderComponentsModule	class	feature-libs/order/components/order-components.module.ts:L43-L43	27 (1)
OrderConfig	class	feature-libs/order/root/config/order-config.ts:L14-L19	27 (2), 31 (1)
OrderConfirmationGuard	class	feature-libs/order/components/guards/order-confirmation.guard.ts:L17-L37	27 (4)
OrderConfirmationModule	class	feature-libs/order/components/order-confirmation/order-confirmation.module.ts:L143-L143	17 (1), 27 (2), 40 (3)
OrderConfirmationOrderEntriesContextToken	const	feature-libs/order/root/tokens/context.ts:L13-L15	27 (1)
OrderConfirmationShippingComponent	class	feature-libs/order/components/order-confirmation/order-confirmation-shipping/order-confirmation-shipping.component.ts:L59-L136	27 (1)
OrderConsignedEntriesComponent	class	feature-libs/order/components/order-details/order-detail-items/order-consigned-entries/order-consigned-entries.component.ts:L42-L69	42 (1)
OrderCoreModule	class	feature-libs/order/core/order-core.module.ts:L29-L29	27 (2), 50 (1)
OrderDetailBillingComponent	class	feature-libs/order/components/order-details/order-detail-billing/order-detail-billing.component.ts:L27-L66	27 (1)
OrderDetailItemsComponent	class	feature-libs/order/components/order-details/order-detail-items/order-detail-items.component.ts:L45-L105	27 (2)
OrderDetailReorderComponent	class	feature-libs/order/components/order-details/order-detail-reorder/order-detail-reorder.component.ts:L31-L81	27 (1)
OrderDetailsModule	class	feature-libs/order/components/order-details/order-details.module.ts:L191-L191	12 (1), 42 (1)
OrderDetailsOrderEntriesContextToken	const	feature-libs/order/root/tokens/context.ts:L9-L11	27 (1)
OrderDetailsService	class	feature-libs/order/components/order-details/order-details.service.ts:L22-L56	27 (12), 28 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OrderDocumentFlowDialogComponent	class	feature-libs/order/document-flow/components/order-document-flow-dialog/order-document-flow-dialog.component.ts:L68-L178	27 (1)
OrderDocumentFlowFacade	class	feature-libs/order/document-flow/root/facade/order-document-flow.facade.ts:L29-L45	27 (1)
OrderDocumentFlowService	class	feature-libs/order/document-flow/core/services/order-document-flow.service.ts:L19-L56	27 (1)
OrderEntriesSource	enum	feature-libs/cart/base/root/models/import-export.model.ts:L7-L15	25 (1)
OrderEntry	interface	core-libs/core/src/occ/occ-models/occ.models.ts:L904-L924 (+1)	25 (2), 30 (2), 31 (1), 32 (1), 36 (2), 53 (7), 54 (1)
OrderEntryPromotionsNormalizer	class	feature-libs/cart/base/occ/adapters/converters/order-entry-promotions-normalizer.ts:L15-L18	25 (2)
OrderEvent	class	feature-libs/order/root/events/order.events.ts:L14-L30	27 (1)
OrderFacade	class	feature-libs/order/root/facade/order.facade.ts:L31-L64	27 (6)
OrderFeatureModule	class	core-libs/schematics/src/shared/utils/config-utils_spec.ts:L120-L121	27 (1)
OrderGuestRegisterFormComponent	class	feature-libs/order/components/order-confirmation/order-guest-register-form/order-guest-register-form.component.ts:L44-L92	27 (1)
OrderHistory	interface	feature-libs/order/root/model/order.model.ts:L84-L95 (+1)	27 (1)
OrderHistoryAdapter	class	feature-libs/order/core/connectors/order-history.adapter.ts:L19-L112	27 (1), 36 (4), 52 (2)
OrderHistoryComponent	class	feature-libs/order/components/order-history/order-history.component.ts:L53-L156	27 (2)
OrderHistoryFacade	class	feature-libs/order/root/facade/order-history.facade.ts:L49-L149	27 (4), 36 (6), 50 (6)
OrderHistoryService	class	feature-libs/order/core/facade/order-history.service.ts:L30-L226	36 (2), 50 (2)
OrderModule	class	feature-libs/order/order.module.ts:L15-L15	27 (4), 35 (1), 36 (4), 47 (1), 50 (4), 52 (1)
OrderOccEndpoints	interface	feature-libs/order/occ/model/occ-order-endpoints.model.ts:L9-L102	27 (1)
OrderOccModule	class	feature-libs/order/occ/order-occ.module.ts:L90-L90	27 (1)
OrderOutlets	enum	feature-libs/order/root/model/order-outlets.model.ts:L10-L14	27 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OrderOverviewComponent	class	feature-libs/order/components/order-details/order-overview/order-overview.component.ts:L57-L303	27 (4)
OrderOverviewComponentService	class	feature-libs/order/components/order-details/order-overview/order-overview-component.service.ts:L11-L15	36 (2)
OrderOverviewDeliveryDateComponent	class	feature-libs/requested-delivery-date/root/components/order-overview-delivery-date/order-overview-delivery-date.component.ts:L21-L66	32 (2)
OrderPlacedEvent	class	feature-libs/order/root/events/order.events.ts:L35-L44	26 (5), 27 (2), 34 (1), 35 (1)
OrderReturnGuard	class	feature-libs/order/components/amend-order/returns/order-return.guard.ts:L17-L40	27 (1)
OrderReturnRequestFacade	class	feature-libs/order/root/facade/order-return-request.facade.ts:L45-L127	27 (2)
OrderReturnService	class	feature-libs/order/components/amend-order/returns/order-return.service.ts:L27-L100	27 (1)
OrderRootModule	class	feature-libs/order/root/order-root.module.ts:L180-L180	27 (5)
OrdersEffect	class	feature-libs/order/core/store/effects/orders.effect.ts:L24-L79	27 (2)
OrderService	class	feature-libs/order/core/facade/order.service.ts:L23-L191	26 (1), 27 (1)
OrderState	interface	feature-libs/order/core/store/order-state.ts:L37-L47	27 (1)
OrderStoreModule	class	feature-libs/order/core/store/order-store.module.ts:L21-L21	27 (1)
OrderSummaryComponent	class	feature-libs/cart/base/components/cart-shared/order-summary/order-summary.component.ts:L20-L39	17 (2), 51 (1)
ORGANIZATION_ADMINISTRATION_FEATURE	const	feature-libs/organization/administration/root/feature-name.ts:L7-L7	28 (2)
OrganizationBadRequestHandler	class	feature-libs/organization/administration/core/http-interceptors/bad-request/bad-request.handler.ts:L20-L94	28 (3)
OrganizationClearData	class	feature-libs/organization/administration/core/store/actions/organization.action.ts:L11-L13	28 (4)
OrganizationConflictHandler	class	feature-libs/organization/administration/core/http-interceptors/conflict/conflict.handler.ts:L20-L97	28 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OrganizationPageMetaResolver	class	feature-libs/organization/administration/core/services/organization-page-meta.resolver.ts:L41-L131	28 (2)
OrganizationsGuardsModule	class	feature-libs/organization/administration/core/guards/organization-guards.module.ts:L21-L21	28 (1)
OrganizationStoreModule	class	feature-libs/organization/administration/core/store/organization-store.module.ts:L23-L23	28 (1)
OrganizationTableType	enum	feature-libs/organization/administration/components/shared/organization.model.ts:L9-L35	28 (8)
OrganizationUIConfig	class	feature-libs/organization/administration/root/config/organization-ui.config.ts:L17-L30	28 (3)
OrgUnitEffects	class	feature-libs/organization/administration/core/store/effects/org-unit.effect.ts:L30-L476	54 (1)
OrgUnitGuard	class	feature-libs/organization/administration/core/guards/org-unit.guard.ts:L17-L38	28 (4)
OrgUnitService	class	feature-libs/organization/administration/core/services/org-unit.service.ts:L40-L504	34 (2)
OriginValidationOptions	interface	core-libs/setup/ssr/express-utils/express-origin-validation-middleware.ts:L10-L12	39 (1)
OutboundHttpError	class	core-libs/core/src/error-handling/http-error-handler/outbound-http-error.ts:L10-L14	06 (2), 39 (1), 41 (6)
OutletContextData	class	core-libs/storefront/cms-structure/outlet/outlet.model.ts:L21-L44	17 (9), 32 (1), 51 (5)
OutletDirective	class	core-libs/storefront/cms-structure/outlet/outlet.directive.ts:L34-L236	02 (1), 04 (1), 16 (3), 17 (3), 18 (3)
OutletModule	class	core-libs/storefront/cms-structure/outlet/outlet.module.ts:L53-L89	02 (8), 17 (2), 51 (2)
OutletPosition	enum	core-libs/storefront/cms-structure/outlet/outlet.model.ts:L9-L13	17 (5), 18 (1), 51 (6), 53 (2)
OutletRefDirective	class	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.directive.ts:L18-L36	17 (4), 51 (4)
OutletRefModule	class	core-libs/storefront/cms-structure/outlet/outlet-ref/outlet-ref.module.ts:L16-L16	17 (2), 18 (1)
OutletRendererService	class	core-libs/storefront/cms-structure/outlet/outlet-renderer.service.ts:L16-L49	17 (12), 51 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
OutletRenderStrategy	class	core-libs/storefront/layout/launch-dialog/services/outlet-render.strategy.ts:L27-L99	17 (4), 29 (1)
OutletService	class	core-libs/storefront/cms-structure/outlet/outlet.service.ts:L13-L98	17 (10), 18 (3), 24 (1), 51 (4)
Owner	interface	feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L11-L29	30 (1)
OwnerType	enum	feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L48-L54	30 (2)

P

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
Page	interface	core-libs/core/src/cms/model/page.model.ts:L10-L26	15 (4), 40 (2)
PAGE_LAYOUT_HANDLER	const	core-libs/storefront/cms-structure/page/page-layout/page-layout-handler.ts:L11-L13 (+1)	12 (1), 16 (3), 25 (2), 30 (4), 51 (2)
PageBreadcrumbResolver	interface	core-libs/core/src/cms/page/page.resolvers.ts:L53-L58	40 (1), 52 (1)
PageComponentModule	class	core-libs/storefront/cms-structure/page/component/page-component.module.ts:L23-L41	02 (1), 16 (1)
PageContext	class	core-libs/core/src/routing/models/page-context.model.ts:L22-L30	05 (2), 14 (8), 15 (4), 33 (1), 51 (7)
PageDescriptionResolver	interface	core-libs/core/src/cms/page/page.resolvers.ts:L42-L47	40 (1)
PageEffects	class	core-libs/core/src/cms/store/effects/page.effect.ts:L31-L115	13 (2), 15 (2)
PageEvent	class	core-libs/storefront/events/page/page.events.ts:L13-L18	20 (7)
PageHeadingResolver	interface	core-libs/core/src/cms/page/page.resolvers.ts:L20-L25	40 (2), 52 (1)
PageImageResolver	interface	core-libs/core/src/cms/page/page.resolvers.ts:L65-L70	40 (1)
PageLayoutComponent	class	core-libs/storefront/cms-structure/page/page-layout/page-layout.component.ts:L30-L58	05 (1), 10 (1), 15 (1), 16 (10), 17 (2), 18 (1), 25 (1), 28 (1), 29 (1), 30 (1), 36 (1), 51 (6), 53 (1)
PageLayoutHandler	interface	core-libs/storefront/cms-structure/page/page-layout/page-layout-handler.ts:L15-L22	16 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
PageLayoutModule	class	core-libs/storefront/cms-structure/page/page-layout/page-layout.module.ts:L24-L24	16 (2), 29 (1)
PageLayoutService	class	core-libs/storefront/cms-structure/page/page-layout/page-layout.service.ts:L29-L278	12 (1), 16 (5), 18 (1), 30 (1)
PageLinkService	class	core-libs/core/src/cms/page/routing/page-link.service.ts:L19-L103	40 (4)
PageMeta	interface	core-libs/core/src/cms/model/page.model.ts:L40-L78	15 (1), 40 (8), 52 (3)
PageMetaConfig	class	core-libs/core/src/cms/page/config/page-meta.config.ts:L14-L16	40 (3)
PageMetaLinkService	class	core-libs/storefront/cms-structure/seo/page-meta-link.service.ts:L19-L72	40 (3)
PageMetaModule	class	core-libs/core/src/cms/page/page-meta.module.ts:L22-L29	40 (2)
PageMetaResolver	class	core-libs/core/src/cms/page/page-meta.resolver.ts:L17-L54	12 (1), 15 (1), 24 (3), 26 (1), 31 (2), 40 (12), 46 (2), 51 (1), 52 (8)
PageMetaResolversConfig	interface	core-libs/core/src/cms/page/config/page-meta.config.ts:L22-L29	40 (1)
PageMetaService	class	core-libs/core/src/cms/facade/page-meta.service.ts:L32-L140	12 (1), 15 (3), 40 (9), 45 (1), 46 (3), 52 (1)
PageRobotsMeta	enum	core-libs/core/src/cms/model/page.model.ts:L85-L90	15 (2), 40 (2), 52 (1)
PageRobotsResolver	interface	core-libs/core/src/cms/page/page.resolvers.ts:L78-L90	40 (1), 52 (1)
PageSection	enum	core-libs/storefront/cms-structure/utils/cms-structure.model.ts:L9-L13	16 (2)
PageSlotComponent	class	core-libs/storefront/cms-structure/page/slot/page-slot.component.ts:L54-L218	04 (2), 15 (2), 16 (4), 17 (4), 18 (1), 33 (1), 51 (2)
PageSlotModule	class	core-libs/storefront/cms-structure/page/slot/page-slot.module.ts:L18-L23	16 (4)
PageSlotService	class	core-libs/storefront/cms-structure/page/slot/page-slot.service.ts:L16-L74	16 (6)
PageTemplateDirective	class	core-libs/storefront/cms-structure/page/page-layout/page-template.directive.ts:L47-L156	07 (2), 16 (4), 18 (1), 51 (1)
PageTitleComponent	class	core-libs/storefront/cms-components/navigation/page-header/page-title.component.ts:L25-L45	45 (2)
PageTitleResolver	interface	core-libs/core/src/cms/page/page.resolvers.ts:L31-L36	40 (1), 52 (1)
PageType	enum	core-libs/core/src/model/cms.model.ts:L38-L43 (+2)	14 (1), 15 (5), 30 (1), 46 (2), 52 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
Pagination	interface	core-libs/core/src/model/unused.model.ts:L33-L38 (+1)	46 (4)
PaginationBuilder	class	core-libs/storefront/shared/components/list-navigation/pagination/pagination.builder.ts:L52-L376	45 (4)
PaginationComponent	class	core-libs/storefront/shared/components/list-navigation/pagination/pagination.component.ts:L47-L224	45 (1), 54 (1)
PaginationConfig	class	core-libs/storefront/shared/components/list-navigation/pagination/config/pagination.config.ts:L15-L17	29 (1), 45 (1)
PaginationModel	interface	core-libs/core/src/model/misc.model.ts:L73-L79 (+1)	46 (5)
PaginationModule	class	core-libs/storefront/shared/components/list-navigation/pagination/pagination.module.ts:L31-L31	45 (1)
PaginationOptions	interface	core-libs/storefront/shared/components/list-navigation/pagination/pagination.model.ts:L33-L96	45 (1)
ParagraphComponent	class	core-libs/storefront/cms-components/content/paragraph/paragraph.component.ts:L26-L52	45 (2), 51 (3)
ParamsMapping	interface	core-libs/core/src/routing/configurable-routes/routes-config.ts:L59-L61	14 (2)
PASSWORD_PATTERN	const	core-libs/core/src/util/regex-pattern.ts:L11-L12	46 (1)
PasswordVisibilityToggleDirective	class	core-libs/storefront/shared/components/form/password-visibility-toggle/password-visibility-toggle.directive.ts:L23-L64	45 (1)
PasswordVisibilityToggleModule	class	core-libs/storefront/shared/components/form/password-visibility-toggle/password-visibility-toggle.module.ts:L29-L29	45 (1)
PAYMENT_CARD_TYPE_NORMALIZER	const	feature-libs/checkout/base/core/connectors/checkout-payment/converters.ts:L14-L16	26 (1)
PAYMENT_DETAILS_NORMALIZER	const	core-libs/core/src/checkout/connectors/payment/converters.ts:L11-L13	26 (4)
PAYMENT_DETAILS_SERIALIZER	const	feature-libs/checkout/base/core/connectors/checkout-payment/converters.ts:L10-L12	26 (1)
PaymentDetails	interface	core-libs/core/src/model/payment.model.ts:L13-L28 (+1)	46 (2)
PDFComponent	class	core-libs/storefront/cms-components/content/pdf/pdf.component.ts:L22-L44	45 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
PDFInvoicesBadRequestHandler	class	feature-libs/pdf-invoices/core/http-interceptors/bad-request/pdf-invoices-badrequest.handler.ts:L20-L91	27 (1)
PDFInvoicesComponentsModule	class	feature-libs/pdf-invoices/components/pdf-invoices-components.module.ts:L53-L53	36 (1)
PDFInvoicesFacade	class	feature-libs/pdf-invoices/root/facade/pdf-invoices.facade.ts:L28-L54	27 (1)
PDFInvoicesService	class	feature-libs/pdf-invoices/core/services/pdf-invoices.service.ts:L19-L81	27 (2)
PdpPickupOptionsContainerComponent	class	feature-libs/pickup-in-store/components/container/pdp-pickup-options-container/pdp-pickup-options-container.component.ts:L61-L197	32 (2)
PersistFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L31-L62	18 (1)
PersistFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/persist/persist-focus.service.ts:L22-L58	18 (1)
PERSONALIZATION_FEATURE	const	feature-libs/tracking/personalization/root/feature-name.ts:L7-L7	33 (3)
PersonalizationAction	interface	feature-libs/tracking/personalization/core/model/personalization-context.model.ts:L7-L14	33 (1)
PersonalizationConfig	class	feature-libs/tracking/personalization/root/config/personalization-config.ts:L14-L26	33 (3)
PersonalizationContext	interface	feature-libs/tracking/personalization/core/model/personalization-context.model.ts:L16-L19	33 (1), 34 (1)
PersonalizationContextService	class	feature-libs/tracking/personalization/core/services/personalization-context.service.ts:L22-L70	33 (4)
PersonalizationCoreModule	class	feature-libs/tracking/personalization/core/personalization-core.module.ts:L10-L10	33 (4)
PersonalizationModule	class	feature-libs/tracking/personalization/personalization.module.ts:L13-L13	33 (5)
PersonalizationRootModule	class	feature-libs/tracking/personalization/root/personalization-root.module.ts:L37-L37	06 (1), 33 (2)
PickupInfoContainerComponent	class	feature-libs/pickup-in-store/components/container/pickup-info-container/pickup-info-container.component.ts:L21-L68	32 (2)
PickupInStoreComponentsModule	class	feature-libs/pickup-in-store/components/pickup-in-store-components.module.ts:L34-L34	17 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
PickupInStoreCoreModule	class	feature-libs/pickup-in-store/core/pickup-in-store-core.module.ts:L23-L23	32 (1)
PickupInStoreRootModule	class	feature-libs/pickup-in-store/root/pickup-in-store-root.module.ts:L60-L60	32 (1)
PickUpItemsDetailsComponent	class	feature-libs/pickup-in-store/components/container/pickup-items-details/pickup-items-details.component.ts:L47-L73	32 (1)
PickupLocationAdapter	class	feature-libs/pickup-in-store/core/connectors/pickup-location.adapter.ts:L13-L19	32 (1)
PickupLocationConnector	class	feature-libs/pickup-in-store/core/connectors/pickup-location.connector.ts:L16-L28	32 (1)
PickupLocationsSearchFacade	class	feature-libs/pickup-in-store/root/facade/pickup-locations-search.facade.ts:L46-L67	32 (1)
PickupOptionDialogComponent	class	feature-libs/pickup-in-store/components/container/pickup-option-dialog/pickup-option-dialog.component.ts:L55-L207	32 (2)
PickupOptionFacade	class	feature-libs/pickup-in-store/root/facade/pickup-option.facade.ts:L28-L38	32 (1)
PLACE_ORDER	const	core-libs/schematics/src/shared/constants.ts:L1052-L1052	26 (2)
PLATFORM_ID	const	core-libs/schematics/src/shared/constants.ts:L20-L20	02 (1), 07 (2), 19 (1), 40 (1), 46 (3)
POINT_OF_SERVICE_NORMALIZER	const	core-libs/core/src/checkout/connectors/checkout/converters.ts:L11-L13	26 (2), 32 (1)
PointOfService	interface	core-libs/core/src/model/point-of-service.model.ts:L11-L25 (+1)	46 (3)
PointOfServiceNames	type	feature-libs/pickup-in-store/root/model/point-of-service-names.model.ts:L10-L13	32 (1)
PointOfServiceStock	interface	core-libs/core/src/model/product.model.ts:L232-L247 (+1)	46 (1)
PopoverComponent	class	core-libs/storefront/shared/components/popover/popover.component.ts:L45-L290	45 (3)
PopoverOptions	interface	core-libs/storefront/shared/components/popover/popover.model.ts:L38-L84	45 (1)
PopoverService	class	core-libs/storefront/shared/components/popover/popover.service.ts:L14-L39	45 (2)
PositioningService	class	core-libs/storefront/shared/services/positioning/positioning.service.ts:L27-L349	45 (2)
POST_PRODUCT_REVIEW_SUCCESS	const	core-libs/core/src/product/store/actions/product-reviews.action.ts:L18-L19	24 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
PreferredStoreFacade	class	feature-libs/pickup-in-store/root/facade/preferred-store.facade.ts:L31-L57	32 (1)
PreferredStoreService	class	feature-libs/pickup-in-store/core/services/preferred-store.service.ts:L33-L98	32 (1)
PrerenderingLoggerService	class	core-libs/setup/ssr/logger/services/prerendering-logger.service.ts:L18-L39	07 (1), 39 (3), 41 (1)
Price	interface	core-libs/core/src/model/product.model.ts:L68-L75 (+1)	46 (2), 54 (1)
PriceType	enum	core-libs/core/src/model/product.model.ts:L63-L66 (+2)	46 (2)
Principal	interface	core-libs/core/src/model/misc.model.ts:L35-L38 (+1)	46 (1)
Priority	enum	core-libs/core/src/util/applicable.ts:L7-L24	41 (2), 46 (3), 50 (4), 54 (1)
PROCESS_FEATURE	const	core-libs/core/src/process/store/process-state.ts:L9-L9	19 (1), 23 (1), 25 (1), 27 (1)
ProcessesLoaderResetAction	class	core-libs/core/src/state/utils/processes-loader/processes-loader.action.ts:L57-L63	19 (1)
ProcessesLoaderState	interface	core-libs/core/src/state/utils/processes-loader/processes-loader-state.ts:L9-L11	19 (1)
ProcessModule	class	core-libs/core/src/process/process.module.ts:L13-L20	02 (1), 19 (5)
ProcessStoreModule	class	core-libs/core/src/process/store/process-store.module.ts:L17-L17	19 (3)
Product	interface	core-libs/core/src/model/product.model.ts:L167-L210 (+1)	30 (2), 31 (2), 46 (7), 51 (1), 52 (8), 54 (2)
PRODUCT_CONFIGURATOR_TEXTFIELD_FEATURE	const	feature-libs/product-configurator/textfield/root/feature-name.ts:L7-L8 (+1)	30 (1)
PRODUCT_DETAIL_ENTITY	const	core-libs/core/src/product/store/product-state.ts:L15-L15	24 (1)
PRODUCT_DETAILS_URL_MATCHER	const	core-libs/storefront/cms-pages/product-details-page/product-details-url-matcher.ts:L42-L50	05 (3), 14 (4), 16 (2), 24 (1)
PRODUCT_FEATURE	const	core-libs/core/src/product/store/product-state.ts:L14-L14	24 (1)
PRODUCT_LISTING_URL_MATCHER	const	core-libs/storefront/cms-pages/product-listing-page/product-listing-url-matcher.ts:L42-L50	05 (2), 16 (2)
PRODUCT_MULTI_DIMENSIONAL_LIST_FEATURE	const	feature-libs/product-multi-dimensional/list/root/feature-name.ts:L7-L8	47 (1)
PRODUCT_NORMALIZER	const	core-libs/core/src/product/connectors/product/converters.ts:L11-L13	01 (2), 21 (5), 24 (4), 25 (1), 27 (1), 31 (2), 46 (2), 52 (7)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
PRODUCT_REFERENCES_NORMALIZER	const	core-libs/core/src/product/connectors/references/converters.ts:L11-L13	24 (1)
PRODUCT_REVIEW_NORMALIZER	const	core-libs/core/src/product/connectors/reviews/converters.ts:L11-L13	24 (1)
PRODUCT_REVIEW_SERIALIZER	const	core-libs/core/src/product/connectors/reviews/converters.ts:L15-L17	21 (1), 24 (1), 52 (2)
PRODUCT_SEARCH_PAGE_NORMALIZER	const	core-libs/core/src/product/connectors/search/converters.ts:L14-L16	24 (1)
PRODUCT_SUGGESTION_NORMALIZER	const	core-libs/core/src/product/connectors/search/converters.ts:L18-L20	24 (1)
ProductAdapter	class	core-libs/core/src/product/connectors/product/product.adapter.ts:L11-L30	01 (2), 21 (6), 24 (3), 52 (6)
ProductAttributesComponent	class	core-libs/storefront/cms-components/product/product-tabs/product-attributes/product-attributes.component.ts:L19-L25	24 (1)
ProductAvailabilityAdapter	class	core-libs/core/src/product/connectors/product/prduct-availability.adapter.ts:L10-L22	21 (2), 24 (3)
ProductAvailabilityService	class	core-libs/core/src/product/facade/product-availability.service.ts:L27-L60	24 (2), 54 (1)
ProductCarouselComponent	class	core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.component.ts:L41-L121	24 (4)
ProductCarouselModule	class	core-libs/storefront/cms-components/product/carousel/product-carousel/product-carousel.module.ts:L53-L53	24 (1)
ProductCarouselService	class	core-libs/storefront/cms-components/product/carousel/product-carousel.service.ts:L21-L64	24 (1)
ProductCatalogService	class	core-libs/core/src/product/services/product-catalog.service.ts:L13-L23	24 (1), 36 (4)
ProductConfiguratorConfig	interface	feature-libs/product-configurator/common/core/model/product-configurator.config.ts:L9-L9	30 (2)
ProductConnector	class	core-libs/core/src/product/connectors/product/product.connector.ts:L16-L33	21 (2), 24 (1), 25 (1)
ProductDetailOutlets	enum	core-libs/storefront/cms-components/product/product-outlets.model.ts:L7-L12	24 (1), 51 (3)
ProductDetailsPageEvent	class	core-libs/storefront/events/product/product-page.events.ts:L13-L20	20 (5)
ProductDetailsPageModule	class	core-libs/storefront/cms-pages/product-details-page/product-details-page.module.ts:L38-L38	16 (2)
ProductEffects	class	core-libs/core/src/product/store/effects/product.effect.ts:L23-L105	24 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ProductEventBuilder	class	core-libs/core/src/product/event/product-event.builder.ts:L22-L117	24 (2)
ProductEventModule	class	core-libs/core/src/product/event/product-event.module.ts:L11-L15	24 (1)
ProductFacetService	class	core-libs/storefront/cms-components/product/product-list/product-facet-navigation/services/product-facet.service.ts:L27-L120	24 (1)
ProductImageNormalizer	class	core-libs/core/src/occ/adapters/product/converters/product-image-normalizer.ts:L15-L106	21 (5), 24 (4)
ProductImagesComponent	class	core-libs/storefront/cms-components/product/product-images/product-images.component.ts:L33-L113	24 (5), 51 (2)
ProductImagesModule	class	core-libs/storefront/cms-components/product/product-images/product-images.module.ts:L40-L40	51 (1)
ProductImageZoomProductImagesComponent	class	feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-product-images/product-image-zoom-product-images.component.ts:L36-L57	24 (1), 51 (1)
ProductImageZoomViewComponent	class	feature-libs/product/image-zoom/components/product-image-zoom/product-image-zoom-view/product-image-zoom-view.component.ts:L73-L429	42 (1)
ProductImportInfo	interface	feature-libs/cart/base/root/models/import-to-cart.model.ts:L21-L27	25 (2)
ProductImportInfoService	class	feature-libs/cart/base/core/services/product-import-info.service.ts:L22-L91	25 (1)
ProductInterestSearchResult	interface	core-libs/core/src/model/product-interest.model.ts:L21-L25 (+1)	46 (1)
ProductList	interface	core-libs/core/src/model/unused.model.ts:L63-L70 (+1)	46 (1)
ProductListComponent	class	core-libs/storefront/cms-components/product/product-list/container/product-list.component.ts:L46-L106	24 (3)
ProductListComponentService	class	core-libs/storefront/cms-components/product/product-list/container/product-list-component.service.ts:L40-L302	24 (2)
ProductListingPageModule	class	core-libs/storefront/cms-pages/product-listing-page/product-listing-page.module.ts:L53-L53	16 (2)
ProductListItemContext	class	core-libs/storefront/cms-components/product/product-list/model/product-list-item-context.model.ts:L15-L17	24 (2), 25 (1), 30 (1)
ProductListItemContextSource	class	core-libs/storefront/cms-components/product/product-list/model/product-list-item-context-source.model.ts:L18-L20	24 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ProductListModule	class	core-libs/storefront/cms-components/product/product-list/product-list.module.ts:L103-L103	24 (2)
ProductListOutlets	enum	core-libs/storefront/cms-components/product/product-outlets.model.ts:L13-L16	24 (1), 51 (1)
ProductLoadingService	class	core-libs/core/src/product/services/product-loading.service.ts:L36-L243	21 (1), 24 (4), 46 (2)
ProductModule	class	core-libs/core/src/product/product.module.ts:L42-L49	24 (2), 52 (1)
ProductMultiDimensionalSelectorGuard	class	feature-libs/product-multi-dimensional/selector/components/guards/product-multi-dimensional-selector.guard.ts:L33-L111	24 (1)
ProductNameNormalizer	class	core-libs/core/src/occ/adapters/product/converters/product-name-normalizer.ts:L14-L65	21 (5), 24 (3), 52 (1)
ProductOccEndpoint	interface	core-libs/core/src/occ/occ-models/occ-endpoints.model.ts:L14-L19	21 (2)
ProductOccModule	class	core-libs/core/src/occ/adapters/product/product-occ.module.ts:L76-L76	21 (8), 24 (3), 52 (7)
ProductPageEventBuilder	class	core-libs/storefront/events/product/product-page-event.builder.ts:L26-L125	20 (3)
ProductPageEventModule	class	core-libs/storefront/events/product/product-page-event.module.ts:L11-L15	20 (2)
ProductPageMetaResolver	class	core-libs/core/src/product/services/product-page-meta.resolver.ts:L40-L210	24 (1), 40 (4), 52 (1)
ProductReferenceNormalizer	class	core-libs/core/src/occ/adapters/product/converters/product-reference-normalizer.ts:L13-L49	21 (1)
ProductReferences	interface	core-libs/core/src/model/product.model.ts:L157-L159	46 (1)
ProductReferencesAdapter	class	core-libs/core/src/product/connectors/references/product-references.adapter.ts:L10-L25	21 (1), 24 (2)
ProductReferencesComponent	class	core-libs/storefront/cms-components/product/carousel/product-references/product-references.component.ts:L42-L109	24 (3)
ProductReferencesEffects	class	core-libs/core/src/product/store/effects/product-references.effect.ts:L17-L52	24 (2)
ProductReferenceService	class	core-libs/core/src/product/facade/product-reference.service.ts:L18-L52	24 (1), 36 (1)
ProductReviewsAdapter	class	core-libs/core/src/product/connectors/reviews/product-reviews.adapter.ts:L10-L28	21 (1), 24 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ProductReviewsEffects	class	core-libs/core/src/product/store/effects/product-reviews.effect.ts:L21-L111	24 (2)
ProductReviewService	class	core-libs/core/src/product/facade/product-review.service.ts:L19-L44	24 (1)
ProductSchemaBuilder	class	core-libs/storefront/cms-structure/seo/structured-data/builders/product/product-schema.builder.ts:L23-L59	40 (3)
ProductScope	enum	core-libs/core/src/product/model/product-scope.ts:L7-L21	21 (1), 24 (6), 46 (2), 52 (2), 54 (1)
ProductScopesConfig	interface	core-libs/core/src/occ/adapters/product/product-occ-config.ts:L20-L26	21 (1)
ProductSearchAdapter	class	core-libs/core/src/product/connectors/search/product-search.adapter.ts:L16-L37	21 (1), 24 (2)
ProductSearchByCategoryEffects	class	core-libs/core/src/product/store/effects/product-search-by-category.effect.ts:L21-L105	24 (2)
ProductSearchByCodeEffects	class	core-libs/core/src/product/store/effects/product-search-by-code.effect.ts:L22-L114	24 (2)
ProductSearchByCodeService	class	core-libs/core/src/product/facade/product-search-by-code.service.ts:L18-L62	24 (1)
ProductSearchLoadByCode	class	core-libs/core/src/product/store/actions/product-search-by-code.action.ts:L22-L32	24 (1)
ProductSearchPage	interface	core-libs/core/src/model/product-search.model.ts:L68-L79 (+1)	46 (4), 54 (1)
ProductSearchService	class	core-libs/core/src/product/facade/product-search.service.ts:L19-L43	20 (1), 24 (2), 40 (1)
ProductService	class	core-libs/core/src/product/facade/product.service.ts:L20-L89	20 (1), 24 (5), 52 (1)
ProductsSearchEffects	class	core-libs/core/src/product/store/effects/product-search.effect.ts:L22-L107	24 (2)
ProductStoreModule	class	core-libs/core/src/product/store/product-store.module.ts:L43-L43	24 (1)
ProductSummaryComponent	class	core-libs/storefront/cms-components/product/product-summary/product-summary.component.ts:L28-L43	24 (1), 51 (1)
ProductTypes	enum	core-libs/core/src/model/product.model.ts:L267-L271	46 (2)
ProductURLPipe	class	core-libs/core/src/routing/configurable-routes/url-translation/product-url.pipe.ts:L11-L16	14 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ProductVariantsContainerComponent	class	feature-libs/product/variants/components/product-variants-container/product-variants-container.component.ts:L35-L59	24 (1)
ProductVariantsGuard	class	feature-libs/product/variants/components/guards/product-variants.guard.ts:L26-L81	24 (3)
ProfileTagComponent	class	integration-libs/cds/src/profiletag/cms-components/profile-tag.component.ts:L18-L21	34 (1)
ProfileTagConfig	interface	integration-libs/cds/src/config/profile-tag.config.ts:L7-L14	34 (1)
ProfileTagModule	class	integration-libs/cds/src/profiletag/profile-tag.module.ts:L32-L32	34 (2)
ProfileTagPushEventsService	class	integration-libs/cds/src/profiletag/services/profile-tag-push-events.service.ts:L71-L409	33 (1), 34 (3)
PromotionLocation	enum	feature-libs/cart/base/root/models/cart.model.ts:L25-L31	25 (2)
PromotionsComponent	class	core-libs/storefront/cms-components/misc/promotions/promotions.component.ts:L17-L24	45 (1)
PromotionsModule	class	core-libs/storefront/cms-components/misc/promotions/promotions.module.ts:L15-L15	45 (1)
PROPAGATE_ERROR_TO_SERVER	const	core-libs/setup/ssr/error-handling/error-response/propagate-error-to-server.ts:L20-L22	07 (4), 39 (6), 41 (1)
PropagatingToServerErrorHandler	class	core-libs/setup/ssr/error-handling/multi-error-handlers/propagating-to-server-error-handler.ts:L29-L38	07 (2), 39 (4), 41 (5)
ProtectedRoutesGuard	class	core-libs/core/src/routing/protected-routes/protected-routes.guard.ts:L14-L34	05 (3), 10 (1), 14 (5)
ProtectedRoutesService	class	core-libs/core/src/routing/protected-routes/protected-routes.service.ts:L12-L90	05 (1), 14 (2), 29 (2)
PROVIDE_OUTLET_OPTIONS	const	core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L13-L15	17 (5), 51 (5), 53 (3)
provideCmsStructure()	function	core-libs/storefront/cms-structure/utils/cms-structure.util.ts:L31-L38	15 (2)
provideConfig()	function	core-libs/core/src/config/config-providers.ts:L18-L27	11 (2)
provideConfigFactory()	function	core-libs/core/src/config/config-providers.ts:L37-L48	11 (1)
provideConfigFromMetaTags()	function	core-libs/core/src/occ/config/config-from-meta-tag-factory.ts:L37-L42	21 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
provideDefaultConfig()	function	core-libs/core/src/config/config-providers.ts:L55-L61	01 (1)
provideDefaultConfigFactory()	function	core-libs/core/src/config/config-providers.ts:L69-L79	42 (1)
provideDefaultFeatureToggles()	function	core-libs/core/src/features-config/feature-toggles/feature-toggles-providers.ts:L61-L69	42 (1)
ProvideLcpPresenceDirective	class	core-libs/storefront/shared/lcp-context/provide-lcp-presence.directive.ts:L34-L47	16 (2)
provideLcpPresenceForCmsComponent()	const	core-libs/storefront/cms-structure/cms-lcp-context/provide-lcp-presence-for-cms-component.ts:L20-L36	16 (2), 45 (1)
provideLoginEventsTracking()	function	integration-libs/cds/src/profiletag/providers/login-events.provider.ts:L20-L48	34 (2)
provideMultiErrorHandler()	function	core-libs/core/src/error-handling/multi-error-handler/provide-multi-error-handler.ts:L11-L19	02 (1), 41 (2), 54 (2)
provideOutlet()	function	core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L41-L47	17 (3)
ProvideOutletOptions	interface	core-libs/storefront/cms-structure/outlet/outlet.providers.ts:L17-L32	17 (4), 51 (1), 53 (1)
provideServer()	function	core-libs/setup/ssr/providers/ssr-providers.ts:L27-L47	07 (1), 41 (3), 46 (3)
PUNCHOUT_REQUISITION_NORMALIZER	const	integration-libs/punchout/core/connectors/converters.ts:L15-L17	36 (1)
PUNCHOUT_SESSION_NORMALIZER	const	integration-libs/punchout/core/connectors/converters.ts:L11-L13	36 (1)
PunchoutAdapter	class	integration-libs/punchout/core/connectors/punchout.adapter.ts:L10-L25	36 (1)
PunchoutAuthHttpHeaderService	class	integration-libs/punchout/root/services/punchout-auth-http-header.service.ts:L24-L67	36 (3)
PunchoutAuthService	class	integration-libs/punchout/core/services/punchout-auth.service.ts:L31-L98	36 (1)
PunchoutCartInterceptor	class	integration-libs/punchout/root/interceptors/punchout-cart.interceptor.ts:L23-L50	06 (3), 36 (4)
PunchoutConnector	class	integration-libs/punchout/core/connectors/punchout.connector.ts:L13-L29	36 (1)
PunchoutDetectionService	class	integration-libs/punchout/root/services/punchout-detection.service.ts:L13-L35	36 (1)
PunchoutExpiredRefreshTokenHandler	class	integration-libs/punchout/root/services/punchout-expired-refresh-token-handler.service.ts:L17-L46	36 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
PunchoutFacade	class	integration-libs/punchout/root/facade/punchout.facade.ts:L37-L79	36 (2)
PunchoutModule	class	integration-libs/punchout/punchout.module.ts:L15-L15	36 (1)
PunchoutNavigationGuard	class	integration-libs/punchout/root/guards/punchout-navigation.guard.ts:L26-L141	36 (2)
PunchoutNavigationGuardConfig	class	integration-libs/punchout/root/config/punchout-guard-config.ts:L16-L34	36 (3)
PunchoutNavigationModule	class	integration-libs/punchout/root/guards/punchout-navigation.module.ts:L21-L21	36 (1)
PunchoutRequisitionComponent	class	integration-libs/punchout/components/punchout-requisition/punchout-requisition.component.ts:L31-L96	36 (1)
PunchoutRootModule	class	integration-libs/punchout/root/punchout.root.module.ts:L84-L84	36 (1)
PunchoutService	class	integration-libs/punchout/core/facade/punchout.service.ts:L43-L369	36 (2)
PunchoutSessionComponent	class	integration-libs/punchout/components/punchout-session/punchout-session.component.ts:L22-L52	36 (1)
PunchoutStatePersistenceService	class	integration-libs/punchout/root/services/punchout-state-persistence.service.ts:L15-L82	36 (1)
PunchoutStoreService	class	integration-libs/punchout/root/services/punchout-store.service.ts:L12-L49	36 (2)
PunchoutUiRestrictionService	class	integration-libs/punchout/root/services/punchout-ui-restriction.service.ts:L19-L59	36 (2)
PwaModule	class	core-libs/storefront/cms-structure/pwa/pwa.module.ts:L56-L56	40 (10)
PWAModuleConfig	class	core-libs/storefront/cms-structure/pwa/pwa.module-config.ts:L14-L19	40 (3)

Q

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
QUALTRICS_FEATURE	const	feature-libs/qualtrics/root/feature-name.ts:L7-L7	33 (2)
QualtricsComponent	class	feature-libs/qualtrics/components/qualtrics-loader/qualtrics.component.ts:L19-L34	33 (8)
QualtricsComponentsModule	class	feature-libs/qualtrics/components/qualtrics-components.module.ts:L35-L35	33 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
QualtricsConfig	class	feature-libs/qualtrics/components/qualtrics-loader/config/qualtrics-config.ts:L18-L30	33 (3)
QualtricsEmbeddedFeedbackComponent	class	feature-libs/qualtrics/components/qualtrics-embedded-feedback/qualtrics-embedded-feedback.component.ts:L13-L13	33 (4)
QualtricsLoaderService	class	feature-libs/qualtrics/components/qualtrics-loader/qualtrics-loader.service.ts:L42-L154	33 (6)
QualtricsModule	class	feature-libs/qualtrics/qualtrics.module.ts:L13-L13	33 (3)
QualtricsRootModule	class	feature-libs/qualtrics/root/qualtrics-root.module.ts:L27-L27	33 (1)
Query	interface	core-libs/core/src/util/command-query/query.service.ts:L39-L42	20 (2), 23 (10), 35 (2)
QueryNotifier	type	core-libs/core/src/util/command-query/query.service.ts:L31-L31	20 (1)
QueryService	class	core-libs/core/src/util/command-query/query.service.ts:L47-L160	20 (3), 29 (1)
QueryState	interface	core-libs/core/src/util/command-query/query.service.ts:L33-L37	20 (1)
QuickOrderComponent	class	feature-libs/cart/quick-order/components/quick-order/quick-order.component.ts:L57-L264	25 (2)
QuickOrderConfig	class	feature-libs/cart/quick-order/root/config/quick-order-config.ts:L14-L25	25 (1)
QuickOrderOrderEntriesContextToken	const	feature-libs/cart/quick-order/root/tokens/context.ts:L9-L11	25 (1)
QuickOrderRootModule	class	feature-libs/cart/quick-order/root/quick-order-root.module.ts:L69-L69	25 (1)
QuickOrderService	class	feature-libs/cart/quick-order/core/services/quick-order.service.ts:L40-L460	25 (2)
Quote	type	feature-libs/quote/root/model/quote.model.ts:L41-L44	31 (2)
QUOTE_NORMALIZER	const	feature-libs/quote/core/connectors/converters.ts:L20-L22	31 (3)
QuoteAction	interface	feature-libs/quote/root/model/quote.model.ts:L46-L49	31 (1)
QuoteActionType	enum	feature-libs/quote/root/model/quote.model.ts:L58-L73	31 (2)
QuoteAdapter	class	feature-libs/quote/core/connectors/quote.adapter.ts:L19-L138	31 (1)
QuoteBadRequestHandler	class	feature-libs/quote/root/http-interceptors/quote-bad-request.handler.ts:L21-L181	31 (4)
QuoteCartEventListener	class	feature-libs/quote/core/event/quote-cart-event.listener.ts:L21-L54	31 (4), 53 (1)
QuoteCartGuard	class	feature-libs/quote/components/cart-guard/quote-cart.guard.ts:L28-L81	31 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
QuoteCartGuardComponent	class	feature-libs/quote/components/cart-guard/quote-cart-guard.component.ts:L17-L17	31 (3)
QuoteCartService	class	feature-libs/quote/core/services/quote-cart.service.ts:L15-L64	31 (2)
QuoteComponentsModule	class	feature-libs/quote/components/quote-components.module.ts:L42-L42	31 (1)
QuoteConfig	interface	feature-libs/quote/root/config/quote.config.ts:L9-L9	31 (2)
QuoteConfirmDialogComponent	class	feature-libs/quote/components/confirm-dialog/quote-confirm-dialog.component.ts:L41-L115	31 (2)
QuoteConnector	class	feature-libs/quote/core/connectors/quote.connector.ts:L22-L167	31 (2)
QuoteCoreConfig	class	feature-libs/quote/core/config/quote-core.config.ts:L22-L29	31 (1)
QuoteCoreModule	class	feature-libs/quote/core/quote-core.module.ts:L21-L25	31 (2)
QuoteDetailsReloadQueryEvent	class	feature-libs/quote/core/event/quote.events.ts:L9-L14	31 (7)
QuoteDiscountType	enum	feature-libs/quote/root/model/quote.model.ts:L95-L98	31 (2)
QuoteFacade	class	feature-libs/quote/root/facade/quote.facade.ts:L39-L135	31 (5)
QuoteLinksComponent	class	feature-libs/quote/components/links/quote-links.component.ts:L37-L104	31 (1)
QuoteListModule	class	feature-libs/quote/components/list/quote-list.module.ts:L53-L53	31 (3)
QuoteMetadata	interface	feature-libs/quote/root/model/quote.model.ts:L120-L128	31 (2)
QuoteModule	class	feature-libs/quote/quote.module.ts:L15-L15	31 (2)
QuoteNotFoundHandler	class	feature-libs/quote/root/http-interceptors/quote-not-found.handler.ts:L21-L61	31 (4)
QuoteOccModule	class	feature-libs/quote/occ/quote-occ.module.ts:L37-L37	31 (2)
QuoteRequestButtonComponent	class	feature-libs/quote/components/request-button/quote-request-button.component.ts:L25-L62	31 (2)
QuoteRoleType	enum	feature-libs/quote/root/model/quote.model.ts:L100-L105	31 (2)
QuoteService	class	feature-libs/quote/core/facade/quote.service.ts:L65-L570	31 (6)
QuotesStateParams	interface	feature-libs/quote/root/model/quote.model.ts:L137-L140	31 (1)
QuoteStarter	interface	feature-libs/quote/root/model/quote.model.ts:L130-L133	31 (2)
QuoteState	enum	feature-libs/quote/root/model/quote.model.ts:L75-L93	31 (2)
QuoteSummaryActionsComponent	class	feature-libs/quote/components/summary/actions/quote-summary-actions.component.ts:L54-L426	31 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
QuoteSummaryComponent	class	feature-libs/quote/components/summary/quote-summary.component.ts:L28-L32	31 (1)
QuoteUIConfig	class	feature-libs/quote/components/config/quote-ui.config.ts:L51-L53	31 (2)

R

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ReadConfiguration	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L129-L140	30 (2)
ReadConfigurationSuccess	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L153-L159	30 (1)
ReadOnlyPostfix	const	feature-libs/product-configurator/common/core/model/common-configurator.model.ts:L70-L70	30 (1)
ReadOrderEntryConfiguration	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L65-L72 (+1)	30 (1)
RecentSearchesComponent	class	integration-libs/cds/src/recent-searches/recent-searches.component.ts:L60-L140	34 (1)
RecentSearchesHeaderComponent	class	integration-libs/cds/src/recent-searches/recent-searches-header.component.ts:L18-L34	34 (1)
RecentSearchesModule	class	integration-libs/cds/src/recent-searches/recent-searches.module.ts:L47-L47	34 (2)
RecentSearchesService	class	integration-libs/cds/src/recent-searches/recent-searches.service.ts:L14-L80	34 (1)
Region	interface	core-libs/core/src/model/address.model.ts:L19-L24 (+1)	46 (1)
REGISTER_USER_SUCCESS	const	core-libs/core/src/user/store/actions/user-register.action.ts:L11-L11	23 (3)
RegisterComponentService	class	feature-libs/user/profile/components/register/register-component.service.ts:L20-L86	23 (1), 34 (1)
RegisterUserSuccess	class	core-libs/core/src/user/store/actions/user-register.action.ts:L13-L18	23 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
REMOVE_CART	const	feature-libs/cart/base/core/store/actions/cart.action.ts:L33-L33	25 (1)
RemoveCart	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L263-L268	25 (3)
RemoveCartEvent	class	feature-libs/cart/base/root/events/cart.events.ts:L170-L175	25 (3), 26 (3), 27 (1)
RemoveConfiguration	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L375-L381 (+1)	30 (5)
RemoveLocation	const	feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L57-L60	32 (2)
RemoveMessage	class	core-libs/core/src/global-message/store/actions/global-message.actions.ts:L23-L26	41 (3)
RemoveMessagesByType	class	core-libs/core/src/global-message/store/actions/global-message.actions.ts:L28-L31	41 (2)
RemovePickupOption	const	feature-libs/pickup-in-store/core/store/actions/pickup-option.action.ts:L21-L24	32 (1)
RenamedSymbol	interface	core-libs/schematics/src/shared/utils/file-utils.ts:L112-L117	43 (1), 54 (1)
RenderingCache	class	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.ts:L11-L128	39 (5)
RenderingCacheSizeManager	class	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache-size-manager.ts:L14-L110	39 (4)
RenderingEntry	interface	core-libs/setup/ssr/optimized-engine/rendering-cache/rendering-cache.model.ts:L10-L20	39 (2)
RenderingStrategy	enum	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L183-L187	39 (1)
RenderingStrategyResolverOptions	interface	core-libs/setup/ssr/optimized-engine/rendering-strategy-resolver-options.ts:L7-L10	39 (1)
ReorderOrderFacade	class	feature-libs/order/root/facade/reorder-order.facade.ts:L22-L27	27 (1)
ReorderOrderService	class	feature-libs/order/core/facade/reorder-order.service.ts:L25-L93	27 (1)
REPLENISHMENT_ORDER_FORM_SERIALIZER	const	feature-libs/order/root/tokens/converters.ts:L60-L62	27 (1)
REPLENISHMENT_ORDER_HISTORY_NORMALIZER	const	feature-libs/order/root/tokens/converters.ts:L56-L58	27 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ReplenishmentOrder	interface	feature-libs/order/root/model/replenishment-order.model.ts:L26-L70 (+1)	27 (1)
ReplenishmentOrderDetailsService	class	feature-libs/order/components/replenishment-order-details/replenishment-order-details.service.ts:L25-L56	27 (2)
ReplenishmentOrderHistoryFacade	class	feature-libs/order/root/facade/replenishment-order-history.facade.ts:L47-L145	27 (2)
ReplenishmentOrderHistoryService	class	feature-libs/order/core/facade/replenishment-order-history.service.ts:L29-L249	27 (1)
ReplenishmentOrderScheduledEvent	class	feature-libs/order/root/events/order.events.ts:L49-L58	26 (1), 27 (3)
REQUEST	const	core-libs/setup/ssr/tokens/express.tokens.ts:L15-L17	07 (6), 39 (6), 41 (1), 46 (1), 47 (1)
RequestContext	interface	core-libs/setup/ssr/optimized-engine/request-context.ts:L17-L21	39 (2)
RequestedDeliveryDateAdapter	class	feature-libs/requested-delivery-date/core/connectors/requested-delivery-date.adapter.ts:L9-L18	32 (1)
RequestedDeliveryDateBadRequestHandler	class	feature-libs/requested-delivery-date/core/http-interceptors/bad-request/requested-delivery-date-badrequest.handler.ts:L20-L49	32 (2)
RequestedDeliveryDateComponentsModule	class	feature-libs/requested-delivery-date/root/components/requested-delivery-date-components.module.ts:L30-L30	36 (1)
RequestedDeliveryDateFacade	class	feature-libs/requested-delivery-date/root/facade/requested-delivery-date.facade.ts:L24-L33	32 (2)
RequestedDeliveryDateModule	class	feature-libs/requested-delivery-date/requested-delivery-date.module.ts:L14-L14	32 (2)
RequiredPick	type	core-libs/core/src/util/type-utils.ts:L7-L11	46 (1)
RescheduleServiceOrderComponent	class	integration-libs/s4-service/order/components/reschedule-service-order/reschedule-service-order.component.ts:L66-L169	36 (2)
RescheduleServiceOrderFacade	class	integration-libs/s4-service/root/facade/reschedule-service-order.facade.ts:L14-L24	36 (3)
ResetCartDetails	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L242-L247	19 (1), 25 (2)
ResetCartDetailsByIds	class	feature-libs/cart/base/core/store/actions/cart.action.ts:L252-L257	25 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
ResetGiveUserConsentProcess	class	core-libs/core/src/user/store/actions/user-consents.action.ts:L104-L110	19 (1)
ResetSubscribeCustomerCouponProcess	class	core-libs/core/src/user/store/actions/customer-coupon.action.ts:L136-L141	19 (1)
RESPONSE	const	core-libs/setup/ssr/tokens/express.tokens.ts:L24-L26	07 (4), 39 (2)
RestoreSavedCart	class	feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L130-L141	25 (1)
RestoreSavedCartSuccess	class	feature-libs/cart/saved-cart/core/store/actions/saved-cart.action.ts:L143-L154	25 (1)
RestoreSavedCartSuccessEvent	class	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L54-L61	26 (1)
RetractValueCode	const	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L390-L390	30 (1)
RETURN_REQUEST_DETAILS	const	feature-libs/order/core/store/order-state.ts:L24-L24	27 (1)
ReturnOrderComponent	class	feature-libs/order/components/amend-order/returns/return-order/return-order.component.ts:L31-L76	27 (2)
ReturnRequest	interface	feature-libs/order/root/model/order.model.ts:L49-L62 (+1)	27 (2)
ReturnRequestService	class	feature-libs/order/components/return-request-detail/return-request.service.ts:L20-L95	27 (2)
RootConfig	const	core-libs/core/src/config/config-tokens.ts:L44-L47	01 (6), 10 (2), 11 (10), 12 (1), 42 (2), 50 (3), 51 (2), 53 (3)
RootFeatureToggles	const	core-libs/core/src/features-config/feature-toggles/feature-toggles-tokens.ts:L48-L58	42 (4), 44 (1)
ROUTE_PARAMS	const	feature-libs/organization/administration/root/route-params.ts:L10-L18	28 (3)
RouteBreadcrumbConfig	interface	core-libs/core/src/cms/page/routing/route-page-meta.model.ts:L45-L55	40 (1)
RouteConfig	interface	core-libs/core/src/routing/configurable-routes/routes-config.ts:L15-L57	05 (2), 10 (1), 14 (3), 51 (3)
RouteLoadStrategy	enum	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L11-L20	05 (2), 14 (2), 51 (2)
RoutePageMetaConfig	interface	core-libs/core/src/cms/page/routing/route-page-meta.model.ts:L30-L40	40 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
RouterEffects	class	core-libs/core/src/routing/store/effects/router.effect.ts:L18-L43	05 (1)
RouterState	interface	core-libs/core/src/routing/store/routing-state.ts:L13-L16	14 (1), 20 (1)
RoutesConfig	interface	core-libs/core/src/routing/configurable-routes/routes-config.ts:L11-L13	14 (3), 34 (1), 51 (2)
ROUTING_FEATURE	const	core-libs/core/src/routing/store/routing-state.ts:L11-L11	14 (1)
RoutingConfig	class	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L26-L28	01 (1), 05 (1), 10 (1), 14 (2), 22 (1), 26 (1), 27 (1), 29 (2), 30 (1), 31 (3), 35 (2), 36 (2), 51 (4), 53 (2)
RoutingConfigDefinition	interface	core-libs/core/src/routing/configurable-routes/config/routing-config.ts:L30-L45	14 (2), 54 (1)
RoutingConfigService	class	core-libs/core/src/routing/configurable-routes/routing-config.service.ts:L13-L93	10 (1), 14 (2)
RoutingContextService	class	core-libs/storefront/context/routing-context.service.ts:L19-L64	14 (2)
RoutingModule	class	core-libs/core/src/routing/routing.module.ts:L83-L109 (+1)	02 (2), 05 (5), 10 (4), 11 (1), 14 (4)
RoutingPageMetaResolver	class	core-libs/core/src/cms/page/routing/routing-page-meta.resolver.ts:L38-L199	40 (1)
RoutingParamsService	class	core-libs/core/src/routing/facade/routing-params.service.ts:L19-L45	14 (3)
RoutingService	class	core-libs/core/src/routing/facade/routing.service.ts:L28-L170	14 (4)
RulebasedConfiguratorAdapter	class	feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.adapter.ts:L12-L161	30 (3)
RulebasedConfiguratorComponentsModule	class	feature-libs/product-configurator/rulebased/components/rulebased-configurator-components.module.ts:L60-L60	30 (2)
RulebasedConfiguratorConnector	class	feature-libs/product-configurator/rulebased/core/connectors/rulebased-configurator.connector.ts:L20-L191	30 (5)
RulebasedConfiguratorCoreModule	class	feature-libs/product-configurator/rulebased/core/rulebased-configurator-core.module.ts:L27-L34	30 (1)
RulebasedConfiguratorModule	class	feature-libs/product-configurator/rulebased/rulebased-configurator.module.ts:L19-L19	30 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
RulebasedConfiguratorRootModule	class	feature-libs/product-configurator/rulebased/root/rulebased-configurator-root.module.ts:L38-L44	30 (3)
RulebasedCpqConfiguratorModule	class	feature-libs/product-configurator/rulebased/cpq/rulebased-cpq-configurator.module.ts:L17-L17	30 (4), 47 (1)

S

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
S4_SERVICE_CMS_COMPONENTS	const	integration-libs/s4-service/root/s4-service-root.module.ts:L31-L34	36 (1)
S4_SERVICE_FEATURE	const	integration-libs/s4-service/root/feature-name.ts:L7-L7	36 (1)
S4_SERVICE_ORDER_CMS_COMPONENTS	const	integration-libs/s4-service/root/s4-service-root.module.ts:L35-L39	36 (1)
S4omModule	class	integration-libs/s4om/s4om.module.ts:L12-L12	36 (1)
S4omProductCatalogService	class	integration-libs/s4om/root/services/s4om-product-catalog.service.ts:L13-L38	36 (1)
S4omRootModule	class	integration-libs/s4om/root/s4om-root.module.ts:L35-L35	36 (1)
S4ServiceCheckoutModule	class	integration-libs/s4-service/checkout/s4-service-checkout.module.ts:L19-L19	36 (5), 47 (1)
S4ServiceDeliveryModeConfig	class	integration-libs/s4-service/root/model/augmented-types.model.ts:L58-L60	36 (2)
S4ServiceModule	class	integration-libs/s4-service/s4-service.module.ts:L12-L12	36 (1)
S4ServiceOrderCoreModule	class	integration-libs/s4-service/order/core/s4-service-order-core.module.ts:L39-L39	36 (1)
S4ServiceOrderModule	class	integration-libs/s4-service/order/s4-service-order.module.ts:L19-L19	36 (3), 47 (1)
S4ServiceRootModule	class	integration-libs/s4-service/root/s4-service-root.module.ts:L83-L83	36 (2)
SAVE_CART_NORMALIZER	const	feature-libs/cart/base/root/tokens/converters.ts:L29-L31	25 (2)
SaveCartEvent	class	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L16-L23	25 (1)
SaveCartSuccessEvent	class	feature-libs/cart/saved-cart/root/events/saved-cart.events.ts:L25-L33	26 (1)
SavedCartAdapter	class	feature-libs/cart/saved-cart/core/connectors/saved-cart.adapter.ts:L10-L35	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SavedCartOrderEntriesContextToken	const	feature-libs/cart/saved-cart/root/tokens/context.ts:L9-L11	25 (1)
SavedCartService	class	feature-libs/cart/saved-cart/core/facade/saved-cart.service.ts:L44-L477	25 (2)
SavedCartStoreModule	class	feature-libs/cart/saved-cart/core/store/saved-cart-store.module.ts:L14-L14	25 (1)
SceneAdapter	class	integration-libs/epd-visualization/core/connectors/scene/scene.adapter.ts:L13-L30	36 (1)
SceneNodeToProductLookupService	class	integration-libs/epd-visualization/core/services/scene-node-to-product-lookup/scene-node-to-product-lookup.service.ts:L34-L235	36 (1)
ScheduledReplenishmentOrderFacade	class	feature-libs/order/root/facade/scheduled-replenishment-order.facade.ts:L23-L31	27 (1)
ScheduledReplenishmentOrderService	class	feature-libs/order/core/facade/scheduled-replenishment-order.service.ts:L29-L111	27 (1)
ScheduleLinesComponent	class	integration-libs/s4om/root/components/schedule-lines/schedule-lines.component.ts:L19-L52	36 (3)
ScheduleReplenishmentForm	interface	feature-libs/order/root/model/scheduled-replenishment.model.ts:L23-L30 (+1)	27 (1)
Schema	interface	core-libs/schematics/src/add-spartacus/schema.ts:L11-L27 (+2)	43 (2)
SCHEMA_BUILDER	const	core-libs/storefront/cms-structure/seo/structured-data/builders/tokens.ts:L15-L15	40 (8)
SchemaBuilder	interface	core-libs/storefront/cms-structure/seo/structured-data/builders/schema.interface.ts:L9-L11	40 (2)
SchematicConfig	interface	core-libs/schematics/src/shared/utils/lib-utils.ts:L115-L199	08 (4), 43 (7)
SCHEMATICS_CONFIGS	const	core-libs/schematics/src/shared/schematics-config-mappings.ts:L104-L198	43 (3)
ScopedProductData	interface	core-libs/core/src/product/connectors/product/scoped-product-data.ts:L13-L15	21 (2), 46 (1), 52 (1)
ScriptLoader	class	core-libs/core/src/util/script-loader.service.ts:L18-L122	33 (4), 46 (6)
ScriptPlacement	enum	core-libs/core/src/util/script-loader.service.ts:L10-L13	46 (3)
ScrollToTopComponent	class	core-libs/storefront/cms-components/navigation/scroll-to-top/scroll-to-top.component.ts:L34-L124	45 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SEARCH_VARIANTS_FAIL	const	feature-libs/product-configurator/rulebased/core/state/actions/configurator-variant.action.ts:L12-L12	30 (1)
SearchBoxComponent	class	core-libs/storefront/cms-components/navigation/search-box/search-box.component.ts:L94-L951	18 (1), 45 (2)
SearchBoxComponentService	class	core-libs/storefront/cms-components/navigation/search-box/search-box-component.service.ts:L39-L429	24 (1), 45 (2)
SearchBoxConfig	interface	core-libs/storefront/cms-components/navigation/search-box/search-box.model.ts:L7-L18	45 (3)
SearchBoxOutlets	enum	core-libs/storefront/cms-components/navigation/search-box/search-box-outlets.model.ts:L7-L11	45 (2)
SearchBoxProductSelectedEvent	class	core-libs/storefront/cms-components/navigation/search-box/search-box.events.ts:L25-L32	45 (2)
SearchboxService	class	core-libs/core/src/product/facade/searchbox.service.ts:L19-L100	24 (2)
SearchBoxSuggestionSelectedEvent	class	core-libs/storefront/cms-components/navigation/search-box/search-box.events.ts:L12-L20	45 (2)
SearchConfig	interface	core-libs/core/src/product/model/search-config.ts:L8-L24	24 (2), 28 (1)
SearchPageMetaResolver	class	core-libs/core/src/product/services/search-page-meta.resolver.ts:L31-L88	24 (2), 40 (4)
SearchPageResultsEvent	class	core-libs/storefront/events/product/product-page.events.ts:L37-L42	20 (2)
SearchProducts	class	core-libs/core/src/product/store/actions/product-search.action.ts:L27-L33	24 (2)
SearchVariants	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator-variant.action.ts:L15-L20	30 (4)
SearchVariantsSuccess	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator-variant.action.ts:L32-L39	30 (1)
SECURE_PASSWORD_PATTERN	const	core-libs/core/src/util/regex-pattern.ts:L32-L33	46 (1)
SecurePortalConfigInitializer	class	core-libs/core/src/routing/configurable-routes/secure-portal-config/secure-portal-config-initializer.ts:L17-L51	05 (2), 10 (6), 11 (3), 14 (2)
SEGMENT_REFS_FEATURE	const	integration-libs/segment-refs/root/feature-name.ts:L7-L7	34 (1)
SegmentRefsConfig	class	integration-libs/segment-refs/root/config/segment-refs-config.ts:L14-L18	34 (3), 50 (1), 54 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SegmentRefsRootModule	class	integration-libs/segment-refs/root/segment-refs-root.module.ts:L20-L20	34 (1), 50 (1)
SelectFocusUtility	class	core-libs/storefront/layout/a11y/keyboard-focus/services/select-focus.util.ts:L13-L93	18 (4)
SelectionDisplayMode	enum	integration-libs/epd-visualization/components/visual-viewer/models/selection-display-mode.ts:L12-L22	36 (1)
SelectiveCartFacade	class	feature-libs/cart/base/root/facade/selective-cart.facade.ts:L31-L48	25 (3)
SelectiveCartService	class	feature-libs/cart/base/core/facade/selective-cart.service.ts:L33-L133	25 (2)
SemanticPathService	class	core-libs/core/src/routing/configurable-routes/url-translation/semantic-path.service.ts:L15-L172	14 (3)
SeoConfig	class	core-libs/storefront/cms-structure/seo/config/seo.config.ts:L14-L16	40 (2)
SeoMetaService	class	core-libs/storefront/cms-structure/seo/seo-meta.service.ts:L22-L94	40 (8), 52 (2)
SeoModule	class	core-libs/storefront/cms-structure/seo/seo.module.ts:L35-L35	40 (7)
SERVER_REQUEST_ORIGIN	const	core-libs/core/src/util/ssr.tokens.ts:L19-L21	02 (1), 07 (8), 14 (1), 39 (2), 40 (1), 43 (1), 46 (6), 47 (6)
SERVER_REQUEST_URL	const	core-libs/core/src/util/ssr.tokens.ts:L12-L14	07 (5), 13 (2), 14 (1), 34 (1), 40 (1), 45 (1), 46 (6), 47 (1)
ServerOptions	interface	core-libs/setup/ssr/providers/model.ts:L10-L23	07 (1), 39 (1), 46 (1)
ServiceDetailsCardComponent	class	integration-libs/s4-service/order/components/order-summary/service-details-card.component.ts:L24-L82	36 (1)
ServiceDetailsCardModule	class	integration-libs/s4-service/order/components/order-summary/service-details-card.module.ts:L31-L31	36 (1)
ServiceOrderGuard	class	integration-libs/s4-service/order/components/guards/service-order.guard.ts:L16-L35	36 (1)
ServiceOrderOverviewComponentService	class	integration-libs/s4-service/order/components/order-summary/service-order-overview-component.service.ts:L13-L20	36 (2)
SET_ACTIVE_BASE_SITE	const	core-libs/core/src/site-context/store/actions/base-site.action.ts:L19-L19	13 (2)
SET_ACTIVE_CART_ID	const	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L15-L15	25 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SET_ACTIVE_CURRENCY	const	core-libs/core/src/site-context/store/actions/currencies.action.ts:L14-L14	13 (1)
SET_ACTIVE_LANGUAGE	const	core-libs/core/src/site-context/store/actions/languages.action.ts:L14-L14	13 (2)
SET_ACTIVE_SITE_THEME	const	core-libs/core/src/site-theme/store/actions/site-themes.action.ts:L9-L9	13 (2)
SET_ANONYMOUS_CONSENTS	const	core-libs/core/src/anonymous-consents/store/actions/anonymous-consents.action.ts:L28-L29	23 (1)
SET_CART_DATA	const	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L20-L20	25 (1)
SET_CART_TYPE_INDEX	const	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L19-L19	25 (1)
SET_NEXT_OWNER_CART_ENTRY	const	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L33-L34	30 (1)
SetActiveBaseSite	class	core-libs/core/src/site-context/store/actions/base-site.action.ts:L60-L64	10 (1)
SetActiveCartId	class	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L47-L50	25 (2)
SetActiveSiteTheme	class	core-libs/core/src/site-theme/store/actions/site-themes.action.ts:L12-L15	13 (1), 38 (1)
SetCartData	class	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L69-L74	25 (2)
SetCartTypeIndex	class	feature-libs/cart/base/core/store/actions/multi-cart.action.ts:L62-L67	25 (1)
SetCurrentGroup	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L454-L462	30 (1)
SetInteractionState	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L441-L452	30 (1)
SetMenuParentGroup	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L464-L478	30 (1)
SetNextOwnerCartEntry	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L119-L130	30 (1)
SetPickupOption	const	feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L66-L69 (+1)	32 (1)
SetPreferredStoreComponent	class	feature-libs/pickup-in-store/components/container/set-preferred-store/set-preferred-store.component.ts:L27-L64	32 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SetStoreDetailsFailure	const	feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L92-L102	32 (3)
SetStoreDetailsSuccess	const	feature-libs/pickup-in-store/core/store/actions/pickup-location.action.ts:L79-L82	32 (1)
SharedOrganizationModule	class	feature-libs/organization/administration/components/shared/shared-organization.module.ts:L15-L15	28 (1)
ShowEstimatedDeliveryDateModule	class	feature-libs/estimated-delivery-date/show-estimated-delivery-date/show-estimated-delivery-date.module.ts:L37-L37	32 (2)
SITE_THEME_CHANGE	const	core-libs/core/src/site-theme/store/actions/site-themes.action.ts:L10-L10	13 (1)
SITE_THEME_ID	const	core-libs/core/src/site-theme/providers/site-theme-id.ts:L7-L7	13 (1)
SiteAdapter	class	core-libs/core/src/site-context/connectors/site.adapter.ts:L17-L57	13 (3), 21 (2)
SiteConnector	class	core-libs/core/src/site-context/connectors/site.connector.ts:L22-L56	13 (4), 23 (1)
SiteContextComponentService	class	core-libs/storefront/cms-components/misc/site-context-selector/site-context-component.service.ts:L28-L137	45 (4)
SiteContextConfig	class	core-libs/core/src/site-context/config/site-context-config.ts:L14-L19	11 (4), 13 (3), 21 (1), 38 (1)
SiteContextConfigInitializer	class	core-libs/core/src/site-context/config/config-loader/site-context-config-initializer.ts:L25-L152	01 (3), 10 (4), 11 (5), 13 (6), 21 (2), 37 (1)
SiteContextEventBuilder	class	core-libs/core/src/site-context/events/site-context-event.builder.ts:L22-L73	13 (2)
SiteContextEventModule	class	core-libs/core/src/site-context/events/site-context-event.module.ts:L11-L15	13 (3)
SiteContextInterceptor	class	feature-libs/asm/customer-360/root/interceptors/site-context.interceptor.ts:L26-L73 (+1)	06 (14), 13 (7), 21 (6), 29 (4), 52 (2)
SiteContextModule	class	core-libs/core/src/site-context/site-context.module.ts:L46-L66	02 (1), 13 (2)
SiteContextOccModule	class	core-libs/core/src/occ/adapters/site-context/site-context-occ.module.ts:L38-L38	06 (1), 13 (2), 21 (3)
SiteContextParamsService	class	core-libs/core/src/site-context/services/site-context-params.service.ts:L20-L108	13 (6)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SiteContextRoutesHandler	class	core-libs/core/src/site-context/services/site-context-routes-handler.ts:L25-L173	13 (9), 54 (1)
SiteContextSelectorComponent	class	core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.component.ts:L42-L83	45 (3)
SiteContextSelectorModule	class	core-libs/storefront/cms-components/misc/site-context-selector/site-context-selector.module.ts:L55-L55	45 (1)
SiteContextStoreModule	class	core-libs/core/src/site-context/store/site-context-store.module.ts:L43-L43	13 (2)
SiteContextUrlSerializer	class	core-libs/core/src/site-context/services/site-context-url-serializer.ts:L31-L160	02 (3), 05 (5), 10 (1), 13 (8), 54 (1)
SiteTheme	interface	core-libs/core/src/model/misc.model.ts:L140-L143	13 (1), 46 (1), 53 (1)
SiteThemeChange	class	core-libs/core/src/site-theme/store/actions/site-themes.action.ts:L17-L22	13 (1)
SiteThemeConfig	class	core-libs/core/src/site-theme/config/site-theme-config.ts:L15-L19	13 (3), 53 (2)
SiteThemeInitializer	class	core-libs/core/src/site-theme/services/site-theme-initializer.ts:L23-L157	13 (3)
SiteThemePersistenceService	class	core-libs/core/src/site-theme/services/site-theme-persistence.service.ts:L15-L44	13 (1)
SiteThemeService	class	core-libs/core/src/site-theme/facade/site-theme.service.ts:L28-L138	13 (4), 38 (1), 45 (1)
SiteThemeStoreModule	class	core-libs/core/src/site-theme/store/site-theme-store.module.ts:L42-L42	13 (1)
SiteThemeSwitcherComponent	class	core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.component.ts:L40-L78	38 (1), 45 (1)
SiteThemeSwitcherComponentService	class	core-libs/storefront/cms-components/misc/site-theme-switcher/site-theme-switcher.component.service.ts:L12-L26	45 (1)
SkipFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L17-L24	18 (1)
SkipFocusDirective	class	core-libs/storefront/layout/a11y/keyboard-focus/skip-focus.directive.ts:L31-L78	18 (7)
SkipLink	class	core-libs/storefront/layout/a11y/skip-link/config/skip-link.config.ts:L18-L23	18 (1)
SkipLinkComponent	class	core-libs/storefront/layout/a11y/skip-link/component/skip-link.component.ts:L21-L29	18 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SkipLinkConfig	class	core-libs/storefront/layout/a11y/skip-link/config/skip-link.config.ts:L14-L16	18 (1)
SkipLinkDirective	class	core-libs/storefront/layout/a11y/skip-link/directive/skip-link.directive.ts:L11-L26	18 (1)
SkipLinkModule	class	core-libs/storefront/layout/a11y/skip-link/skip-link.module.ts:L45-L45	18 (3)
SkipLinkService	class	core-libs/storefront/layout/a11y/skip-link/service/skip-link.service.ts:L15-L112	18 (1)
SlotConfig	type	core-libs/storefront/layout/config/layout-config.ts:L36-L46	18 (2), 50 (1), 51 (1)
SlotDecorator	class	core-libs/core/src/cms/decorators/slot-decorator.ts:L11-L23	12 (1), 15 (3), 33 (3)
SlotGroup	type	core-libs/storefront/layout/config/layout-config.ts:L48-L57	16 (1), 18 (4), 51 (2)
SMART_EDIT_CONTEXT	const	core-libs/core/src/routing/models/page-context.model.ts:L20-L20	14 (1), 15 (1)
SMART_EDIT_FEATURE	const	feature-libs/smartedit/root/feature-name.ts:L7-L7	33 (2)
SmartEditComponentDecorator	class	feature-libs/smartedit/core/decorators/smart-edit-component-decorator.ts:L14-L32	15 (1), 33 (2)
SmartEditConfig	class	feature-libs/smartedit/root/config/smart-edit-config.ts:L14-L19	33 (2)
SmartEditCoreModule	class	feature-libs/smartedit/core/smart-edit-core.module.ts:L14-L18	33 (5)
SmartEditLauncherService	class	feature-libs/smartedit/root/services/smart-edit-launcher.service.ts:L24-L119	33 (3)
SmartEditModule	class	feature-libs/smartedit/smart-edit.module.ts:L13-L13	33 (4)
SmartEditRootModule	class	feature-libs/smartedit/root/smart-edit-root.module.ts:L34-L34	06 (1), 33 (3)
SmartEditService	class	feature-libs/smartedit/core/services/smart-edit.service.ts:L22-L192	33 (5)
SmartEditSlotDecorator	class	feature-libs/smartedit/core/decorators/smart-edit-slot-decorator.ts:L14-L28	15 (1), 33 (2)
Sort	interface	core-libs/core/src/model/unused.model.ts:L40-L43 (+1)	46 (4)
SortingComponent	class	core-libs/storefront/shared/components/list-navigation/sorting/sorting.component.ts:L37-L106	04 (1), 45 (1)
SortModel	interface	core-libs/core/src/model/misc.model.ts:L81-L85 (+1)	46 (4)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SpinnerComponent	class	core-libs/storefront/shared/components/spinner/spinner.component.ts:L17-L21	22 (2), 26 (1)
SsrCallbackFn	type	core-libs/setup/ssr/optimized-engine/optimized-ssr-engine.ts:L22-L31	39 (1)
SsrOptimizationOptions	interface	core-libs/setup/ssr/optimized-engine/ssr-optimization-options.ts:L18-L181	07 (1), 39 (4), 44 (1), 47 (1)
StandardCmsComponentConfig	interface	core-libs/core/src/cms/config/cms-config.ts:L13-L31	15 (1)
State	interface	core-libs/core/src/routing/store/routing-state.ts:L27-L29	05 (2), 19 (5)
StateConfig	class	core-libs/core/src/state/config/state-config.ts:L24-L35	19 (4), 24 (1)
StateEventService	class	core-libs/core/src/state/event/state-event.service.ts:L22-L77	19 (4), 20 (9), 22 (1), 50 (3)
StateModule	class	core-libs/core/src/state/state.module.ts:L11-L18	02 (1), 13 (2), 19 (6)
StatePersistenceService	class	core-libs/core/src/state/services/state-persistence.service.ts:L21-L132	19 (7), 25 (3), 34 (2), 35 (1), 36 (3)
StateTransferType	enum	core-libs/core/src/state/config/state-config.ts:L16-L18	07 (1), 13 (2), 15 (1), 19 (4), 24 (1)
StateWithAsm	interface	feature-libs/asm/core/store/asm-state.ts:L15-L17	29 (1)
StateWithCms	interface	core-libs/core/src/cms/store/cms-state.ts:L16-L18	15 (1)
StateWithProcess	interface	core-libs/core/src/process/store/process-state.ts:L11-L13	19 (1)
STATUS	enum	feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts:L14-L18	31 (1)
STATUS_NAME	enum	feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts:L20-L24	31 (1)
Stock	interface	core-libs/core/src/model/product.model.ts:L77-L81 (+1)	46 (2), 54 (1)
StockAdapter	class	feature-libs/pickup-in-store/core/connectors/stock.adapter.ts:L14-L34	32 (1)
StockConnector	class	feature-libs/pickup-in-store/core/connectors/stock.connector.ts:L17-L43	32 (1)
StockLevel	class	feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L27-L33	32 (1)
StockLevelAtStore	const	feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L87-L90	32 (1)
StockLevelFail	class	feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L43-L52	32 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
StockLevelOnHold	class	feature-libs/pickup-in-store/core/store/actions/stock.action.ts:L35-L41	32 (1)
StockNotificationComponent	class	core-libs/storefront/cms-components/product/stock-notification/stock-notification.component.ts:L52-L202	24 (1)
StorageSyncType	enum	core-libs/core/src/state/config/state-config.ts:L10-L14	19 (4)
STORE	const	core-libs/schematics/src/shared/constants.ts:L50-L50	14 (1)
STORE_COUNT_NORMALIZER	const	feature-libs/storefinder/core/connectors/converters.ts:L15-L17 (+1)	32 (2)
STORE_FINDER_DATA	const	feature-libs/storefinder/core/store/store-finder-state.ts:L11-L11 (+1)	32 (1)
STORE_FINDER_FEATURE	const	feature-libs/storefinder/root/feature-name.ts:L7-L7 (+2)	32 (1)
STORE_FINDER_SEARCH_PAGE_NORMALIZER	const	feature-libs/storefinder/core/connectors/converters.ts:L11-L13 (+1)	32 (2)
StoreEntities	interface	feature-libs/storefinder/root/model/store-entities.model.ts:L13-L15	32 (1)
StoreFinderAdapter	class	feature-libs/storefinder/core/connectors/store-finder.adapter.ts:L11-L22	32 (1)
StoreFinderComponent	class	feature-libs/storefinder/components/store-finder/store-finder.component.ts:L16-L16	32 (4)
StoreFinderComponentsModule	class	feature-libs/storefinder/components/store-finder-components.module.ts:L117-L117	32 (1)
StoreFinderConfig	class	feature-libs/storefinder/core/config/store-finder-config.ts:L14-L23	32 (1)
StoreFinderCoreModule	class	feature-libs/storefinder/core/store-finder-core.module.ts:L22-L22	32 (1)
StoreFinderFacade	class	feature-libs/storefinder/root/facade/store-finder.facade.ts:L43-L73	32 (3)
StoreFinderGridComponent	class	feature-libs/storefinder/components/store-finder-grid/store-finder-grid.component.ts:L29-L53	32 (2)
StoreFinderListComponent	class	feature-libs/storefinder/components/store-finder-search-result/store-finder-list/store-finder-list.component.ts:L53-L123	04 (1), 32 (2)
StoreFinderListItemComponent	class	feature-libs/storefinder/components/store-finder-list-item/store-finder-list-item.component.ts:L21-L52	32 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
StoreFinderMapComponent	class	feature-libs/storefinder/components/store-finder-map/store-finder-map.component.ts:L23-L61	32 (2)
StoreFinderModule	class	feature-libs/storefinder/store-finder.module.ts:L19-L19	29 (1), 32 (1)
StoreFinderRootModule	class	feature-libs/storefinder/root/store-finder-root.module.ts:L38-L38	32 (1)
StoreFinderSearchComponent	class	feature-libs/storefinder/components/store-finder-search/store-finder-search.component.ts:L41-L72	32 (1)
StoreFinderSearchResultComponent	class	feature-libs/storefinder/components/store-finder-search-result/store-finder-search-result.component.ts:L32-L111	32 (2)
StoreFinderService	class	feature-libs/storefinder/core/facade/store-finder.service.ts:L29-L260	32 (3)
StoreFinderStoreComponent	class	feature-libs/storefinder/components/store-finder-store/store-finder-store.component.ts:L32-L63	32 (2)
StoreFinderStoreModule	class	feature-libs/storefinder/core/store/store-finder-store.module.ts:L27-L27	32 (1)
StoreFinderStoresCountComponent	class	feature-libs/storefinder/components/store-finder-stores-count/store-finder-stores-count.component.ts:L32-L51	32 (2)
StorefrontComponent	class	core-libs/storefront/layout/main/storefront.component.ts:L67-L229	02 (8), 10 (1), 18 (4), 38 (2), 42 (3), 43 (1), 45 (1)
StorefrontComponentModule	class	core-libs/storefront/layout/main/storefront-component.module.ts:L34-L34	02 (3), 10 (1), 18 (2)
StorefrontOutlets	enum	core-libs/storefront/layout/main/storefront-outlets.model.ts:L7-L9	18 (1)
StoreLocationService	class	feature-libs/storefinder/core/service/store-location.service.ts:L13-L41	32 (1)
StoreModule	class	feature-libs/pickup-in-store/components/presentational/store/store.module.ts:L29-L29	02 (1)
StrategyRequest	interface	integration-libs/cds/src/cds-models/cds-strategy-request.model.ts:L12-L28	34 (2)
StringTemplate	class	core-libs/core/src/config/utils/string-template.ts:L7-L32	11 (3)
STRONG_PASSWORD_PATTERN	const	core-libs/core/src/util/regex-pattern.ts:L29-L30	46 (1)
StructuredDataFactory	class	core-libs/storefront/cms-structure/seo/structured-data/structured-data.factory.ts:L20-L54	40 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
StructuredDataModule	class	core-libs/storefront/cms-structure/seo/structured-data/structured-data.module.ts:L36-L36	40 (5)
SubListComponent	class	feature-libs/organization/administration/components/shared/sub-list/sub-list.component.ts:L43-L73	28 (2)
SubListModule	class	feature-libs/organization/administration/components/shared/sub-list/sub-list.module.ts:L34-L34	28 (1)
SubListService	class	feature-libs/organization/administration/components/shared/sub-list/sub-list.service.ts:L19-L64	28 (2)
SubscriptionActionMode	type	feature-libs/subscription-billing/root/model/subscription.model.ts:L54-L58	31 (1)
SubscriptionActionsFacade	class	feature-libs/subscription-billing/root/facade/subscription-actions.facade.ts:L33-L56	31 (2)
SubscriptionActionsModalComponent	class	feature-libs/subscription-billing/components/actions-modal/subscription-actions-modal.component.ts:L61-L270	04 (3), 31 (1)
SubscriptionActionsService	class	feature-libs/subscription-billing/core/facade/subscription-actions.service.ts:L25-L160	31 (3)
SubscriptionBillingDetailsPageMetaResolver	class	feature-libs/subscription-billing/core/services/subscription-billing-details-page-meta.resolver.ts:L24-L72	31 (1)
SubscriptionBillingFacade	class	feature-libs/subscription-billing/root/facade/subscription-billing.facade.ts:L29-L53	31 (2)
SubscriptionBillingModule	class	feature-libs/subscription-billing/subscription-billing.module.ts:L18-L18	31 (1)
SubscriptionBillingService	class	feature-libs/subscription-billing/core/facade/subscription-billing.service.ts:L35-L166	31 (1)
SubscriptionCartPriceBodyComponent	class	feature-libs/subscription-billing/root/components/cart/price-body/subscription-cart-price-body.component.ts:L21-L34	31 (1)
SubscriptionCartPriceHeadingComponent	class	feature-libs/subscription-billing/root/components/cart/price-heading/subscription-cart-price-heading.component.ts:L22-L31	31 (1)
SubscriptionDetail	interface	feature-libs/subscription-billing/root/model/subscription.model.ts:L11-L28	31 (1)
SubscriptionDetailsComponent	class	feature-libs/subscription-billing/components/details/subscription-details.component.ts:L43-L75	31 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
SubscriptionDetailsPageMetaResolver	class	feature-libs/subscription-billing/core/services/subscription-details-page-meta.resolver.ts:L24-L70	31 (1)
SubscriptionFacade	class	feature-libs/subscription-billing/root/facade/subscription.facade.ts:L28-L50	31 (2)
SubscriptionListComponent	class	feature-libs/subscription-billing/components/list/subscription-list.component.ts:L39-L79	04 (2)
SubscriptionProductPriceComponent	class	feature-libs/subscription-billing/components/product/price/subscription-product-price.component.ts:L29-L52	31 (1)
SubscriptionProductService	class	feature-libs/subscription-billing/root/services/subscription-product.service.ts:L15-L29	31 (1)
SubscriptionService	class	feature-libs/subscription-billing/core/facade/subscription.service.ts:L27-L147	31 (1), 50 (1)
SupplementHashAnchorsPipe	class	core-libs/storefront/shared/pipes/suplement-hash-anchors/supplement-hash-anchors.pipe.ts:L22-L50	45 (1), 51 (1)

T

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
Tab	interface	core-libs/storefront/cms-components/content/tab/tab.model.ts:L11-L37 (+1)	45 (1)
TabConfig	interface	core-libs/storefront/cms-components/content/tab/tab.model.ts:L39-L63	45 (2)
TabFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L98-L100	18 (1)
TabFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/tab/tab-focus.service.ts:L14-L160	18 (1)
TableConfig	class	core-libs/storefront/shared/components/table/config/table.config.ts:L24-L43	28 (2), 45 (1)
TableModule	class	core-libs/storefront/shared/components/table/table.module.ts:L30-L30	45 (1)
TableService	class	core-libs/storefront/shared/components/table/table.service.ts:L34-L186	45 (1)
TabModule	class	core-libs/storefront/cms-components/content/tab/tab.module.ts:L17-L17	45 (1)
TabParagraphContainerComponent	class	core-libs/storefront/cms-components/content/tab-paragraph-container/tab-paragraph-container.component.ts:L57-L220	45 (3)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
TEST_CONFIG	const	core-libs/core/src/config/test-config.module.ts:L42-L55	11 (4), 39 (1), 42 (1), 47 (2)
TEST_CONFIG_COOKIE_NAME	const	core-libs/core/src/config/test-config.module.ts:L31-L33	11 (3), 47 (1)
TestConfig	interface	core-libs/core/src/config/test-config.module.ts:L27-L29	11 (1)
TestConfigModule	class	core-libs/core/src/config/test-config.module.ts:L75-L106	10 (1), 11 (6), 42 (1), 44 (4), 47 (6), 50 (1)
TestConfigModuleOptions	interface	core-libs/core/src/config/test-config.module.ts:L65-L67	02 (1)
TestConfigServerModule	class	core-libs/setup/ssr/testing/test-config-server.module.ts:L37-L58	07 (1), 39 (2), 47 (2)
TestingTimeUtils	class	core-libs/core/src/util/testing-time-utils.ts:L7-L21	46 (3)
TextfieldConfiguratorModule	class	feature-libs/product-configurator/textfield/textfield-configurator.module.ts:L23-L23	30 (2)
TextfieldConfiguratorRootModule	class	feature-libs/product-configurator/textfield/root/textfield-configurator-root.module.ts:L58-L64	30 (1)
THEME_CONTEXT_ID	const	core-libs/core/src/site-context/providers/context-ids.ts:L10-L10	13 (1), 38 (1)
ThemeModule	class	core-libs/storefront/layout/theme/theme.module.ts:L25-L25	13 (1), 18 (1), 38 (4)
ThemeService	class	core-libs/storefront/layout/theme/theme.service.ts:L19-L62	13 (2), 38 (9)
TicketClosedEvent	class	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L33-L35	31 (1)
TicketCreatedEvent	class	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L52-L54	31 (2)
TicketDetails	interface	feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts:L26-L35	31 (3)
TicketEvent	interface	feature-libs/customer-ticketing/root/model/customer-ticketing.model.ts:L61-L69	31 (2)
TicketReopenedEvent	class	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L29-L31	31 (2)
Time	interface	core-libs/core/src/model/misc.model.ts:L17-L21 (+1)	46 (1)
TimeUtils	class	core-libs/core/src/util/time-utils.ts:L7-L39	46 (1)
Title	interface	core-libs/core/src/model/misc.model.ts:L87-L90 (+2)	46 (1)
TITLE_NORMALIZER	const	feature-libs/user/profile/core/connectors/converters.ts:L28-L30 (+1)	23 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
TmsCollector	interface	feature-libs/tracking/tms/core/model/tms.model.ts:L21-L40	33 (4)
TmsCollectorConfig	interface	feature-libs/tracking/tms/core/config/tms-config.ts:L14-L29	33 (2)
TmsCollectors	interface	feature-libs/tracking/tms/core/config/tms-config.ts:L31-L33	33 (7)
TmsConfig	class	feature-libs/tracking/tms/core/config/tms-config.ts:L42-L47	33 (6)
TmsService	class	feature-libs/tracking/tms/core/services/tms.service.ts:L28-L116	33 (5)
ToggleAnonymousConsentsBannerDissmissed	class	core-libs/core/src/anonymous-consents/store/actions/anonymous-consents.action.ts:L109-L113	23 (1)
TokenRevocationInterceptor	class	core-libs/core/src/auth/user-auth/http-interceptors/token-revocation.interceptor.ts:L23-L59	06 (6), 22 (6)
TokenTarget	enum	feature-libs/asm/root/services/asm-auth-storage.service.ts:L18-L21	29 (2)
TrackingModule	class	integration-libs/cds/src/profiletag/tracking/tracking.module.ts:L20-L20	34 (3)
TRANSFER_ANONYMOUS_CONSENT	const	core-libs/core/src/user/store/actions/user-consents.action.ts:L27-L27	23 (1)
TRANSFER_STATE_META_REDUCER	const	core-libs/core/src/state/reducers/index.ts:L23-L25	07 (4), 19 (4), 54 (2)
TransferAnonymousConsent	class	core-libs/core/src/user/store/actions/user-consents.action.ts:L112-L122	23 (1)
Translatable	interface	core-libs/core/src/i18n/translatable.ts:L7-L11	37 (3), 41 (1)
TranslatePipe	class	core-libs/core/src/i18n/translate.pipe.ts:L29-L96	37 (5), 45 (1), 53 (7)
TranslationChunksConfig	interface	core-libs/assets/src/translations/translation-chunks-config.ts:L7-L9 (+1)	26 (1), 27 (2), 29 (1), 36 (1), 37 (4), 53 (4)
TranslationChunkService	class	core-libs/core/src/i18n/translation-chunk.service.ts:L14-L66	37 (4), 53 (2)
TranslationResourceKey	interface	core-libs/core/src/i18n/translation-resources.ts:L7-L9	37 (1)
TranslationResources	interface	core-libs/core/src/i18n/translation-resources.ts:L11-L15	37 (2), 53 (4)
TranslationService	class	core-libs/core/src/i18n/translation.service.ts:L9-L35	37 (6), 52 (1)
TrapFocus	enum	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L102-L117	18 (1)
TrapFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L135-L140	18 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
TrapFocusDirective	class	core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.directive.ts:L17-L66	18 (1)
TrapFocusService	class	core-libs/storefront/layout/a11y/keyboard-focus/trap/trap-focus.service.ts:L19-L74	18 (1)
TreeFileSystem	class	core-libs/schematics/src/shared/utils/tree-file-system.ts:L12-L157	08 (3), 43 (1)
TrendingSearchesComponent	class	integration-libs/cds/src/trending-searches/trending-searches.component.ts:L33-L74	34 (1)
TrendingSearchesModule	class	integration-libs/cds/src/trending-searches/trending-searches.module.ts:L37-L37	34 (2)
TrendingSearchesService	class	integration-libs/cds/src/trending-searches/trending-searches.service.ts:L37-L94	34 (1)
Trigger	interface	feature-libs/order/root/model/replenishment-order.model.ts:L78-L81 (+1)	27 (1)
TruncationTooltipDirective	class	core-libs/storefront/layout/a11y/truncation-tooltip/truncation-tooltip.directive.ts:L30-L65	18 (1)
Type	enum	core-libs/core/src/occ/occ-models/occ.models.ts:L2949-L2953	02 (1), 03 (3), 45 (2)

U

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
UiType	enum	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L317-L343 (+1)	30 (2)
UnifiedInjector	class	core-libs/core/src/lazy-loading/unified-injector.ts:L27-L94	01 (7), 03 (11), 05 (2), 11 (3), 12 (7), 14 (1), 16 (2), 21 (1), 26 (2), 28 (4), 31 (1), 40 (2), 41 (1), 46 (1), 50 (1), 53 (2)
UnitAddressItemService	class	feature-libs/organization/administration/components/unit/links/addresses/services/unit-address-item.service.ts:L23-L106	28 (1), 54 (1)
UnitLevelOrdersViewerGuard	class	feature-libs/organization/unit-order/core/guards/unit-level-orders-viewer.guard.ts:L23-L55	28 (1)
UnitListComponent	class	feature-libs/organization/administration/components/unit/list/unit-list.component.ts:L19-L36	28 (1)
UnitListService	class	feature-libs/organization/administration/components/unit/services/unit-list.service.ts:L36-L190	28 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
UnitOrderFacade	class	feature-libs/organization/unit-order/root/facade/unit-order.facade.ts:L34-L82	28 (4)
UnitTreeService	class	feature-libs/organization/administration/components/unit/services/unit-tree.service.ts:L22-L139	28 (1)
UnitUserListComponent	class	feature-libs/organization/administration/components/unit/links/users/list/unit-user-list.component.ts:L40-L53	34 (1)
UnitUserRolesCellComponent	class	feature-libs/organization/administration/components/unit/links/users/list/unit-user-link-cell.component.ts:L36-L49	34 (1)
UnknownErrorHandler	class	core-libs/core/src/global-message/http-interceptors/handlers/unknown-error/unknown-error.handler.ts:L22-L51	41 (11), 50 (1)
UPDATE_CART_ENTRY	const	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L28-L28	30 (2)
UPDATE_CART_ENTRY_SUCCESS	const	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L29-L30	30 (1)
UPDATE_CONFIGURATION	const	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L23-L23 (+1)	30 (1)
UPDATE_CONFIGURATION_FINALIZE_SUCCESS	const	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L45-L46	30 (1)
UPDATE_USER_ADDRESS	const	core-libs/core/src/user/store/actions/user-addresses.action.ts:L20-L20	23 (1)
UpdateBudget	class	feature-libs/organization/administration/core/store/actions/budget.action.ts:L132-L140	28 (2)
UpdateCartEntry	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator-cart.action.ts:L98-L105	30 (1)
UpdateConfiguration	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L178-L185 (+1)	30 (3)
UpdateConfigurationFail	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L187-L201	30 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
UpdateConfigurationFinalizeFail	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L311-L320	30 (2)
UpdateConfigurationFinalizeSuccess	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L303-L309	30 (2)
UpdateConfigurationSuccess	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L203-L209	30 (1)
UpdateEmailComponent	class	feature-libs/user/profile/components/update-email/update-email.component.ts:L55-L67	34 (1)
UpdatePasswordComponent	class	feature-libs/user/profile/components/update-password/update-password.component.ts:L55-L74	34 (1)
UpdatePriceSummary	class	feature-libs/product-configurator/rulebased/core/state/actions/configurator.action.ts:L322-L328	30 (4)
UpdateProfileComponent	class	feature-libs/user/profile/components/update-profile/update-profile.component.ts:L62-L82	34 (1)
UpdateProfileComponentService	class	feature-libs/user/profile/components/update-profile/update-profile-component.service.ts:L20-L83	23 (1)
UpdateType	enum	feature-libs/product-configurator/rulebased/core/model/configurator.model.ts:L364-L368	30 (1)
UpdateUserAddress	class	core-libs/core/src/user/store/actions/user-addresses.action.ts:L84-L92	23 (2)
UpdateUserAddressEvent	class	core-libs/core/src/user/events/user.events.ts:L14-L18	23 (3), 26 (1)
UploadAttachmentSuccessEvent	class	feature-libs/customer-ticketing/root/events/customer-ticketing.events.ts:L59-L61	31 (3)
URL	const	core-libs/schematics/src/shared/constants.ts:L520-L520	41 (1)
URL_TESTING_ALLOWLISTED_PARAMS	const	core-libs/core/src/routing/configurable-routes/url-translation/testing/mock-url.pipe.ts:L16-L18	14 (1)
UrlCommand	type	core-libs/core/src/routing/configurable-routes/url-translation/url-command.ts:L12-L12	14 (1)
UrlCommandRoute	interface	core-libs/core/src/routing/configurable-routes/url-translation/url-command.ts:L7-L10	14 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
UrlCommands	type	core-libs/core/src/routing/configurable-routes/url-translation/url-command.ts:L14-L14	14 (1)
UrlMatcherFactory	type	core-libs/core/src/routing/url-matcher/url-matcher-factory.ts:L9-L9	05 (2), 14 (1), 16 (1)
UrlMatcherService	class	core-libs/core/src/routing/services/url-matcher.service.ts:L18-L198	05 (2), 14 (3)
UrlModule	class	core-libs/core/src/routing/configurable-routes/url-translation/url.module.ts:L16-L16	14 (1)
UrlParsingService	class	core-libs/core/src/routing/configurable-routes/url-translation/url-parsing.service.ts:L12-L64	14 (1)
UrlPipe	class	core-libs/core/src/routing/configurable-routes/url-translation/url.pipe.ts:L12-L18	14 (1), 51 (1)
UrlTestingModule	class	core-libs/core/src/routing/configurable-routes/url-translation/testing/url-testing.module.ts:L15-L29	14 (2)
UrlTreeWithSiteContext	interface	core-libs/core/src/site-context/services/site-context-url-serializer.ts:L22-L24	13 (2)
UsageIdConfig	interface	integration-libs/epd-visualization/root/config/epd-visualization-config.ts:L77-L94	36 (1)
USE_AUTHORIZATION_CODE_FLOW_BY_DEFAULT	const	core-libs/core/src/auth/user-auth/config/default-auth-config.ts:L15-L19	22 (2), 42 (1)
USE_CAPTCHA_TOKEN	const	core-libs/core/src/occ/utils/interceptor-util.ts:L11-L11	06 (1), 21 (2)
USE_CLIENT_TOKEN	const	core-libs/core/src/occ/utils/interceptor-util.ts:L9-L9	06 (1), 21 (2), 22 (2), 23 (4), 25 (4), 27 (2), 28 (1), 32 (3), 36 (1)
USE_CUSTOMER_SUPPORT_AGENT_TOKEN	const	core-libs/core/src/occ/utils/interceptor-util.ts:L10-L10	06 (1), 21 (1), 29 (1)
USE_MY_ACCOUNT_V2_CONSENT	const	core-libs/storefront/cms-components/myaccount/my-account-v2/use-my-account-v2-consent-notification-perference.ts:L15-L18	23 (1), 47 (1)
USE_MY_ACCOUNT_V2_EMAIL	const	feature-libs/user/profile/components/update-email/use-my-account-v2-email.ts:L9-L12	23 (1)
USE_MY_ACCOUNT_V2_NOTIFICATION_PREFERENCE	const	core-libs/storefront/cms-components/myaccount/my-account-v2/use-my-account-v2-consent-notification-perference.ts:L9-L13	23 (1), 47 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
USE_MY_ACCOUNT_V2_ORDER	const	feature-libs/order/root/tokens/context.ts:L17-L20	27 (4)
USE_MY_ACCOUNT_V2_PASSWORD	const	feature-libs/user/profile/components/update-password/use-my-account-v2-password.ts:L9-L12	23 (1)
USE_MY_ACCOUNT_V2_PROFILE	const	feature-libs/user/profile/components/update-profile/use-my-account-v2-profile.ts:L9-L12	23 (1)
USE_STACKED_OUTLETS	const	core-libs/storefront/cms-structure/outlet/outlet.model.ts:L16-L16	17 (4)
useFeatureStyles()	function	core-libs/core/src/features-config/utils/use-feature-styles.ts:L32-L40	42 (1)
User	interface	core-libs/core/src/model/misc.model.ts:L40-L58 (+3)	23 (2), 29 (4), 46 (3)
USER_ACCOUNT_FEATURE	const	feature-libs/user/account/root/feature-name.ts:L7-L7	23 (1)
USER_ACCOUNT_NORMALIZER	const	feature-libs/user/account/core/connectors/converters.ts:L15-L17 (+1)	23 (1)
USER_ADDRESSES	const	core-libs/core/src/user/store/user-state.ts:L33-L33	23 (1)
USER_FEATURE	const	core-libs/core/src/user/store/user-state.ts:L22-L22	23 (1)
USER_SIGN_UP_SERIALIZER	const	feature-libs/user/profile/core/connectors/converters.ts:L24-L26 (+1)	23 (1)
UserAccountAdapter	class	feature-libs/user/account/core/connectors/user-account.adapter.ts:L14-L20	23 (1)
UserAccountChangedEvent	class	feature-libs/user/account/root/events/user-account.events.ts:L14-L16	23 (6)
UserAccountComponentsModule	class	feature-libs/user/account/components/user-account-component.module.ts:L27-L27	23 (1)
UserAccountConfig	class	feature-libs/user/account/root/config/user-account-config.ts:L14-L23	23 (1)
UserAccountCoreModule	class	feature-libs/user/account/core/user-account-core.module.ts:L14-L14	23 (1)
UserAccountEventListener	class	feature-libs/user/account/root/events/user-account-event.listener.ts:L22-L49	23 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
UserAccountEventModule	class	feature-libs/user/account/root/events/user-account-event.module.ts:L12-L18	23 (2)
UserAccountFacade	class	feature-libs/user/account/root/facade/user-account.facade.ts:L22-L25	23 (3)
UserAccountModule	class	feature-libs/user/account/user-account.module.ts:L19-L19	23 (2), 34 (1)
UserAccountOccModule	class	feature-libs/user/account/occ/user-account-occ.module.ts:L19-L19	23 (1)
UserAccountRootModule	class	feature-libs/user/account/root/user-account-root.module.ts:L75-L75	23 (2)
UserAccountService	class	feature-libs/user/account/core/facade/user-account.service.ts:L25-L57	23 (4)
UserAddressAdapter	class	core-libs/core/src/user/connectors/address/user-address.adapter.ts:L10-L27	23 (1)
UserAddressService	class	core-libs/core/src/user/facade/user-address.service.ts:L33-L319	23 (3)
UserAuthEventBuilder	class	core-libs/core/src/auth/user-auth/events/user-auth-event.builder.ts:L20-L64	20 (1), 22 (2)
UserAuthEventModule	class	core-libs/core/src/auth/user-auth/events/user-auth-event.module.ts:L11-L15	22 (1)
UserAuthModule	class	core-libs/core/src/auth/user-auth/user-auth.module.ts:L101-L124	05 (1), 06 (3), 10 (1), 22 (4), 42 (1)
UserComponentModule	class	core-libs/storefront/cms-components/user/user.module.ts:L23-L23	23 (1)
UserConsentAdapter	class	core-libs/core/src/user/connectors/consent/user-consent.adapter.ts:L10-L24	23 (1), 34 (3), 52 (1)
UserConsentService	class	core-libs/core/src/user/facade/user-consent.service.ts:L40-L308	23 (2)
UserCostCenterAdapter	class	core-libs/core/src/user/connectors/cost-center/user-cost-center.adapter.ts:L11-L19	23 (1)
UserCurrencyPreferenceSaverService	class	feature-libs/user/profile/root/services/user-currency-preference-saver.service.ts:L22-L62	23 (2)
UserDetailsComponent	class	feature-libs/organization/administration/components/user/details/user-details.component.ts:L54-L81	34 (2)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
UserEmailFacade	class	feature-libs/user/profile/root/facade/user-email.facade.ts:L21-L29	23 (2)
UserEventBuilder	class	core-libs/core/src/user/events/user-event.builder.ts:L17-L54	23 (4), 50 (1)
UserEventModule	class	core-libs/core/src/user/events/user-event.module.ts:L11-L15	23 (3)
UserGroupItemService	class	feature-libs/organization/administration/components/user-group/services/user-group-item.service.ts:L22-L67	28 (1)
UserGuard	class	feature-libs/organization/administration/core/guards/user.guard.ts:L17-L38	28 (5), 34 (1)
UserIdHttpHeaderInterceptor	class	feature-libs/asm/root/interceptors/user-id-http-header.interceptor.ts:L30-L85	06 (5), 21 (2), 29 (4), 54 (1)
UserIdService	class	core-libs/core/src/auth/user-auth/facade/user-id.service.ts:L24-L89	21 (1), 22 (4), 23 (4), 29 (2)
UserInterestsAdapter	class	core-libs/core/src/user/connectors/interests/user-interests.adapter.ts:L14-L57	23 (1)
UserListService	class	feature-libs/organization/administration/components/user/services/user-list.service.ts:L41-L96	34 (1)
UserLoginCurrencyService	class	feature-libs/user/account/root/services/user-login-currency.service.ts:L25-L80	23 (1)
UserModule	class	core-libs/core/src/user/user.module.ts:L21-L27	23 (3)
UserNotificationPreferenceAdapter	class	core-libs/core/src/user/connectors/notification-preference/user-notification-preference.adapter.ts:L10-L28	23 (1)
UserOccModule	class	core-libs/core/src/occ/adapters/user/user-occ.module.ts:L72-L72	21 (2), 23 (4), 34 (1)
UserPasswordFacade	class	feature-libs/user/profile/root/facade/user-password.facade.ts:L21-L48	23 (2)
UserPaymentAdapter	class	core-libs/core/src/user/connectors/payment/user-payment.adapter.ts:L10-L16	23 (1), 35 (4), 52 (1)
UserPaymentService	class	core-libs/core/src/user/facade/user-payment.service.ts:L20-L98	23 (1)
UserProfileAdapter	class	feature-libs/user/profile/core/connectors/user-profile.adapter.ts:L11-L42	23 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
UserProfileFacade	class	feature-libs/user/profile/root/facade/user-profile.facade.ts:L23-L42	23 (4), 25 (1)
UserProfileFacadeTransitionalToken	class	core-libs/core/src/user/user-transitional-tokens.ts:L10-L12	23 (3)
UserProfileModule	class	feature-libs/user/profile/user-profile.module.ts:L21-L21	23 (2)
UserProfileRootModule	class	feature-libs/user/profile/root/user-profile-root.module.ts:L54-L58	23 (1)
UserRegisterFacade	class	feature-libs/user/profile/root/facade/user-register.facade.ts:L23-L43	23 (2)
UserRegisterService	class	feature-libs/user/profile/core/facade/user-register.service.ts:L29-L97	23 (2)
UserRegistrationFacade	class	feature-libs/organization/user-registration/root/facade/user-registration.facade.ts:L22-L31	28 (1)
UserRegistrationFormService	class	feature-libs/organization/user-registration/components/form/user-registration-form.service.ts:L38-L209	34 (2)
UserRegistrationService	class	feature-libs/organization/user-registration/core/facade/user-registration.service.ts:L17-L42	28 (1)
UserSignUp	interface	feature-libs/user/profile/root/model/user-profile.model.ts:L33-L41 (+1)	34 (1)
UserState	interface	core-libs/core/src/user/store/user-state.ts:L52-L65	23 (2)
UserStoreModule	class	core-libs/core/src/user/store/user-store.module.ts:L27-L27	23 (3)

V

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
VARIANT_CONFIGURATOR_NORMALIZER	const	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L12-L14	30 (1)
VARIANT_CONFIGURATOR_SERIALIZER	const	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.converters.ts:L16-L18	30 (1)
VariantConfiguratorOccAdapter	class	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.adapter.ts:L40-L407	30 (1)

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
VariantConfiguratorOccModule	class	feature-libs/product-configurator/rulebased/occ/variant/variant-configurator-occ.module.ts:L85-L85	30 (4)
VariantConfiguratorPageLayoutHandler	class	feature-libs/product-configurator/rulebased/root/variant/variant-configurator-page-layout-handler.ts:L23-L72	30 (3)
VariantQualifier	enum	core-libs/core/src/model/product.model.ts:L222-L230	24 (1), 46 (1)
VariantType	enum	core-libs/core/src/model/product.model.ts:L57-L61	46 (2)
VerificationTokenFacade	class	feature-libs/user/account/root/facade/verification-token.facade.ts:L22-L26	23 (2)
VerificationTokenService	class	feature-libs/user/account/core/facade/verification-token.service.ts:L18-L39	23 (1)
VideoComponent	class	core-libs/storefront/cms-components/content/video/video.component.ts:L39-L124	45 (1)
VIEW_ALL_STORES	const	feature-libs/storefinder/core/store/actions/view-all-stores.action.ts:L11-L11	32 (1)
ViewAllStoresSuccess	class	feature-libs/storefinder/core/store/actions/view-all-stores.action.ts:L35-L41	32 (1)
ViewConfig	class	core-libs/storefront/shared/config/view-config.ts:L14-L26	24 (1), 31 (1), 45 (3)
VisibleFocusConfig	interface	core-libs/storefront/layout/a11y/keyboard-focus/keyboard-focus.model.ts:L19-L25	18 (1)
VisibleFocusDirective	class	core-libs/storefront/layout/a11y/keyboard-focus/visible/visible-focus.directive.ts:L25-L87	18 (1)
VisualizationAdapter	class	integration-libs/epd-visualization/core/connectors/visualization/visualization.adapter.ts:L14-L28	36 (1)
VisualPickingProductListService	class	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/product-list/visual-picking-product-list.service.ts:L37-L197	36 (1)
VisualPickingTabComponent	class	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.component.ts:L36-L72	36 (2)
VisualPickingTabService	class	integration-libs/epd-visualization/components/visual-picking/visual-picking-tab/visual-picking-tab.service.ts:L33-L234	36 (1)
VisualViewerComponent	class	integration-libs/epd-visualization/components/visual-viewer/visual-viewer.component.ts:L40-L307	36 (1)
VisualViewerService	class	integration-libs/epd-visualization/components/visual-viewer/visual-viewer.service.ts:L77-L1639	36 (5)

W

SIMBOLO	TIPO	DICHIARAZIONE	CAPITOLI (OCCORRENZE)
W3cTraceContext	interface	core-libs/setup/ssr/logger/loggers/w3c-trace-context/w3c-trace-context.model.ts:L18-L23	39 (2)
WebComponentHandler	class	core-libs/storefront/cms-structure/page/component/handlers/web-component.handler.ts:L30-L137	16 (6), 51 (2), 54 (1)
WindowObject	interface	feature-libs/tracking/tms/core/model/tms.model.ts:L14-L16	33 (2)
WindowRef	class	core-libs/core/src/window/window-ref.ts:L16-L108	07 (4), 13 (1), 19 (2), 23 (1), 31 (1), 33 (1), 36 (1), 46 (11), 50 (1)
Wishlist	interface	feature-libs/user/wishlist/root/model/wishlist.model.ts:L16-L19	25 (1)
WishListFacade	class	feature-libs/cart/wish-list/root/facade/wish-list.facade.ts:L30-L46	25 (1), 50 (1)
WishListService	class	feature-libs/cart/wish-list/core/facade/wish-list.service.ts:L33-L122	25 (1)
WishListV2Module	class	feature-libs/cart/wish-list/wish-list-v2.module.ts:L50-L50	25 (3), 50 (1)
WishListV2Service	class	feature-libs/cart/wish-list/core/facade/wish-list-v2.service.ts:L30-L138	25 (3)
WishListWithV2Module	class	feature-libs/cart/wish-list/wish-list-with-v2.module.ts:L14-L14	47 (1), 50 (1)
WithCredentialsInterceptor	class	core-libs/core/src/occ/interceptors/with-credentials.interceptor.ts:L21-L53	06 (4), 21 (5)

Legenda capitoli:

- 01 Dependency Injection in Angular e in Spartacus
- 02 NgModule, standalone e bootstrap dell'applicazione
- 03 Lazy loading: router e moduli caricati a runtime
- 04 Change detection, Zone, zoneless e signals
- 05 Il Router di Angular
- 06 HttpClient e interceptor
- 07 SSR, hydration e TransferState
- 08 Schematics e builder
- 10 Bootstrap e ordine di inizializzazione
- 11 Configurazione: provideConfig, default, merge, validator, initializer
- 12 Feature module e lazy loading in Spartacus
- 13 Site context e site theme
- 14 Routing: route semantiche, URL, page guard, contesto
- 15 CMS (core): pagine, slot, componenti, caricamento

- 16 CMS (storefront): rendering dinamico delle pagine
- 17 Outlet
- 18 Layout, breakpoint, launch dialog e accessibilità
- 19 State management: NgRx, loader/entity/process state
- 20 Facade, command & query, eventi
- 21 Livello OCC: adapter, connector, converter, endpoint
- 22 Autenticazione: OAuth, token, interceptor, refresh
- 23 Utente, profilo, consensi
- 24 Prodotto e ricerca
- 25 Carrello
- 26 Checkout
- 27 Ordini
- 28 B2B: organization
- 29 ASM (Assisted Service Module)
- 30 Product configurator
- 31 Quote, customer ticketing, subscription billing
- 32 Pickup in store, store finder, delivery date
- 33 Tracking, SmartEdit, Qualtrics
- 34 Integration libs: CDC, CDP, CDS, segment-refs
- 35 Integration libs: OPF, digital payments, OPPS
- 36 Integration libs: S/4, OMF, CPQ quote, punchout, EPD
- 37 Internazionalizzazione (i18n)
- 38 Stili e theming
- 39 SSR ottimizzato
- 40 SEO e PWA
- 41 Errori HTTP, error handling, logger e global message
- 42 Feature toggle e deprecazioni
- 43 Schematics e migrazioni di Spartacus
- 44 Strategia di test e tooling del monorepo
- 45 Componenti CMS e shared components di storefront
- 46 Utility, window e modelli di dominio
- 47 L'app demo storefrontapp
- 50 Ricette: configurazione, DI, interceptor, eventi
- 51 Ricette: componenti CMS, outlet, route e layout
- 52 Ricette: OCC, adapter, converter, endpoint, meta resolver
- 53 Ricette: feature lib lazy, carrello/checkout, stili, traduzioni
- 54 API pubblica vs interna, upgrade, anti-pattern, checklist



SBU.DEV · SPARTACUS SOTTO IL COFANO · SHA 5B7A1C2B