

---

# **Itk25nowTheme**

Entwicklerdokumentation

André Fischer

02.04.2026

## Inhaltsverzeichnis

|           |   |           |
|-----------|---|-----------|
| <b>1</b>  | <b>Einordnung</b>   | <b>2</b>  |
| <b>2</b>  | <b>Architekturüberblick</b>   | <b>3</b>  |
| <b>3</b>  | <b>Plugin-Hauptklasse</b>   | <b>4</b>  |
| <b>4</b>  | <b>theme.json (Parent-Theme)</b>  | <b>4</b>  |
| <b>5</b>  | <b>Custom Fields</b>  | <b>5</b>  |
| 5.1       | ThemeCustomFieldService .....   | 5         |
| 5.1.1     | Setproduct_box_settings (Relation: <b>category</b> ) .....              | 5         |
| 5.2       | CustomFieldInstaller .....  | 6         |
| <b>6</b>  | <b>System-Konfiguration (config.xml)</b>                                | <b>6</b>  |
| <b>7</b>  | <b>Dependency Injection (services.xml)</b>                              | <b>7</b>  |
| <b>8</b>  | <b>Storefront-Controller</b>  | <b>8</b>  |
| 8.1       | EasterEggController .....   | 8         |
| 8.2       | AccountDeleteController .....   | 9         |
| <b>9</b>  | <b>Twig-Extension: ItkTwigExtension (PHP-API)</b>                       | <b>9</b>  |
| <b>10</b> | <b>Weitere Subscriber und Services</b>                                  | <b>10</b> |
| 10.1      | ProductCustomFieldSubscriber .....                                      | 10        |
| 10.2      | ProductCustomFieldExcludeSubscriber .....                               | 11        |
| 10.3      | WishlistProductIdsPageSubscriber .....                                  | 11        |
| 10.4      | AccountTypeChangeSubscriber .....                                       | 11        |
| 10.5      | HolidayService .....  | 11        |
| 10.6      | CategoryCopyService/ itk:copy:category (Experimentell - nicht deployed) | 12        |
| <b>11</b> | <b>CLI-Befehle</b>  | <b>12</b> |
| <b>12</b> | <b>Storefront-JavaScript (Einstieg)</b>                                 | <b>12</b> |
| <b>13</b> | <b>Abhängigkeiten zur Laufzeit (Überblick)</b>                          | <b>13</b> |

## 1 Einordnung

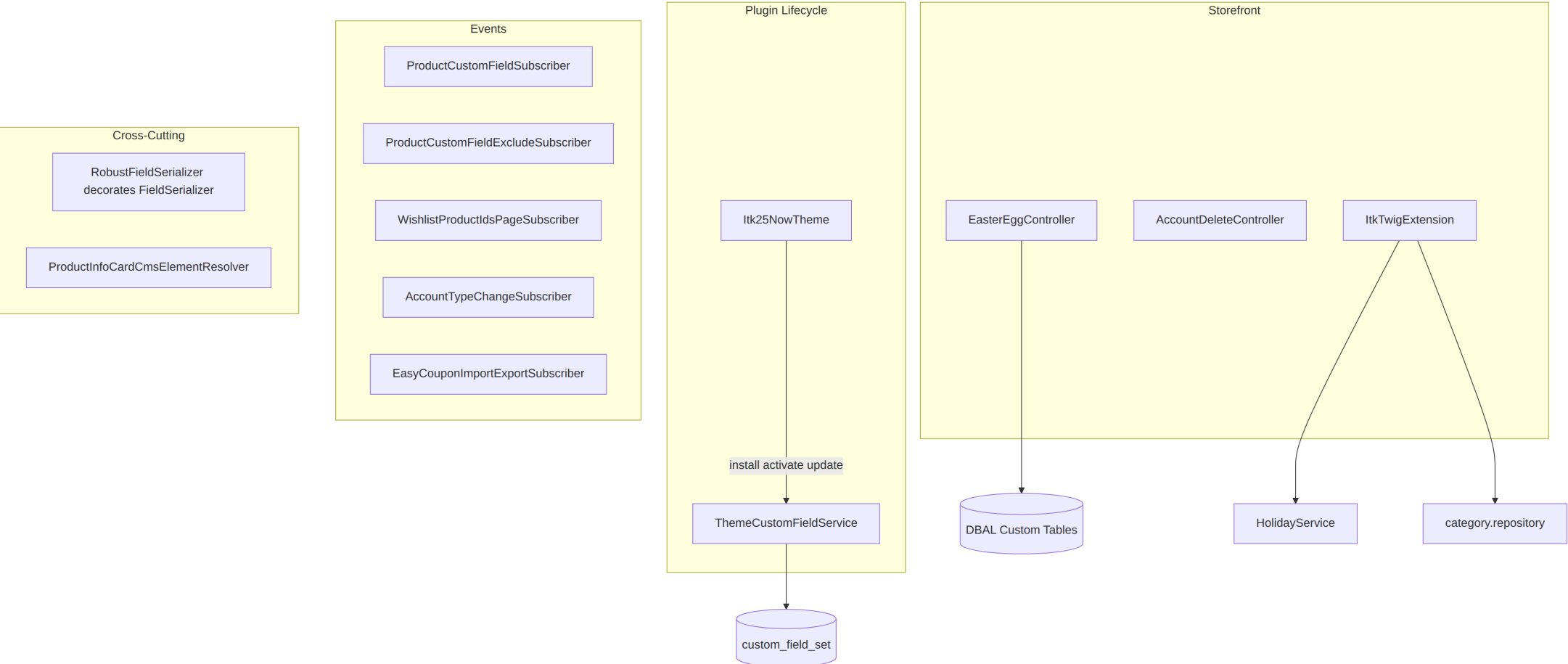
- **Shopware-Plugin-Klasse:** `ItkComputerGmbH\\Itk25NowTheme\\Itk25NowTheme`
- **PSR-4:** `ItkComputerGmbH\\Itk25NowTheme\\` → `custom/plugins/Itk25nowTheme/src/`
- **Theme:** implementiert `Shopware\\Storefront\\Framework\\ThemeInterface`; technischer Theme-Name laut `theme.json`: **25now**
- **Child-Theme:** `Itk25NowChildTheme` (UltraForce) erbt von diesem Theme (`parentTheme: Itk25NowTheme`).

Diese Doku beschreibt **PHP, DI, Konfiguration, Events, Import/Export, CMS-Resolver, Storefront-Controller, Twig-Extension (PHP-API), Migrationen, CLI und Logging**. Inhalte von **.twig-Dateien** werden nicht wiedergegeben; bei E-Mail-/View-Verwendung nur Bundle-Pfade genannt.

**Wichtig:** Die Doku bezieht sich auf den Staging Stand - nicht alles, was hier Dokumentiert ist, ist auch im Live deployed. Ist natürlich trotzdem sinnvoll einen Überblick zu haben, auch über Sachen die nicht deployed sind.

---

# 2 Architekturüberblick



### 3 Plugin-Hauptklasse

Datei: `custom/plugins/Itk25nowTheme/src/Itk25NowTheme.php`

| Methode           | Verhalten   |
|-------------------|---|
| <b>install</b>    | Ruft <code>installCustomFields()</code> auf.  |
| <b>activate</b>   | Ruft <code>installCustomFields()</code> auf (Anlegen/Aktualisieren der Produktbox-Custom Fields).   |
| <b>update</b>     | Ruft <code>installCustomFields()</code> auf.  |
| <b>deactivate</b> | Leer (Custom Fields bleiben).   |
| <b>uninstall</b>  | <code>parent::uninstall</code> ; wenn <b>kein</b> <code>keepUserData</code> , dann <code>uninstallCustomFields('product_box_settings', ...)</code> via <code>ThemeCustomFieldService</code> . |

**installCustomFields:** Liest aus dem Container `ThemeCustomFieldService` und ruft `installProductBoxCustomFields($context->getContext())` auf. Fehler → `logger` mit Meldung „Fehler bei der Installation der Custom Fields ...“.

### 4 theme.json (Parent-Theme)

Pfad: `custom/plugins/Itk25nowTheme/src/Resources/theme.json`

| Schlüssel     | Inhalt (Kurz)   |
|---------------|---|
| <b>name</b>   | <code>25now</code>  |
| <b>script</b> | <code>@Storefront, @Plugins, @Itk25NowTheme</code> – Theme-JS nach <code>Storefront/Plugins</code> .  |
| <b>views</b>  | <code>@Storefront, @Itk25NowTheme, @Plugins</code> – Theme-Templates <b>vor</b> <code>@Plugins</code> , damit Plugin-Overrides nach dem Theme greifen können. |
| <b>style</b>  | <code>@Storefront, @Plugins</code> , dann <code>app/storefront/src/scss/main.scss</code> .  |
| <b>asset</b>  | Eigene Assets + Fonts + <code>@Storefront</code> .  |

`\newPage{}`

## 5 Custom Fields

Wichtig anzumerken ist, dass das Theme nicht alle Custom Fields installiert. Eine Doku, welche Custom Fields vom Theme gebraucht werden wird noch benötigt.

### 5.1 ThemeCustomFieldService

Datei: `custom/plugins/Itk25nowTheme/src/Service/ThemeCustomFieldService.php`

- **installProductBoxCustomFields(Context)**  
Lädt `getProductBoxCustomFieldSetData()`, sucht per `EqualsFilter` auf Setname `product_box_settings`.
  - **Existiert:** `update` mit bestehender `id`; **relations** werden entfernt (`unset`), um Integritätskonflikte zu vermeiden; bestehende Felder erhalten ihre `id` für Update statt Duplicate-Create.
  - **Neu:** `create` des kompletten Sets.
- **installCustomerCustomFields(Context)**  
Analog für Set `customer_trusted_shops_settings` (Relation `customer`), Feld u. a. `trustedShopConsent`.
- **uninstallCustomFields(string \$setName, Context)**  
Löscht das Set per Name, falls vorhanden.

#### 5.1.1 Set product\_box\_settings (Relation: category)

| Feldname (Custom Field)                               | Typ               | Zweck (Kurz)                                    |
|---|-------------------|---|
| <code>custom_product_box_show_description</code>      | <code>bool</code> | Kurzbeschreibung in der Box                     |
| <code>custom_product_box_show_advantages</code>       | <code>bool</code> | Produktvorteile                                 |
| <code>custom_product_box_horizontal_layout</code>     | <code>bool</code> | Horizontales Layout                             |
| <code>custom_product_box_show_delivery_time</code>    | <code>bool</code> | Lieferzeit                                      |
| <code>custom_product_box_show_advantages_icons</code> | <code>bool</code> | Label: Icons bei Vorteilen<br><b>verstecken</b> |

**Hinweis:** `ItkTwigExtension::getBoxSettings()` liefert zusätzlich den Schlüssel **showDeliveryInAdvantages** aus `custom_product_box_show_delivery_in_advantages` – dieses Feld ist in `getProductBoxCustomFieldSetData()` **nicht** definiert; ohne manuelles Feld bleibt der Default **false**.

## 5.2 CustomFieldInstaller

Datei: `custom/plugins/Itk25nowTheme/src/Service/CustomFieldInstaller.php`

- **installCustomFields(string \$xmlFilePath, Context)** – parst XML, legt/aktualisiert ein Custom-Field-Set (Pfad muss existieren).
- **installAdditionalHolidaysField(Context)** – Set **itk25\_additional\_holidays**, Relation **sales\_channel**, Feld **custom\_additional\_holidays** (Text; Format laut CLI-Hilfe: YYYY-MM-DD semikolongetrennt). Wird von **itk:custom-fields:install** aufgerufen.

## 6 System-Konfiguration (config.xml)

Pfad: `custom/plugins/Itk25nowTheme/src/Resources/config/config.xml`

| Key  | Typ  | Pflicht | Verwendung   |
|--|------|---------|--|
| <b>Itk25Now Theme.config. companyEmail</b> | text | ja      | Support-Mail bei Kontolöschung (AccountDeleteController) |
| <b>Itk25Now Theme.config. company Name</b> | text | ja      | Absendername   |

Fallbacks im Code: `info@25now.de / 25NOW`, wenn Config leer.

## 7 Dependency Injection (services.xml)

Pfad: `custom/plugins/Itk25nowTheme/src/Resources/config/services.xml`

Auszug der wichtigsten Services (vollständige Definition im XML):

| Service / ID   | Klasse                            | Tag / Besonderheit   |
|--|-----------------------------------|--|
| ItkComputerGmbH\Itk25NowTheme\Twig\ItkTwigExtension                              | ItkTwigExtension                  | twig.extension   |
| ItkComputerGmbH\Itk25NowTheme\Core\Content\Cms\ProductInfoCardCmsElementResolver | ProductInfoCardCmsElementResolver | shopware.cms.data_resolver   |
| ThemeCustomFieldService / CustomFieldInstaller                                   | -                                 | normale Services   |
| CustomFieldInstallCommand  | -                                 | console.command; Argument: %kernel.project_dir%/custom/plugins/Itk25nowTheme     |
| ProductCustomFieldSubscriber   | -                                 | kernel.event_subscriber  |
| ProductCustomFieldExcludeSubscriber  | -                                 | kernel.event_subscriber  |
| WishlistProductIdsPageSubscriber   | -                                 | kernel.event_subscriber  |
| AccountTypeChangeSubscriber  | -                                 | kernel.event_subscriber  |
| EasyCouponImportExportSubscriber   | -                                 | kernel.event_subscriber  |
| EasterEggController  | -                                 | public="true", setContainer  |
| AccountDeleteController  | -                                 | public="true", setContainer, <a href="#">setTwig</a>                             |
| RegistrationIpLogService   | -                                 | DBAL <a href="#">Connection</a> (siehe Abschnitt <i>Bekannt Inkonsistenzen</i> ) |
| HolidayService   | -                                 | optional <a href="#">cache . app</a> , Logger                                    |
| RobustFieldSerializer  | -                                 | <b>decorates</b> FieldSerializer, decoration-priority="100"                      |

### Logger (Monolog, Level 100 = DEBUG):

| Logger-ID                  | Datei unter <code>var / log /</code> |
|----------------------------|--------------------------------------|
| itk_cms_ajax_modal_twig    | itk_cms_ajax_modal_twig.log          |
| itk_product_cf_edit_logger | itk_product_cf_edit_logger.log       |

| Logger-ID              | Datei unter <code>var/log/</code> |
|------------------------|-----------------------------------|
| itk_product_exclude    | itk_product_exclude.log           |
| itk_holiday_service    | itk_holiday_service.log           |
| itk_easy_coupon_import | itk_easy_coupon_import.log        |

## 8 Storefront-Controller

### 8.1 EasterEggController

- **Klasse:** `ItkComputerGmbH\Itk25NowTheme\Controller\EasterEggController`
- **Abhängigkeiten:** `Doctrine\DBAL\Connection`, `NetInventors\NetiNextEasyCoupon\Service\CustomerAssignmentService`, `neti_easy_coupon_customer_assignment.repository`
- **Voraussetzung:** Plugin **NetiNextEasyCoupon** muss verfügbar sein.

| Route-Name                    | Methode | Pfad                  | Verhalten (Kurz)   |
|-------------------------------|---------|-----------------------|--|
| frontend.itk.easter-egg-claim | POST    | /itk/easter-egg/claim | Body: <code>egg_id</code> . Nur eingeloggte Kunden. Tageslimit: max. <b>3</b> Zuweisungen pro Tag (Zählung über <code>_itk_easter_eggs.customer_assigned</code> = numerischer Wert aus <b>Kundennummer</b> ). Prüfung Sichtbarkeit <code>visible_from/visible_to</code> , <code>live/Session</code> . Bei Gutschein: UUID in <code>coupon</code> → Code aus <code>neti_easy_coupon</code> , ggf. Assignment + <code>triggerCustomerAssignmentEvent</code> . JSON: <code>success</code> , <code>message</code> , <code>coupon_code</code> . |

| Route-Name                                  | Methode | Pfad                                | Verhalten (Kurz)   |
|---|---------|-------------------------------------|--|
| frontend.itk.easter-egg.add-voucher-to-cart | POST    | /itk/easter-egg/add-voucher-to-cart | Body: <code>code</code> → CustomerAssignmentService::addVoucherToCartByCode. |

**Tabellen (Custom, nicht DAL):** `_itk_easter_eggs`, `_itk_easter_eggs_log` (Nutzung u. a. in Controller und `ItkTwigExtension`).

## 8.2 AccountDeleteController

- **Route:** POST `/account/delete-request`, Name `frontend.account.delete.request`
- Eingeloggt erforderlich; sonst Redirect Login + Flash.
- Versendet zwei Mails via `MailService` (Kontext `Context::createDefaultContext()`), Inhalte gerendert über:
  - `@Itk25NowTheme/storefront/emails/account-delete-support.html.twig` Mail an den Support
  - `@Itk25NowTheme/storefront/emails/account-delete-customer.html.twig` Mail an den Kunden

## 9 Twig-Extension: ItkTwigExtension (PHP-API)

Datei: `custom/plugins/Itk25nowTheme/src/Twig/ItkTwigExtension.php`

Tag: `twig.extension`

Injections: `category.repository`, `HolidayService`, `request_stack`, `Doctrine\DBAL\Connection`, `router`, `Logger itk_cms_ajax_modal_twig`.

Alle registrierten **TwigFunction**-Namen:

| Funktion                       | Rückgabe / Wirkung   |
|--------------------------------|--|
| <code>get_theme_version</code> | <code>string</code> : Version aus <code>composer.json</code> neben <code>src/</code> (unknown bei Fehlen). |

| Funktion                              | Rückgabe / Wirkung   |
|---------------------------------------|--|
| <b>is_darkmode_enabled</b>            | <code>bool</code> : prüft <code>custom_theme_settings_use_darkmode</code> in Produkt-, Kategorie-, Landingpage-, Hersteller-Übersetzungen, Sales Channel, ggf. Kategorie-Hierarchie / SEO-Kategorie.   |
| <b>is_black_week_blue_enabled</b>     | <code>bool</code> : läuft Kategorie-Hierarchie hoch, Feld <code>custom_black_week_is_black_week_blue</code> .  |
| <b>is_black_week_enabled</b>          | <code>bool</code> : wie Darkmode, aber Black-Week-Feld auf Produkt/Kategorie/Landing/SalesChannel/Hersteller + Hierarchie.   |
| <b>get_box_settings</b>               | <code>array</code> : Keys <code>showDescription</code> , <code>showAdvantages</code> , <code>horizontalLayout</code> , <code>showDeliveryTime</code> , <code>showDeliveryInAdvantages</code> , <code>hideAdvantagesIcons</code> aus Kategorie-Custom-Fields (Defaults siehe Code).   |
| <b>get_category_from_id</b>           | <code>CategoryEntity</code>  |
| <b>adjust_workdays</b>                | <code>DateTimeInterface</code> : Delegation an <code>HolidayService::adjustToWorkday</code> ; Sales-Channel-ID ggf. aus Request-Attribut <code>sw-sales-channel-id</code> .  |
| <b>get_recommended_properties</b>     | <code>array</code> : Zeilen aus <code>itksw6_recommended_product_properties</code> , UUIDs als Hex.  |
| <b>get_itk_easter_eggs</b>            | <code>array</code> : Osterei-Logik mit DB <code>_itk_easter_eggs</code> / <code>_itk_easter_eggs_log</code> , Zufall/ <code>random_factor</code> , Reservierung über <code>live/live_at</code> .   |
| <b>get_itk_easter_eggs_user_limit</b> | <b>Deklariert <code>int</code></b> , bei <code>\$customer === null</code> liefert der Code fälschlich <code>[]</code> (Typinkonsistenz im Quellcode). Sonst 0 oder 1 je nach Limits (Tag / 30 Min).  |
| <b>dd</b>                             | <code>dump + die</code> (nur für Debugging).   |
| <b>itk cms ajax modal</b>             | HTML+Inline-JS: Bootstrap-Modal, lädt per <code>fetch</code> die URL <code>frontend.cms.page</code> (+ optional <code>navigationId</code> ). <b>Options:</b> <code>id</code> , <code>closable</code> , <code>title</code> , <code>autoOpen</code> , <code>timer</code> (Schließen verzögern), <code>onlyOncePerSession</code> ( <code>sessionStorage</code> ), <code>openInTimer</code> (Öffnen verzögern). Logger bei Bedarf. |

## 10 Weitere Subscriber und Services

### 10.1 ProductCustomFieldSubscriber

- **Event:** `product.written` → `onProductWritten`

- Loggt bei gesetztem Produkt-Custom-Field `onco_sfc_load_configuration_id` (INFO), sonst DEBUG, inkl. Admin-User aus Context → `itk_product_cf_edit_logger.log`.

## 10.2 ProductCustomFieldExcludeSubscriber

- **Custom Field (DAL-Pfad):** `customFields.custom_product_infos_important_hide_product` (Multi-Select, als JSON-Liste gespeichert).
- **Events** (alle mit Priorität **-100**, nach Doofinder o. Ä.): `ProductSearchCriteriaEvent`, `ProductSuggestCriteriaEvent`, `ProductCrossSellingCriteriaEvent`, `ProductCompareCriteriaEvent`, `ProductQuickViewCriteriaEvent`, `ProductListingCriteriaEvent`, `WishListPageProductCriteriaEvent`, `GuestWishListPageletProductCriteriaEvent`.
- Mappt Event-Klassen auf Werte (`search`, `listing`, ...); bei Listing wird bei Such-Route/Parametern `search` statt `listing` verwendet.
- Filter: Produkt sichtbar, wenn Feld **null** oder **not contains** Event-Wert im JSON-String.

## 10.3 WishListProductIdsPageSubscriber

- **Events:** `GenericPageLoadedEvent`, `NavigationPageLoadedEvent`, `SearchPageLoadedEvent`, `ProductPageLoadedEvent`
- **Extension:** `itkWishListProductIds` auf der Page als `ArrayStruct` mit `ids` (Hex-Product-IDs), nur für eingeloggte Kunden; liest `customer_wishlist` / `customer_wishlist_product` per DBAL.

## 10.4 AccountTypeChangeSubscriber

- **Events:** `CustomerEvents::CUSTOMER_WRITTEN_EVENT`, `checkout.customer.profile.update`
- Hält **accountType** bei Schreibvorgängen/Profil-Update an den **persistierten** Wert (Änderungsversuche werden überschrieben bzw. Request angepasst).

## 10.5 HolidayService

- Feiertage pro Jahr (API + optional Sales-Channel-Custom-Field `custom_additional_holidays`), Cache `cache.app` mit TTL bis Jahresende, Methoden u. a. `getHolidaysForYear`, `isHoliday`, `isWeekend`, `isWorkday`, `adjustToWorkday`, `clearAllCache`, `cleanupOldHolidays`, `debugCache`.

## 10.6 CategoryCopyService / itk:copy:category (Experimentell - nicht deployed)

- Kopiert eine Kategorie inkl. aller Kinder; neue Wurzel: „<Name> - Kopie“, Kinder behalten Namen.

## 11 CLI-Befehle

| Befehl                           | Klasse                    | Beschreibung   |
|----------------------------------|---------------------------|--|
| <b>itk:copy:category</b>         | CopyCategoryCommand       | Argument <b>category_id</b> (UUID).  |
| <b>itk:holiday:test</b>          | HolidayServiceTestCommand | Optionen: <code>--year</code> , <code>-sales-channel</code> , <code>-clear-cache</code> , <code>--test-api</code> , <code>-test-cache</code> , <code>-test-custom-fields</code> , <code>--cleanup</code> , <code>-interactive</code> , <code>--date</code> . |
| <b>itk:custom-fields:install</b> | CustomFieldInstallCommand | Installiert/aktualisiert Produktbox-Felder, Kunden-Felder (Trusted Shops), über Custom FieldInstaller zusätzliche Feiertage am Sales Channel.  |

## 12 Storefront-JavaScript (Einstieg)

- **src/Resources/app/storefront/src/main.js**: importiert `./js/badges`, registriert Plugin **CompareModal** für Selektor `[data-compare-modal]`.
- **src/Resources/app/storefront/src/js/main.js**: `./labels`, `./checkout`.
- Weitere Plugin-Quelldateien existieren unter `plugin/` (z. B. `gallery-slider.plugin.js`, `horizontal-product-navigation.plugin.js`); ohne Registrierung in `main.js` sind sie im gebauten Theme ggf. inaktiv, sofern nicht anderweitig eingebunden.

## 13 Abhängigkeiten zur Laufzeit (Überblick)

- **Storefront / Core:** Standard.
- **NetiNextEasyCoupon:** `EasterEggController`, `EasyCouponImportExportSubscriber`, Migration Profil `neti_easy_coupon`.
- **Optionale DB-Strukturen:** `_itk_easter_eggs`, `_itk_easter_eggs_log`, `_itk_sw6_recommended_product_properties` (keine Shopware-Migrationen im Theme dafür im überprüften Umfang – Betrieb/DB-Verantwortung).