Custom Catalog Documentation

Complete PDF Manual

Guide for Magento administrators, technical project owners, and implementation teams.

LAST UPDATED
2026-10-13

SOURCE
Open complete documentation online

EXTENSION
Magento 2 Extension Custom Catalog

Table of Contents

Custom Catalog Documentation · custom-catalog

Overview
Custom Catalog
Getting Started
Installation
Configuration
Features
Access modes
Catalogs and assignments
Customer "My Products" Page
Categories and search
PHP service contracts
Add-Ons
REST API
Hyvä Compatibility
Reference
Configuration Fields
Database Reference
Troubleshooting
Common Issues
Caching and Indexing
FAQ

Custom Catalog

Custom Catalog controls which products a signed-in customer can see and buy. Products can come from named catalogs assigned to customers or customer groups, plus direct product grants for individual customers.

Choose an access mode

- **Override hidden if assigned** lets an assignment make an otherwise hidden product available. Products that are normally visible remain available.
- **Strict whitelist (assigned only)** limits the customer to products allowed through active assigned catalogs or direct grants.

The first mode adds exceptions to the normal catalog. Strict mode creates a private assortment.

How assignments combine

A customer's allowed set can include products from directly assigned catalogs, catalogs assigned to the customer's group, and direct customer-product grants. Catalog audience assignments are store-view scoped, and inactive catalogs do not contribute products.

The customer account includes a **My Products** page. It can show all allowed products, catalog products only, or direct grants only.

Start here

1. Install the extension.
2. Choose the access and My Products modes.
3. Create a catalog, add products and assign an audience.
4. Test category pages, search, direct product URLs, cart behavior and My Products.

For subaccount-specific assortments, see Sublogin Custom Catalog.

The optional REST API manages named catalogs and their assignments, and the Hyvä compatibility package provides the storefront integration for Hyvä themes — see the add-on overview.

See Common Issues, Caching and Indexing and the FAQ when an assortment does not behave as expected.

View the product page

Installation

Install the package from the repository configured for your SoftwareSilo order.

Replace `repo.softwaresilo.io`, `softwaresilo`, `https://repo.softwaresilo.io/` and `<token>` with the repository details shown under **Account > Licenses** in your SoftwareSilo account — see Installation & Updates.

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
composer require mageb2b/custom-catalog:*

php bin/magento module:enable MageB2B_ExtensionManager
MageB2B_CustomCatalog
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
php bin/magento cache:flush
```

Check the module state:

```
php bin/magento module:status MageB2B_CustomCatalog
```

In the Admin, open **Catalog > Custom Catalog** and **Stores > Configuration > MageB2B > Custom Catalog**. Continue with configuration before assigning production customers.

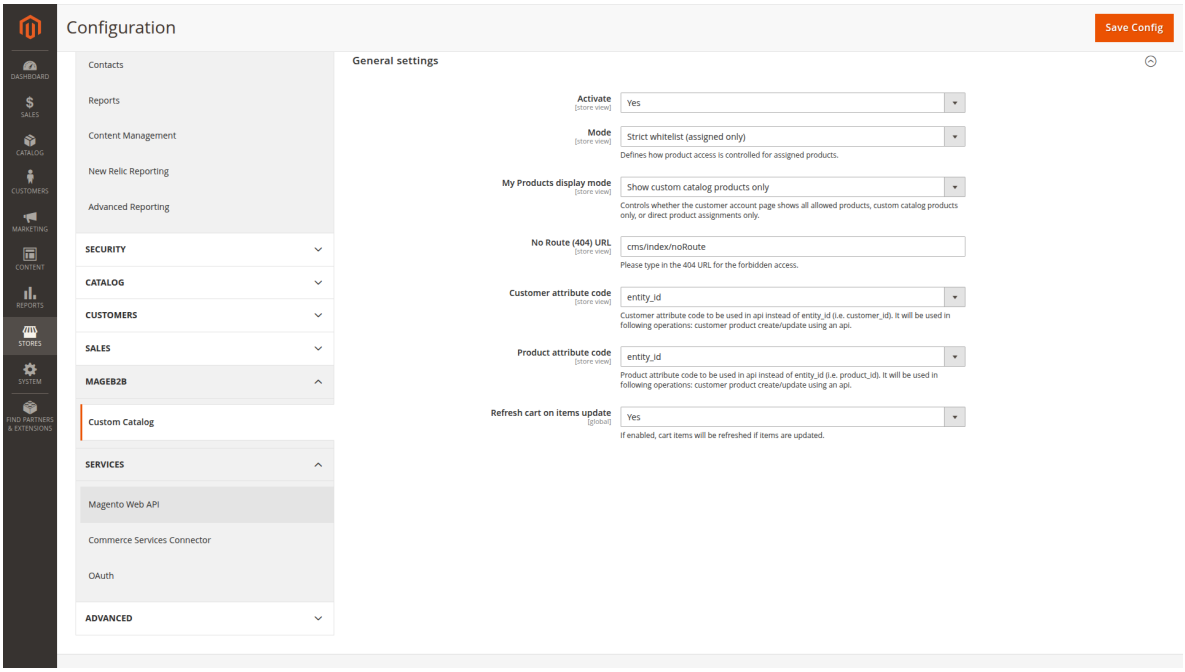
Uninstall

`bin/magento module:uninstall` drops all seven `customcatalog_*` tables and deletes every `core_config_data` row whose path starts with `customcatalog/`. That pattern also covers settings written by other packages — for example the Sublogin Custom Catalog add-on stores its options under `customcatalog/general/` — so back up configuration before uninstalling while related packages remain installed.

Configuration

Open **Stores > Configuration > MageB2B > Custom Catalog** at the website or store-view scope you want to change.

Setting	Purpose
Activate	Enables catalog access control
Mode	Adds assigned hidden products or enforces an assignment-only whitelist
My Products display mode	Shows all allowed, catalog-only or direct-assignment-only products
No Route URL	Destination used when product access is denied
Customer attribute code	External customer identifier used by API-backed writes
Product attribute code	External product identifier used by API-backed writes
Refresh cart on items update	Rechecks affected cart items after assignment changes



The default access mode is **Override hidden if assigned**. Change to strict whitelist only after each affected customer or group has a complete assortment.

Use the standard no-route path unless the store has a deliberate custom destination. A denied product should not reveal product details through a special error page.

The cart refresh setting is global. Keep it enabled when assignment changes must remove

products that a customer can no longer buy.

After saving, clear Magento cache and test with a customer account that has a known assignment.

Access modes

Override hidden if assigned

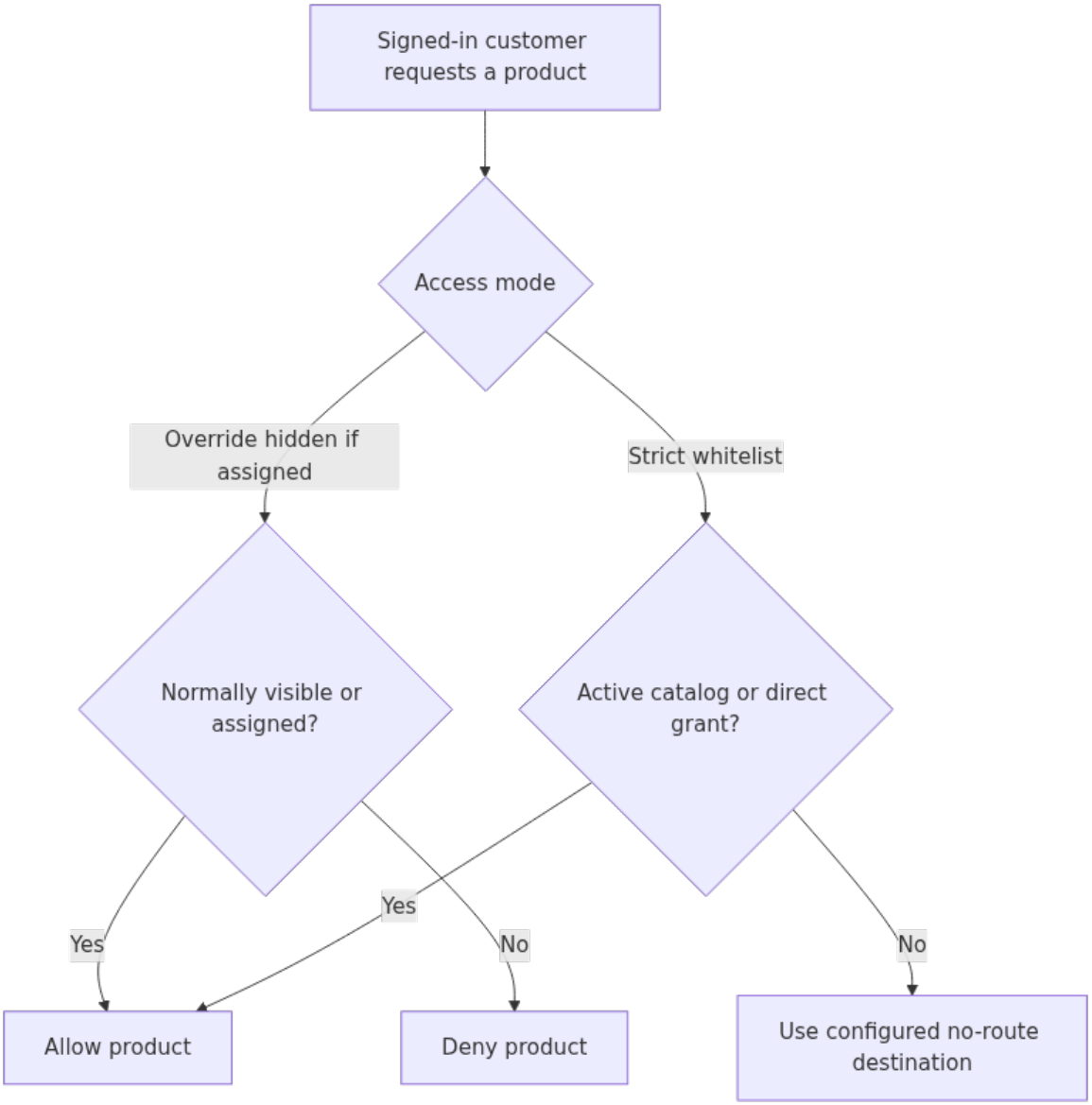
This is the default mode. Magento's normally visible products remain available. A custom catalog or direct grant can also make an assigned product available to that customer.

Use it when customer assortments should add exceptions without turning the rest of the catalog into a whitelist.

Strict whitelist

Strict mode limits a signed-in customer to products resolved from active assigned catalogs and direct grants. Products outside that set are removed from listings and search, and direct product access uses the configured no-route destination.

Before enabling strict mode, verify at least one complete customer journey: category, search, product page, cart, checkout and My Products. Test a customer with direct catalog assignment and one who receives the catalog through a customer group.



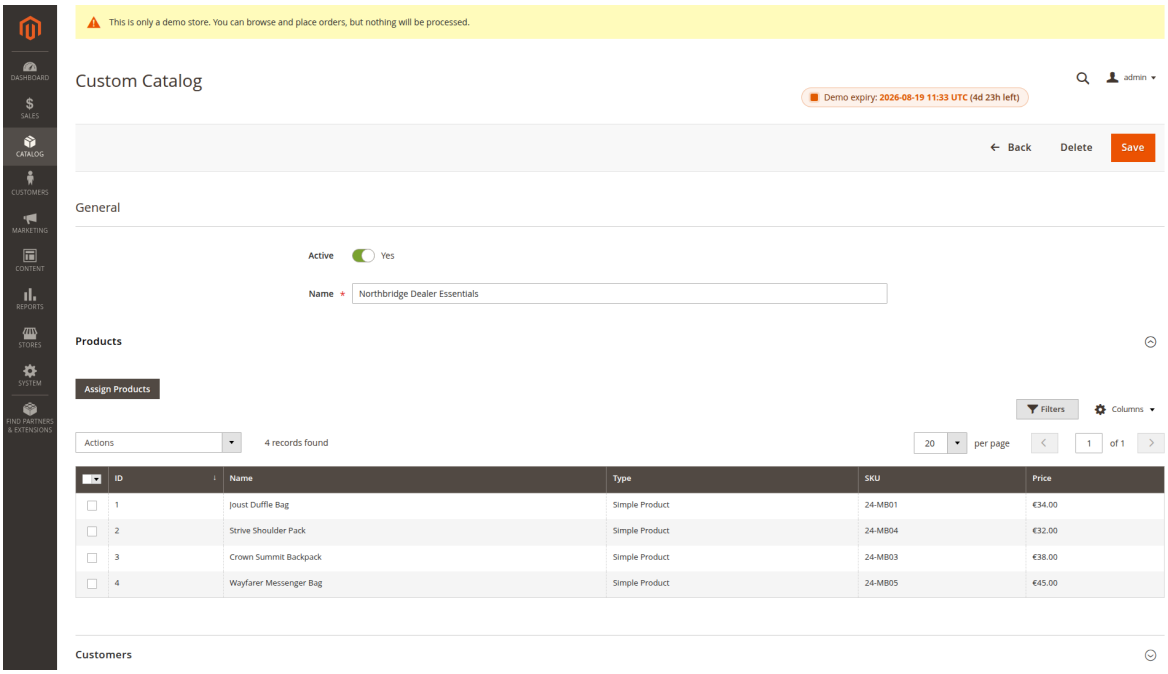
Guests do not have a customer-specific assignment set, and strict mode does not create one: the whitelist resolves to *no restriction* for visitors without a signed-in account, and the cart access check only applies to quotes with a customer ID. Strict whitelist therefore restricts signed-in customers only — guests keep seeing the normal Magento catalog. If the storefront must hide assortments from guests as well, handle that separately (for example with Magento's catalog or store restrictions), and test guest behavior separately instead of assuming it matches a signed-in strict customer.

Catalogs and assignments

Open **Catalog > Custom Catalog** to create a named catalog. Give it a clear business name and keep it active only while its assortment should be available.

Add products

Edit the catalog and assign the products that belong to the assortment. Product membership belongs to the catalog itself and is shared by every audience assigned to that catalog.



Assign an audience

A catalog can be assigned to:

- individual customers;
- Magento customer groups.

Assignments work in both directions. You can pick the audience inside the catalog editor, or open the customer or customer group and pick the catalogs there:

- **Customers > All Customers**, open a customer — the **Custom Catalog** section lists the assigned catalogs and offers an **Assign Custom Catalogs** dialog.
- **Customers > Customer Groups**, open a group — the **Custom Catalog** section does the same for that group. It also appears when creating a new group.

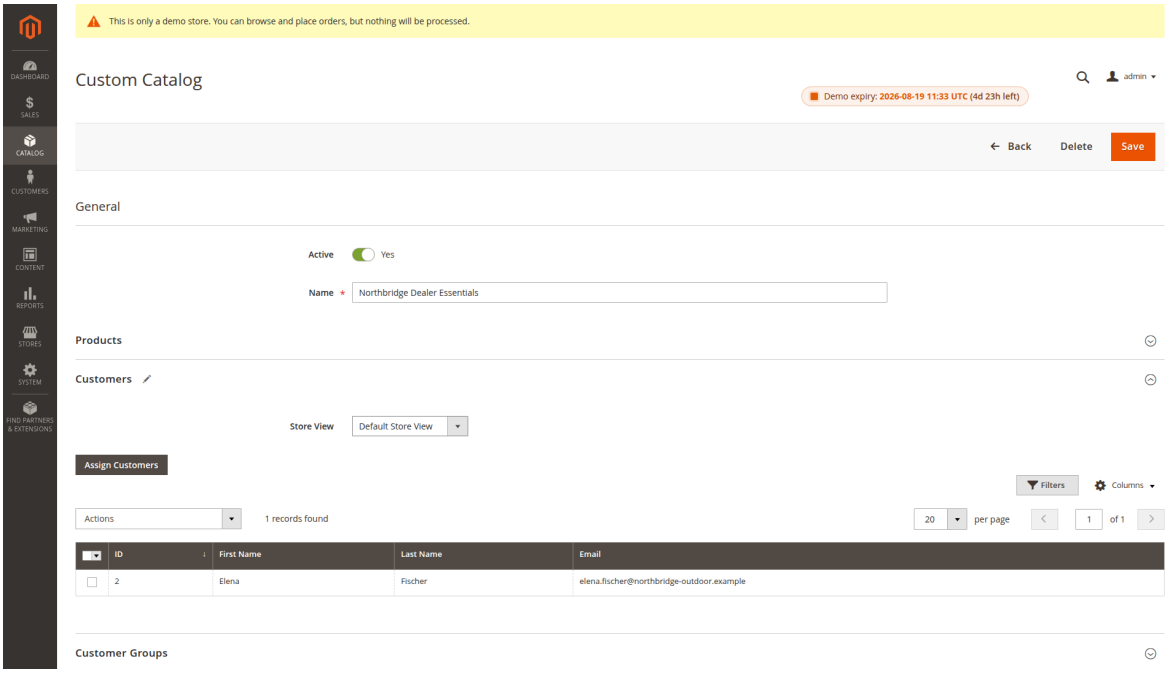
The **NOT LOGGED IN** group (ID 0) is excluded from group assignments — guests cannot receive a catalog audience.

Note the different save behaviour: in the catalog editor, assignment changes are only

staged until you save the catalog. In the customer and group editors, the assign/unassign action is applied immediately.

Store view scope

The store switcher offers **All Store Views** in addition to individual store views. An assignment in All Store Views applies globally as a fallback; an assignment in a specific store view overrides it for that store. Assign a catalog globally first, and only add a store-view assignment where the audience should differ — you do not need to repeat the same assignment in every store view.



Direct product grants

Direct grants connect one customer to one product at a store view. They are useful for small exceptions that do not justify another named catalog.

Manage them on the customer: **Customers > All Customers**, open a customer — the **Custom Product** section lists the directly granted products and offers assign and unassign actions. "Custom Product" is the Admin label for a direct product grant.

In strict mode, the customer's allowed products are the union of active assigned catalog products and direct grants. Deactivating a catalog removes its contribution without deleting the catalog or its product membership.

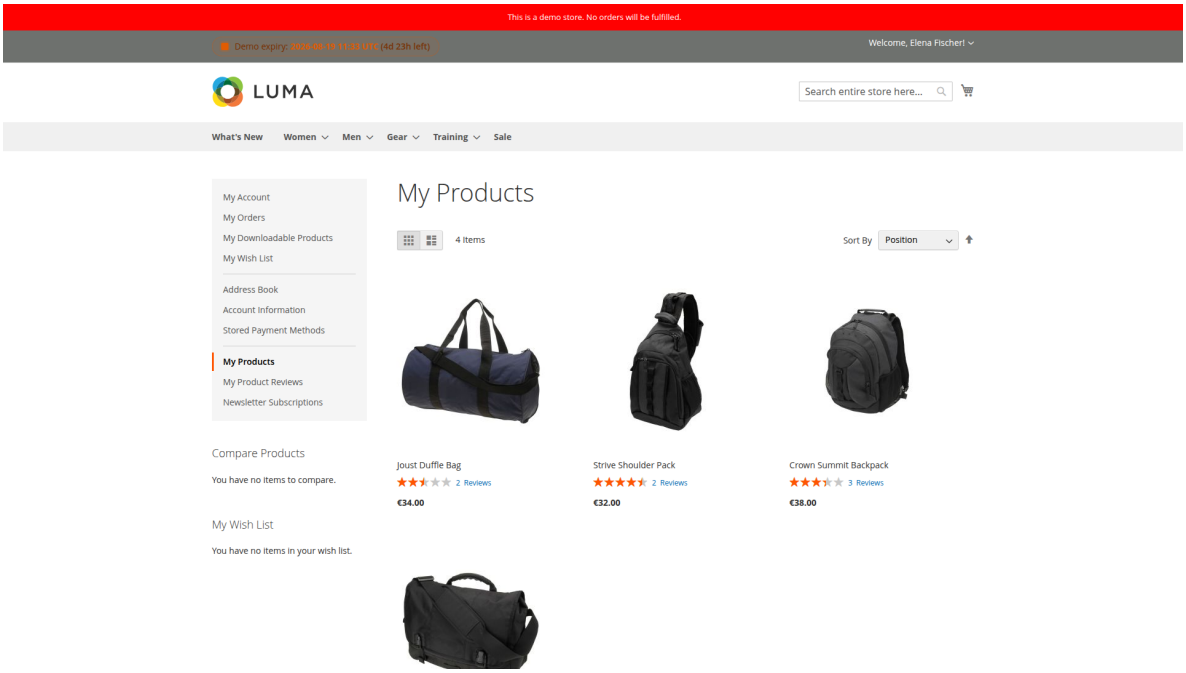
Copy a catalog

The catalog grid offers **Copy** as a row action, plus a mass action for selected rows. Copying a catalog duplicates it including its product and audience assignments, which is the fastest way to create a variant for a new customer segment.

Customer "My Products" Page

Signed-in customers can open **My Products** from the account navigation. The page provides a focused product list for their assigned assortment.

On standard Magento themes the link appears in two places: the storefront top links and the customer account navigation. Both render only while Custom Catalog is active and the customer has at least one product in the configured display mode — a customer whose current display mode resolves to an empty list does not see the links. The page itself is excluded from the full-page cache so the list always reflects the current assignment state.



Choose one display mode in configuration:

Mode	Content
Show all allowed products	Every product the customer can access in the current mode
Show custom catalog products only	Products from active assigned named catalogs
Show direct assigned products only	Products granted directly to the customer

The page mode changes the account view, not the underlying storefront access rule. A product may be accessible but omitted from My Products when the page is configured to show only one assignment source.

If My Products is empty, first check its display mode, then the catalog active state, store-view assignment and direct grants.

Categories and search

In strict mode, Custom Catalog filters category collections and storefront search against the current customer's allowed products. The direct product URL is the final access check: a customer who is not allowed to view the product is sent to the configured no-route page.

Search engines maintain their own indexes, so assignment changes and catalog data must be tested after the relevant Magento indexers have completed. Customer access is applied to the storefront request even though the underlying search index is shared.

After changing a large assortment:

1. run the relevant Magento indexers;
2. clear application and full-page cache;
3. sign in with a fresh customer session;
4. test category, search and direct URL access;
5. check the cart if products were removed from the assortment.

If search still lists a restricted product but its product page is denied, do not treat the search result as proof of access. Rebuild **Catalog Search**, clear the caches and repeat the test with a new customer session. Check the direct URL again after the search result is clean.

When the Pricesystem extension is installed, the same strict whitelist also filters the requested product IDs on Pricesystem's storefront price endpoints, so disallowed products return no price data through that path either.

See caching and indexing for a focused checklist.

PHP service contracts

The base Custom Catalog module contains service contracts for named catalogs, product membership, customer assignments, customer-group assignments and direct customer-product grants. Magento modules can use these contracts without calling controllers or writing assignment tables directly.

REST exposure belongs to the optional Custom Catalog API package. The REST package manages named catalogs and their assignments. It does not expose the base module's direct customer-product repository.

Use repositories and assignment services so cache context, customer access versions and cart refresh behavior remain consistent after writes.

Where access is enforced

The strict-mode whitelist is not only a storefront filter. It is enforced wherever a product can be added to a cart:

- the storefront catalog and category listings (see Search and category filtering);
- the add-to-cart path in the storefront, REST, GraphQL and Admin order/cart creation — a plugin on the quote's add-product check rejects products outside the customer's allowed set;
- related, up-sell and cross-sell product collections and catalog search (including advanced search and ID filters).

An integration that must apply its own decision can listen to the `customcatalog_quote_product_access_validate` event, which the add-product plugin dispatches before rejecting a product.

Add-Ons

The base Custom Catalog package covers named catalogs, audience assignments and storefront access control. Install an add-on only when the project needs an additional integration or storefront implementation.

Add-on	Install it when you need	Composer package
REST API	Magento REST endpoints for named catalogs and product, customer and customer-group assignments	`mageb2b/custom-catalog-api`
Hyvä Compatibility	Custom Catalog storefront components on a Hyvä-based store	`mageb2b/custom-catalog-hyva`

Each add-on requires the base `mageb2b/custom-catalog` package. See the individual guide for exact commands, permissions and verification steps.

REST API

The optional package `mageb2b/custom-catalog-api` exposes named catalog management and assignment services through Magento Web API.

```
composer require mageb2b/custom-catalog-api:*
php bin/magento setup:upgrade
php bin/magento cache:flush
```

Catalog records

Method	Endpoint	ACL resource
`GET`	`/V1/customcatalog/catalog/:id`	`MageB2B_CustomCatalog::manage`
`GET`	`/V1/customcatalog/catalog/search`	`MageB2B_CustomCatalog::manage`
`POST`	`/V1/customcatalog/catalog`	`MageB2B_CustomCatalog::save`
`PUT`	`/V1/customcatalog/catalog/:id`	`MageB2B_CustomCatalog::save`
`DELETE`	`/V1/customcatalog/catalog/:id`	`MageB2B_CustomCatalog::delete`

`DELETE` returns `true` when the catalog was removed.

Assignments

Catalog-centric routes assign and unassign products, customers and customer groups below `/V1/customcatalog/catalog/:catalogId/...`:

Method	Endpoint	Body parameters	ACL resource
`POST`	`/V1/customcatalog/catalog/:catalogId/products/assign`	`productIds`	`MageB2B_CustomCatalog::save`
`POST`	`/V1/customcatalog/catalog/:catalogId/products/unassign`	`productIds`	`MageB2B_CustomCatalog::delete`
`POST`	`/V1/customcatalog/catalog/:catalogId/customers/assign`	`customerIds`, `storeId`	`MageB2B_CustomCatalog::save`
`POST`	`/V1/customcatalog/catalog/:catalogId/customers/unassign`	`customerIds`, `storeId`	`MageB2B_CustomCatalog::delete`
`POST`	`/V1/customcatalog/catalog/:catalogId/customer-groups/assign`	`customerGroupIds`, `storeId`	`MageB2B_CustomCatalog::save`
`POST`	`/V1/customcatalog/catalog/:catalogId/customer-groups/unassign`	`customerGroupIds`, `storeId`	`MageB2B_CustomCatalog::delete`

Customer- and group-centric routes manage catalogs in the other direction:

Method	Endpoint	Body parameters	ACL resource
`POST`	`/V1/customcatalog/customer/:customerId/catalogs/assign`	`catalogIds`, `storeId`	`MageB2B_CustomCatalog::save`
`POST`	`/V1/customcatalog/customer/:customerId/catalogs/unassign`	`catalogIds`, `storeId`	`MageB2B_CustomCatalog::delete`
`GET`	`/V1/customcatalog/customer/:customerId/catalogs/:storeId`	—	`MageB2B_CustomCatalog::manage`
`POST`	`/V1/customcatalog/customer-group/:customerGroupId/catalogs/assign`	`catalogIds`, `storeId`	`MageB2B_CustomCatalog::save`

Method	Endpoint	Body parameters	ACL resource
`POST`	`/V1/customcatalog/customer-group/:customerGroupId/catalogs/unassign`	`catalogIds`, `storeId`	`MageB2B_CustomCatalog::delete`
`GET`	`/V1/customcatalog/customer-group/:customerGroupId/catalogs/:storeId`	—	`MageB2B_CustomCatalog::manage`

Customer and customer-group assignments are store-view scoped, so `storeId` is a required parameter on every customer and group route — in the request body for writes and in the URL for reads. Product membership belongs to the catalog itself and has no store scope.

Every `unassign` route returns the number of deleted assignment rows, and the `GET .../catalogs/:storeId` reads return an `int[]` of assigned catalog IDs. Assign routes return no payload. An HTTP success does not prove that every submitted ID was applied.

Create a Magento integration with only the Custom Catalog resources it needs. After a synchronization, verify one affected customer in the storefront and process partial failures rather than treating an HTTP success as proof that every assignment changed.

ACL resources

The module defines `MageB2B_CustomCatalog::all` with `::manage`, `::save` and `::delete` children, plus `MageB2B_CustomCatalog::config` for the configuration section. The same resource IDs protect the Admin: reads use `::manage`, mutating actions (save, assign, unassign, copy) are bound to `::save`, and deletes use `::delete`. Grant integrations and admin roles the smallest set that covers their operations.

Hyvä Compatibility

The optional `mageb2b/custom-catalog-hyva` package provides the Hyvä storefront integration for Custom Catalog. The base package still owns access rules, assignments and the My Products route.

- Magento module: `MageB2B_CustomCatalogHyva`
- Required base package: `mageb2b/custom-catalog` (1.0.5 or later)
- Required storefront: Hyvä Default Theme (`hyva-themes/magento2-default-theme` 1.1.18 or later) including `Hyva_CompatModuleFallback`

You do not need this package for Luma, Blank or a theme that inherits Magento's standard frontend templates.

Install the add-on

Install and configure the base package first. Then add the Hyvä bridge:

```
composer require mageb2b/custom-catalog-hyva:*
php bin/magento module:enable MageB2B_CustomCatalogHyva
php bin/magento setup:upgrade
php bin/magento cache:flush
```

Follow the normal Hyvä deployment steps for the store, including generated theme assets where applicable. The package registers `MageB2B_CustomCatalogHyva` as the compatibility module for `MageB2B_CustomCatalog` in Hyvä's compat-module registry, so `hyva_`-prefixed layout handles apply on Hyvä themes.

What the add-on changes

- **My Products page** — the `hyva_customcatalog_customer_index` handle renders the assigned-product list through Hyvä's catalog list item templates.
- **Customer menu link** — a **My Products** entry is added to the signed-in customer header links. Like the standard theme links, it renders only while Custom Catalog is active and the customer has at least one product in the configured display mode.

Access control, assignments, the My Products display mode and the no-route destination remain owned by `MageB2B_CustomCatalog` and behave the same on Luma and Hyvä.

Verify the integration

Test with a signed-in customer on the active Hyvä store view: open the **My Products** header link, compare the list with the configured display mode, confirm that a disallowed product URL still reaches the no-route destination, and repeat at a mobile viewport.

Configuration Fields

This reference lists the settings exposed by `mageb2b/custom-catalog` in Magento Admin.

Configuration paths are included for controlled deployments and `config:set`. Use Magento Admin for normal changes so field validation and scope inheritance still apply. The extension status display is not listed because it reports installation state rather than a configurable value.

General settings

Field	Configuration path	Input	Scope
Activate	`customcatalog/general/activate`	Select	Default, Website, Store view
Mode	`customcatalog/general/mode`	Select	Default, Website, Store view
My Products display mode	`customcatalog/general/my_products_display_mode`	Select	Default, Website, Store view
No Route (404) URL	`customcatalog/general/noroute_url`	Text	Default, Website, Store view
Customer attribute code	`customcatalog/general/customer_attribute_code`	Select	Default, Website, Store view
Product attribute code	`customcatalog/general/product_attribute_code`	Select	Default, Website, Store view
Refresh cart on items update	`customcatalog/general/refresh_cart_on_items_update`	Select	Default

The No Route field accepts three forms. An empty value forwards internally to Magento's standard `noroute` page. A value starting with `/` or `http(s)://` redirects to that absolute URL. Any other value is treated as a Magento route path (for example `cms/page/view`).

Database Reference

Custom Catalog separates the catalog definition from its assignments. This makes one curated product set reusable across customers and customer groups.

Catalog membership

Relationship	Why it matters
`customcatalog_catalog_product` links one `customcatalog_catalog` with one `catalog_product_entity`	Defines which products belong to a reusable custom catalog.
`customcatalog_catalog_customer` links one `customcatalog_catalog` with one `customer_entity`	Makes the catalog available to selected customer accounts.
`customcatalog_catalog_customer_group` links one `customcatalog_catalog` with one `customer_group`	Reuses the same catalog for all customers in a Magento customer group.
Catalog assignment records belong to one `store`	Allows visibility to differ between Store Views.

Direct customer assignments

Relationship	Why it matters
`customcatalog_customer_product` links one `customer_entity` with one `catalog_product_entity`	Supports a direct product allowlist without requiring a reusable catalog record.
`customcatalog_customer_product` belongs to one `store`	Keeps direct visibility assignments Store View specific.

Internal version records used to refresh catalog state are not part of this public reference. Use the module's Admin screens, import process or APIs for changes.

Common Issues

A customer sees products outside the intended assortment

Confirm the configured access mode first. **Override hidden if assigned** adds assigned products to Magento's normally visible catalog; it does not turn every other product into a private product. Use **Strict whitelist (assigned only)** when the customer must see only assigned catalog products and direct grants.

Then check every assignment path. A product can be allowed by a catalog assigned directly to the customer, a catalog assigned to the customer group, or a direct customer-product grant.

An assigned product is still unavailable

Check that:

1. Custom Catalog is active in the tested store view.
2. The named catalog is active.
3. The catalog contains the product.
4. The catalog audience includes the current customer or customer group in the current store view.
5. The product itself is enabled and assigned to the website.

If the product is a direct grant, confirm that the grant belongs to the same customer and store context. Sign out and back in before retesting a changed customer assignment.

A direct product URL returns a 404 page

In strict mode, this is the expected response for a product the customer may not access. If the product should be allowed, test the same product through **My Products** and check the assignment paths above. Also confirm that the configured no-route target is valid.

Search or category pages show an old assortment

Assignments can change before Magento's catalog-search index and storefront caches reflect the new result. Follow Caching and indexing, then test in a fresh signed-in session.

Do not treat a successful direct-product check as proof that search is current. Direct access and product collections are separate read paths.

My Products is empty

Check the configured **My Products display mode**. A mode that shows only catalog products will not list direct grants, and a direct-grants-only mode will not list products inherited from a named catalog. Also confirm that the customer is signed in to the

expected store view.

The My Products link is missing

The top link and account-navigation link render only while the module is active in the current store view and the customer has at least one product under the configured display mode. A customer whose allowed products exist only in another mode — for example only direct grants while the page shows catalog products — sees no link. Check the display mode and the assignment paths above before treating it as a template or cache issue.

A removed product remains in the cart

Enable cart refresh when access changes should remove products that are no longer allowed. Without that setting, an older cart can retain an item even though new catalog access is blocked. Recalculate the cart and verify that any custom cart or checkout integration respects Magento item-removal messages.

An assignment works in one store view but not another

Catalog audiences are store-view scoped. Compare the exact store view used for the assignment with the store view in the customer session. A global-looking customer or group name does not make its catalog assignment global.

A bulk or API assignment reports partial success

Bulk operations can accept valid rows and return invalid rows separately. Review the saved and unsaved identifiers instead of assuming the whole request rolled back. Typical causes are invalid entity IDs, duplicate relations and assignments made for the wrong store view.

What to collect for support

Provide the customer ID, customer group, store view, access mode, catalog ID, product SKU and every assignment path that may allow the product. State whether the problem occurs in My Products, search, a category, a direct product URL or the cart. Those details make it possible to distinguish scope, indexing and access-resolution problems.

Caching and Indexing

A customer sees the wrong products

Confirm the active access mode, store view and customer account. Then review active catalogs assigned directly and through the customer group, plus direct product grants.

In strict mode, those sources form the allowed set. In override mode, normally visible Magento products can remain available even without an assignment.

An assigned catalog has no effect

Check that the catalog is active, contains the product and is assigned in the current store view. A catalog assignment and product membership are separate steps.

Search or a category shows stale products

Run the relevant Magento indexers and clear application and full-page cache. Retest with a fresh customer session. Also check search suggestions and the direct product URL.

My Products is empty

Review **My Products display mode**. Catalog-only mode omits direct grants, while direct-only mode omits catalog products. Also verify catalog active state and store-view assignment.

A removed product remains in the cart

Keep **Refresh cart on items update** enabled when assignment writes should recheck cart contents. After a large API synchronization, verify the cart of an affected test customer.

Strict mode appears different for guests

Guests have no customer-specific catalog or direct grants. Test guest behavior separately from signed-in customers and do not use a guest result to validate a customer's whitelist.

A direct product URL still opens

Clear full-page cache and retest in a new session. Confirm the customer does not receive the product through another active catalog, customer-group assignment or direct grant.

FAQ

What is the difference between the two access modes?

Override mode adds assigned hidden products while leaving normally visible products available. Strict whitelist limits signed-in customers to assigned catalog products and direct grants.

Can one catalog be assigned to several customers?

Yes. Assign it to individual customers or to a Magento customer group.

Are assignments store-view specific?

Customer and customer-group catalog assignments are store-view scoped. Direct customer-product grants are also scoped by store view.

What happens when a catalog is inactive?

Its products no longer contribute to the customer's allowed set. The catalog, product membership and audience assignments remain stored.

Can one product be granted without creating a catalog?

Yes. Use a direct customer-product grant for a small exception.

Why is an accessible product missing from My Products?

The page may be configured to show only catalog products or only direct grants. My Products display mode is separate from storefront access.

Does Custom Catalog filter search and categories?

Yes. It filters category and search results and blocks forbidden direct product access. Reindex and clear cache after broad assortment changes.

Is there a REST API?

Yes, through `mageb2b/custom-catalog-api`. It manages named catalogs and product, customer and customer-group catalog assignments.

Does the base product include Sublogin assortments?

No. Sublogin-specific access is provided by the separate Sublogin Custom Catalog package and is documented under the Sublogin KB.