



REST API

Customer Documents Documentation

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

LAST UPDATED
2026-09-26

SOURCE
[Open this article online](#)

EXTENSION
[Magento 2 Extension Customer Documents](#)

REST API

The optional `mageb2b/customerdocuments-api` package exposes administrator-level service contracts for document management and assignments. It is not a customer self-service API.

- Composer package: `mageb2b/customerdocuments-api`
- Magento module: `MageB2B_CustomerDocumentsApi`

Installation

Install the API package in addition to the base Customer Documents extension:

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
composer require mageb2b/customerdocuments-api:*
php bin/magento module:enable MageB2B_CustomerDocumentsApi
php bin/magento setup:upgrade
php bin/magento cache:flush
```

The add-on reuses the base module ACL resources:

- read: `MageB2B_CustomerDocuments::manage`
- write: `MageB2B_CustomerDocuments::save`
- delete: `MageB2B_CustomerDocuments::delete`

Assign only the permissions needed by the integration. See [Admin permissions](#) and Magento's [/swagger](#) page for the schema generated by the installed version.

Authentication

Use a Magento integration or administrator bearer token with the smallest Customer Documents ACL scope needed for the operation. A dedicated integration is preferable to sharing an administrator login.

```
curl -X POST
"https://your-store.com/rest/V1/integration/admin/token" \
-H "Content-Type: application/json" \
-d '{"username":"admin","password":"***"}'
```

Then call endpoints with:

```
-H "Authorization: Bearer <ADMIN_TOKEN>"
```

Endpoint Overview

Documents

Method	URL
GET	/V1/customerdocuments/:id
GET	/V1/customerdocuments/search
POST	/V1/customerdocuments
PUT	/V1/customerdocuments/:id
DELETE	/V1/customerdocuments/:id

Document files

Method	URL
GET	/V1/customerdocuments/:documentId/file
PUT	/V1/customerdocuments/:documentId/file

These operations read or write base64-encoded local document data. External documents do not expose a local file payload.

Customer Assignments

Method	URL
GET	/V1/customerdocuments/customer/:id
GET	/V1/customerdocuments/customer/search
POST	/V1/customerdocuments/customer
PUT	/V1/customerdocuments/customer/:id
DELETE	/V1/customerdocuments/customer/:id

Customer Group Assignments

Method	URL
GET	/V1/customerdocuments/customergroup/:id
GET	/V1/customerdocuments/customergroup/search

Method	URL
POST	/V1/customerdocuments/customergroup
PUT	/V1/customerdocuments/customergroup/:id
DELETE	/V1/customerdocuments/customergroup/:id

Product Assignments

Method	URL
GET	/V1/customerdocuments/product/:id
GET	/V1/customerdocuments/product/search
POST	/V1/customerdocuments/product
PUT	/V1/customerdocuments/product/:id
DELETE	/V1/customerdocuments/product/:id
DELETE	/V1/customerdocuments/product/by-product/:productId
DELETE	/V1/customerdocuments/product/by-document/:documentId

Categories

Method	URL
GET	/V1/customerdocuments/category/:id
GET	/V1/customerdocuments/category/search
POST	/V1/customerdocuments/category
PUT	/V1/customerdocuments/category/:id
DELETE	/V1/customerdocuments/category/:id

Links

Method	URL
GET	/V1/customerdocuments/links
POST	/V1/customerdocuments/links
DELETE	/V1/customerdocuments/links/:id

Request payloads

General notes

- **GET** and **DELETE** routes do not require a request body.
- **.../search** routes use standard Magento **searchCriteria** query parameters.

Example:

```
/V1/customerdocuments/product/search?searchCriteria[filter_groups][0][filters][0][field]=document_id&searchCriteria[filter_groups][0][filters][0][value]=1777&searchCriteria[filter_groups][0][filters][0][condition_type]=eq&searchCriteria[currentPage]=1&searchCriteria[pageSize]=20
```

Save a document

```
{
  "document": {
    "id": 1777,
    "name": "Invoice 1001",
    "description": "January invoice",
    "is_external": 1,
    "external_url": "https://example.com/invoice-1001.pdf",
    "is_public": 0,
    "category_id": 829,
    "website_id": 0,
    "show_in_sidebar": 1,
    "valid_from": "2026-01-01",
    "valid_to": "2026-12-31",
    "customer_ids": [4801],
    "customer_group_ids": [1, 3]
  }
}
```

For a local document, save its metadata with **is_external** set to **0**, then use the file endpoint with the **documentFile** wrapper expected by Magento's service contract. The payload contains a base filename and base64 data. Apply the same allowed-extension, file-size and storage policies used by the Admin upload flow.

POST/PUT /V1/customerdocuments/customer (wrapper: **documentCustomer**)

```
{
  "documentCustomer": {
```

```
"id": 4099,  
"document_id": 1777,  
"customer_id": 4801  
}  
}
```

POST/PUT /V1/customerdocuments/customerGroup (wrapper: customerGroup)

This service contract uses `customerGroup`, while the other assignment endpoints use wrappers beginning with `document`. Keep the wrapper exactly as shown; changing it to `documentCustomerGroup` causes Magento to reject the request payload.

```
{  
  "customerGroup": {  
    "id": 3196,  
    "document_id": 1777,  
    "customer_group_id": 3  
  }  
}
```

POST/PUT /V1/customerdocuments/product (wrapper: documentProduct)

```
{  
  "documentProduct": {  
    "id": 385,  
    "document_id": 1777,  
    "product_id": 43432,  
    "store_id": 0,  
    "position": 10  
  }  
}
```

POST/PUT /V1/customerdocuments/category (wrapper: documentCategory)

```
{  
  "documentCategory": {  
    "id": 829,  
    "name": "Security Policies"  
  }  
}
```

POST `/V1/customerdocuments/links` (wrapper: `documentLink`)

```
{
  "documentLink": {
    "id": 1,
    "customer_document_id": 1777,
    "link_entity_type": "order",
    "link_entity_id": 4801
  }
}
```

`link_entity_type` accepts `order`, `invoice`, `shipment` or `creditmemo`;
`link_entity_id` is the internal ID of that sales entity.

Search and error handling

Search endpoints use Magento `searchCriteria`, including pagination and filter groups. Treat assignment IDs as relation-record IDs, not customer, group or product IDs.

Common HTTP outcomes are:

Code	Meaning
200	Request accepted/successful
400	Payload parsing/validation error
401	Unauthorized
403	ACL denied
404	Entity does not exist
500	Service or runtime error

Validate one create, read, update and delete cycle in a non-production environment before automating bulk changes. An API write can change the same customer-visible access relationships as the Magento Admin form.