REST API Add-On

B2B Quote 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 B2B Quote

REST API Add-On

Install the optional **B2B Quote REST API add-on** when another system needs to work with quotes through Magento Web API. The package declares the routes; the base B2B Quote module provides the service implementations and business rules.

- Package: `mageb2b/b2b-quote-api`
- Magento module: `MageB2B_B2BQuoteApi`

Without this add-on, the REST endpoints documented below are not available. GraphQL is provided by the separate GraphQL Add-On.

Installation

Configure the SoftwareSilo Composer repository if it is not already available, then install the package:

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
composer require mageb2b/b2b-quote-api:*
```

See the installation guide for where to find the repository placeholders and access token (Account > Licenses).

Enable the Magento module and apply the setup changes:

```
php bin/magento module:enable MageB2B_B2BQuoteApi
php bin/magento setup:upgrade
php bin/magento cache:flush
```

The add-on has no separate API configuration screen. Create and authorize administrator integrations through Magento's standard integration management.

Authentication

Magento REST authentication depends on the endpoint type:

- **Admin / Integration endpoints**: Integration token + ACL permission
- **Customer endpoints**: customer token (`self` resource)
- **Guest token endpoints**: `anonymous` access but require a **guest quote access token**

Example (admin/integration token):

```
curl -X GET "https://your-store.com/rest/V1/b2bquote/quotes" \
```

```
    -H "Authorization: Bearer {token}"
```

The examples below show route paths. Request and response bodies follow Magento's generated service-contract schema. Check the target store's Swagger/OpenAPI output or service metadata before implementing a client.

Endpoint Groups

Admin (ACL-protected)

Method	URL	Purpose
`GET`	`/V1/b2bquote/quotes`	List quotes (any customer)
`GET`	`/V1/b2bquote/quotes/{quoteId}`	Get quote by id
`POST`	`/V1/b2bquote/quotes`	Create draft quote
`PUT`	`/V1/b2bquote/quotes/{quoteId}`	Save/update quote
`DELETE`	`/V1/b2bquote/quotes/{quoteId}`	Delete quote
`POST`	`/V1/b2bquote/quotes/{quoteId}/status`	Change status (admin flow)
`POST`	`/V1/b2bquote/quotes/{quoteId}/pricing/apply`	Apply pricing rules
`POST`	`/V1/b2bquote/quotes/{quoteId}/workflow/apply`	Apply workflow rules
`POST`	`/V1/b2bquote/quotes/{quoteId}/approve`	Approve a quote
`POST`	`/V1/b2bquote/quotes/{quoteId}/reject`	Reject a quote
`POST`	`/V1/b2bquote/quotes/{quoteId}/recalc`	Recalculate quote totals
`POST`	`/V1/b2bquote/quotes/{quoteId}/split`	Split a quote into new quotes
`POST`	`/V1/b2bquote/quotes/{targetQuoteId}/merge`	Merge source quotes into the target quote

Method	URL	Purpose
GET	/V1/b2bquote/quotes/{quoteId}/history	Read quote revision history

Customer (ownership checked)

Method	URL	Purpose
GET	/V1/b2bquote/me/quotes	List my quotes
GET	/V1/b2bquote/me/quotes/{quoteId}	Get my quote
POST	/V1/b2bquote/quotes/{quoteId}/submit	Submit quote
POST	/V1/b2bquote/quotes/{quoteId}/accept	Accept quote
POST	/V1/b2bquote/quotes/{quoteId}/items	Add item
PUT	/V1/b2bquote/quotes/{quoteId}/items/{itemId}	Update item
DELETE	/V1/b2bquote/quotes/{quoteId}/items/{itemId}	Remove item
POST	/V1/b2bquote/quotes/{quoteId}/messages	Add message
POST	/V1/b2bquote/quotes/{quoteId}/attachments	Add attachment
POST	/V1/b2bquote/quotes/{quoteId}/addresses	Set addresses
POST	/V1/b2bquote/quotes/{quoteId}/shipping-method	Set the shipping method
POST	/V1/b2bquote/quotes/{quoteId}/payment-method	Set the payment method
POST	/V1/b2bquote/quotes/{quoteId}/items/{itemId}/negotiate	Submit an item price negotiation

Guest (token-based)

Method	URL	Purpose
GET	/V1/b2bquote/guest/quotes/{token}	Get guest quote by access token

Method	URL	Purpose
POST	/V1/b2bquote/guest/quotes/{token}/accept	Accept guest quote by access token

Categories, Tags, Status Transitions, Revisions (ACL-protected)

Area	Method	URL
Categories	GET	/V1/b2bquote/categories
Categories	GET	/V1/b2bquote/categories/active
Categories	GET	/V1/b2bquote/categories/{categoryId}
Categories	POST	/V1/b2bquote/categories
Categories	DELETE	/V1/b2bquote/categories/{categoryId}
Tags	GET	/V1/b2bquote/tags
Tags	GET	/V1/b2bquote/tags/popular
Tags	GET	/V1/b2bquote/tags/{tagId}
Tags	GET	/V1/b2bquote/quotes/{quoteId}/tags
Tags	POST	/V1/b2bquote/tags
Tags	DELETE	/V1/b2bquote/tags/{tagId}
Status transitions	GET	/V1/b2bquote/statuses/transitions
Status transitions	GET	/V1/b2bquote/statuses/transitions/{id}
Status transitions	POST	/V1/b2bquote/statuses/transitions
Status transitions	DELETE	/V1/b2bquote/statuses/transitions/{id}
Revisions	GET	/V1/b2bquote/quotes/{quoteId}/revisions
Revisions	GET	/V1/b2bquote/revisions/{revisionId}
Revisions	GET	/V1/b2bquote/revisions/{revisionId1}/compare/{revisionId2}
Revisions	POST	/V1/b2bquote/quotes/{quoteId}/revisions
Revisions	POST	/V1/b2bquote/quotes/{quoteId}/revisions/{revisionId}/rollback

How authorization is applied

- Administrator and integration routes declare a specific `MageB2B_B2BQuote::*` ACL resource in `webapi.xml`. Grant only the resources the integration needs.
- Customer routes use Magento's `self` resource and call ownership-aware service methods. A valid customer token cannot manage another customer's quote.

- Guest routes are declared `anonymous`, but the quote access token is the authorization secret for that quote. Treat token URLs as sensitive and do not log or expose them unnecessarily.
- A valid token does not bypass quote state, access configuration, accept mode, or transition validation.

Customer example

List the authenticated customer's quotes:

```
curl -X GET "https://your-store.com/rest/V1/b2bquote/me/quotes"
\
  -H "Authorization: Bearer ${CUSTOMER_TOKEN}" \
  -H "Content-Type: application/json"
```

Keep tokens outside shell history and application logs. The variable name above is illustrative; use your deployment's secret-management approach.

Safe request order

For a customer integration, use the same order as the storefront:

1. `GET /V1/b2bquote/me/quotes` and select an owned quote;
2. add or update items with `product_id`, `buy_request`, and `qty`, or remove an item by quote and item ID;
3. set `billing` and `shipping` address objects;
4. set the shipping method code and, when required, the payment method code;
5. send messages, attachments, or a customer counter price;
6. submit the quote, then accept it with the required accept mode.

The service contract recalculates totals after item and commercial changes. Read the quote again after each mutation and use the returned `updated_at` and status before sending the next write. An ownership check is performed by the customer methods, so an ID from another customer must not be treated as an empty result.

Canonical request shapes

The exact JSON wrapper is generated by Magento Web API. The following fields are the service-contract payload, not a promise that every custom extension will accept additional fields:

```
{
  "quoteId": 42,
  "productId": 1001,
  "buyRequest": {
    "sku": "office-chair",
```

```
    "name": "Office chair",
    "price": 199.00,
    "super_attribute": {"color": "black"}
  },
  "qty": 10
}
```

For item updates, use `qty`, `offered_price`, `customer_counter_price`, and `notes`. For addresses use the quote address fields such as `firstname`, `lastname`, `street`, `city`, `region`, `postcode`, `country_id`, and `telephone`. The shipping and payment calls receive their method code. Inspect the generated Swagger schema for the surrounding interface parameter names before serializing the request.

The accept call accepts `create_order` or `add_to_cart` according to the base service contract and store configuration. Its integer response identifies the created order in `create_order` mode or the active cart in `add_to_cart` mode.

Idempotency and retries

The add-on does not add a separate idempotency key. A timeout after a mutation is ambiguous. Read the quote and order/cart state before retrying acceptance, and use a client-side request ID in your integration logs. A retry of an item add or message post can create a duplicate even if the first response was not received.

Errors and diagnostics

Code	Description
`400`	The payload or business input is invalid
`401`	The bearer token is missing, expired, or not accepted for the route
`403`	The integration lacks the declared ACL resource or the actor cannot perform the operation
`404`	The quote or related entity is not available to that context

Magento serializes service exceptions according to its Web API behavior. Do not build client logic around an undocumented assumption that every invalid transition returns one specific status code; inspect the error body and verify the target Magento API schema.

This add-on does **not** add a separate B2B Quote rate limiter, rate-limit response headers, configurable quote webhooks, or a `B2B Quote > API` configuration group. Infrastructure-level limits from a CDN, WAF, reverse proxy, or another Magento module can still apply.

When a call fails:

1. Confirm the route exists in the target store's generated API schema.
2. Confirm the token type matches the route: administrator/integration, customer, or guest access token.
3. Check the exact ACL resource for integration routes.
4. Check ownership and the quote's current state/status.
5. Reproduce with a read operation before retrying a mutation.
6. Review Magento's `var/log/system.log` and `var/log/exception.log` for the corresponding request time.

Related

- Quote Lifecycle
- Quote Management
- Admin Permissions (ACL)
- Guest Quotes
- GraphQL Add-On