GraphQL Add-On

B2B Quote Documentation

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

LAST UPDATED
2026-10-13

SOURCE
Open this article online

EXTENSION
Magento 2 Extension B2B Quote

GraphQL Add-On

Install this add-on when a PWA, mobile app, or another GraphQL client needs to work with B2B Quote. It is separate from the REST add-on and requires the base `mageb2b/b2b-quote` package.

Package: `mageb2b/b2b-quote-graph-ql`

Magento module: `MageB2B_B2BQuoteGraphQl`

Installation

1) Require package

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

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

2) Enable module + run setup

```
php bin/magento module:enable MageB2B_B2BQuoteGraphQl
php bin/magento setup:upgrade
php bin/magento cache:flush
```

Verify that Magento sees both modules:

```
php bin/magento module:status MageB2B_B2BQuote
MageB2B_B2BQuoteGraphQl
```

If your production build compiles dependency injection and static assets, run those normal deployment steps after enabling the package.

Authentication contexts

- `customerB2BQuotes`, `customerB2BQuote`, `customerB2BQuoteProjects`, and customer mutations require a signed-in customer context. Send a Magento customer bearer token.
- `guestB2BQuote` and `acceptGuestB2BQuote` use the guest quote access token.
- Broad queries such as `b2bQuotes` and administrative mutations such as `approveB2BQuote` are intended for an authorized administrator or integration context.

Ownership and transition validation still apply. GraphQL does not bypass the base module's customer-group, quote-access, or workflow rules.

Customer queries

The customer list is paginated:

```
query MyQuotes {
  customerB2BQuotes(pageSize: 10, currentPage: 1) {
    total_count
    items {
      id
      increment_id
      status
      state_code
      grand_total
      expires_at
      updated_at
    }
    page_info {
      current_page
      page_size
      total_pages
    }
  }
}
```

Fetch one owned quote with `customerB2BQuote(id: Int!)`. Use `guestB2BQuote(access_token: String!)` only for the tokenized guest flow. `customerB2BQuoteRevisions(quote_id: Int!)` returns the paginated revision history of an owned quote.

Available query groups

The installed schema provides query fields for:

- customer, guest, and administrator quote retrieval
- tags and popular tags
- categories and active categories
- messages and attachments
- projects, including customer-owned projects
- pricing rules and quote fees
- statuses, transitions, workflow rules, revisions, and workflow states

Use GraphQL schema introspection in the target Magento environment for the complete

field and input definitions. This keeps client work aligned with the exact installed schema.

Customer mutations

Customer operations include creating and submitting a quote, adding or updating items, adding messages, setting addresses, choosing shipping and payment methods, negotiating an item price, recalculating totals, applying pricing rules, accepting a quote, and supported merge/split operations. `changeB2BQuoteStatus` performs a transition-validated status change on a quote owned by the logged-in customer — the same transition, actor, and store checks apply as in Magento Admin.

Use the mutations in this order when building a client:

1. `createB2BQuote` or an existing quote from `customerB2BQuotes`;
2. `addB2BQuoteItem`, followed by `updateB2BQuoteItem` or `removeB2BQuoteItem` as needed;
3. `setB2BQuoteAddresses`, `setB2BQuoteShippingMethod`, and `setB2BQuotePaymentMethod`;
4. `addB2BQuoteMessage` and `negotiateB2BQuoteItemPrice`;
5. `submitB2BQuote`, then `acceptB2BQuote` after the approved status is returned.

Read the returned quote after every mutation. The resolver applies ownership, status transition, expiration, and configuration checks from the base service. GraphQL input validation does not make a quote eligible for submission or acceptance by itself.

Example message mutation:

```
mutation AddQuoteMessage {
  addB2BQuoteMessage(
    input: {
      quote_id: 42
      message: "Please confirm whether the quoted delivery date
includes installation."
      is_visible_on_frontend: true
      notify_customer: false
    }
  ) {
    message {
      id
      message
      created_at
    }
    quote {
      id
      increment_id
      updated_at
    }
```

```
    }
}
```

Administrator and integration mutations

The schema also contains operations to approve or reject a quote, apply workflow rules, and assign or remove tags. Use these only with an authentication context accepted by the corresponding resolver and with the same governance you apply to Magento Admin integrations.

Administrative fields and mutations are not a customer-side shortcut. Keep them on a separate integration client, grant the smallest ACL scope, and do not expose internal quote IDs or guest access tokens in a public PWA state.

Attachments and revisions

The quote schema exposes attachment and revision records for contexts that are authorized to read them. It does not provide an attachment-upload mutation. Upload new files through the storefront, Magento Admin, or the REST add-on, then use GraphQL to read the resulting attachment records. Revisions are read-only snapshots unless the corresponding administrative rollback operation is available in the target schema and the caller has its ACL permission.

Troubleshooting

The customer query returns an authorization error

Confirm that the request includes a current customer token and that the quote belongs to that customer. An Admin integration token is not interchangeable with a customer token for ownership-based fields.

A field is missing from the schema

Check that `MageB2B_B2BQuoteGraphQl` is enabled, clear Magento's configuration cache, and inspect the schema in the same environment used by the client. Do not rely on an IDE's cached schema.

A mutation is rejected

Check the quote state, allowed status transition, storefront access configuration, and ownership. The resolver delegates the business operation to the base quote services, so a syntactically valid mutation can still fail a business rule.

Related

- Quote Lifecycle
- REST API Add-On

- Troubleshooting