



OpenSign 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](#)

OpenSign Add-On

The OpenSign add-on connects an approved B2B quote to a self-hosted OpenSign instance. When signatures are required, the buyer must sign the generated quote PDF before accepting the offer. Magento keeps the signature state and a copy of the completed PDF with the quote.

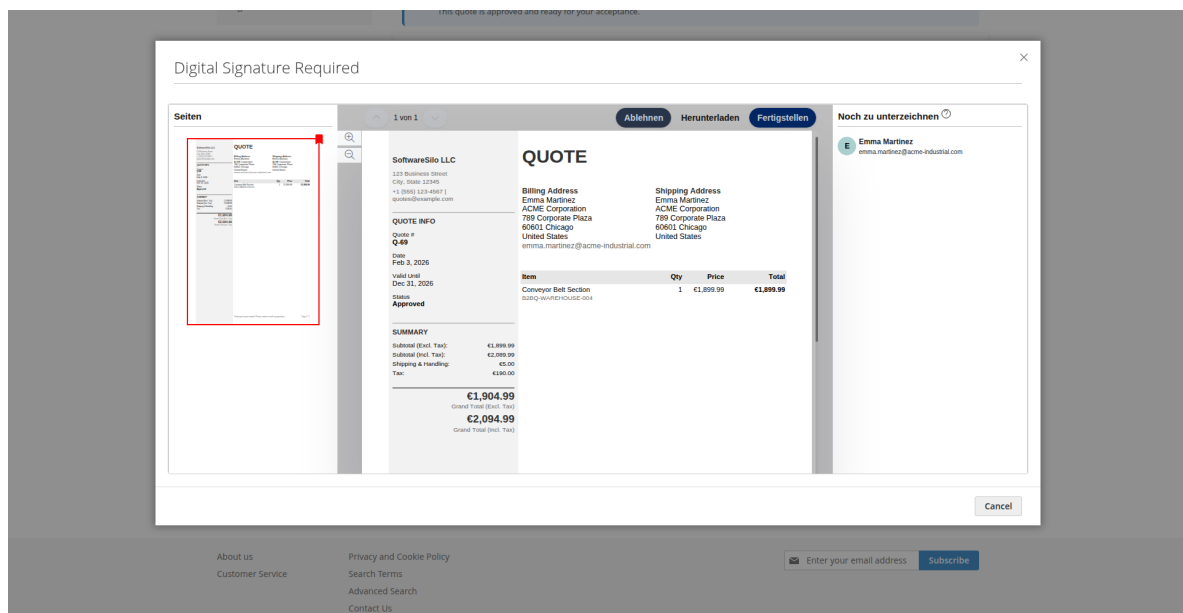
Package: `mageb2b/b2b-quote-opensign`

Magento module: `MageB2B_B2BQuoteOpenSign`

How the signing flow works

1. The customer opens an approved quote.
2. Magento generates its quote PDF and uploads it to OpenSign.
3. OpenSign creates a document, contact-book signer, and signature field.
4. The customer signs in the embedded signing screen.
5. Magento records the completed request, downloads the signed PDF, and moves the quote to the signed status.
6. The customer can continue with quote acceptance.

The add-on prevents signature creation and status updates for a quote that is not approved. It also checks quote ownership before exposing a customer's signing action.



Install the package

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
```

```
composer require mageb2b/b2b-quote-opensign:*

php bin/magento module:enable MageB2B_B2BQuoteOpenSign
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
php bin/magento cache:flush
```

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

Configure OpenSign

Open **Stores** > **Configuration** > **MageB2B** > **B2B Quote** > **OpenSign Digital Signatures**.

Connection and identity

Setting	Purpose
OpenSign API URL	Server-facing Parse API base URL used by Magento
OpenSign Public URL	Browser-facing base URL used for the signing interface and files
Application ID	OpenSign Parse application ID
Session Token	Preferred API credential when configured
Master Key	Fallback API credential when no session token is configured
System User ID	OpenSign <code>_User</code> object ID used as document creator
System Contracts User ID	Matching <code>contracts_Users</code> object ID
Webhook Secret	Shared secret required for completion callbacks

The session token, master key, and webhook secret are stored through Magento's encrypted configuration backend. Limit access to the configuration section and never paste these values into a support ticket or screenshot.

Separate internal and public URLs

Containerised installations often need two routes to the same OpenSign service:

- Magento uses an internal address, for example `http://opensign-proxy:3001/api/app`.
- The buyer's browser uses a public HTTPS address, for example

<https://sign.example.com>.

Do not put a container-only hostname into **OpenSign Public URL**. The signing modal and file links must be reachable from the buyer's browser. Conversely, confirm that the Magento runtime can resolve and reach the API URL before enabling the workflow.

Signature policy

Setting	Effect
Enable Digital Signatures	Enables the integration for the selected website scope
Require Signature for Acceptance	Blocks acceptance until a successful signature exists
Signature Request Expiry (Days)	Sets the lifetime of a newly created signature request; default is seven days

Save the configuration and use **Test Connection**. A successful connection check does not replace a real signing test because PDF upload, browser routing, certificate use, and callback handling occur later in the flow.

Configure the OpenSign service

OpenSign needs a valid P12/PFX certificate and its matching passphrase to finalise signed PDFs. Configure the OpenSign server's **PFX_BASE64** and **PASS_PHRASE** according to the OpenSign deployment documentation. Keep its public/server URL settings aligned with the Magento public and API URLs.

For the buyer workflow, OpenSign must allow the unauthenticated recipient route generated by the add-on. Do not replace it with a route that requires the customer to have an OpenSign account or session.

Verify the complete workflow

Use a non-production approved quote belonging to a fictional demo customer:

1. Open the quote as that customer and start the signature.
2. Confirm that the signing screen displays the correct quote document and signer.
3. Complete a typed signature, then repeat with drawn and uploaded signatures if those modes are allowed by the OpenSign setup.
4. Confirm that OpenSign reports success.
5. Reload the Magento quote and confirm its signed state.
6. Confirm that the signed PDF is available to the customer and an authorised administrator.
7. Accept the signed quote and verify the normal conversion workflow.

The administrator quote view shows the signer, request status, signed date, document identifier, and signed-PDF download when available.

Webhook security

Configure the same high-entropy secret in Magento and the OpenSign callback. The module accepts the shared secret in the `X-OpenSign-Webhook-Secret` header or supported callback payload. Expose the callback over HTTPS and do not disable its verification to work around a routing problem.

The completion process preserves terminal signed requests and ignores unsuccessful remote responses. Magento stores the signed PDF locally so it does not depend on a short-lived OpenSign file token later.

Troubleshooting

The connection test fails

Check DNS and network reachability from the Magento runtime, not only from the host browser. Confirm the API URL, application ID, system IDs, and active session token or master key.

The signing screen does not load

Open the browser console and network panel. The usual causes are an internal hostname in the public URL, HTTPS/mixed-content restrictions, frame/CSP restrictions, or a signing URL that requires an OpenSign login.

OpenSign says the PDF is incompatible

If typed signatures work but drawn or uploaded signatures fail, check the OpenSign deployment and the signature-field dimensions expected by that deployment. The add-on sends both the lowercase and uppercase width/height values required by the verified OpenSign flow.

Finalisation reports an ASN.1 or PFX error

The OpenSign certificate data is missing, truncated, incorrectly encoded, or paired with the wrong passphrase. Regenerate and validate the P12/PFX value in the OpenSign environment; changing Magento quote data will not resolve this error.

The quote remains unsigned after OpenSign succeeds

Check the callback request, shared webhook secret, Magento logs, local signature-request status, and whether the quote was still approved. Do not manually change the quote status before confirming whether the signed PDF was downloaded and stored.

Related guides

- [Quote Lifecycle](#)
- [PDF Export & Layouts](#)
- [Cart and Order Conversion](#)