Punchout Documentation

Complete PDF Manual

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

LAST UPDATED
2026-11-05

SOURCE
Open complete documentation online

EXTENSION
Magento 2 Extension Punchout

Table of Contents

Punchout Documentation · punchout

Overview
Punchout
Getting Started
Installation
Configuration
Go-Live Checklist
Features
Buyer and Mapping Profiles
Authentication and Security Model
Cart API and Session Lifecycle
Storefront Operation Policy
cXML Profile Discovery
Security and Idempotency
Inbound Orders and Pricing
Document Loop and Queue
Analytics
Operations and CLI
Trace and Simulator
Protocol Compatibility
Endpoint Reference
Admin Field Reference
Extension Points
Templates Overview
OCI and SAP Templates
cXML Core Templates
SAP Ariba Templates
Coupa Templates
Oracle Procurement Templates
Template Selection Guide
JAGGAER cXML Setup
Flows
OCI Flow
cXML Flow

Add-Ons
Iframe Compatibility Add-On
Sample Data Add-On
Reference
Configuration Fields
Database Reference
Troubleshooting
Common Issues
Error Catalog
FAQ

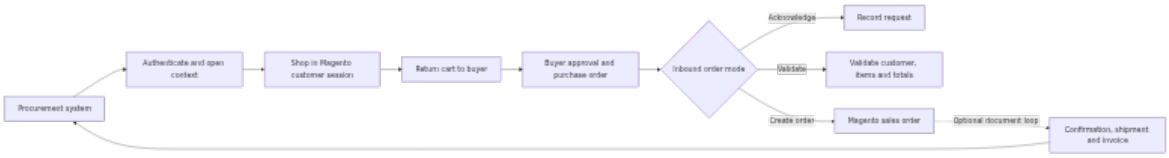
Punchout

SoftwareSilo PunchOut connects a Magento 2 catalog to a buyer's procurement system. A buyer starts an OCI or cXML session, shops in Magento with the customer and pricing context resolved by the buyer profile, and sends the finished cart back instead of placing a normal storefront order.

Use this guide if you are integrating Magento Open Source or Adobe Commerce with an OCI or cXML buyer, including projects involving SAP, SAP Ariba, Coupa or Oracle Procurement.

How the workflow fits together

1. An active **buyer profile** identifies the procurement account and authenticates its request.
2. An OCI login or cXML setup request creates a short-lived PunchOut context.
3. Magento activates the resolved guest or customer session and applies the configured storefront-operation policy.
4. The buyer builds a cart in Magento or through the tokenized cart endpoints.
5. **Return to System** maps the cart and posts it to the buyer's return URL.
6. After approval, a cXML buyer can send an `OrderRequest`. Depending on the configured processing mode, Magento records it, validates it or creates a sales order.
7. Magento can send confirmation, shipment and invoice cXML documents back through the document queue.



Terms used in this guide

- **PunchOut** — the name of the integration pattern: a buyer "punches out" of their procurement system into the supplier's shop and returns the filled cart for internal approval instead of ordering directly.
- **Procurement system** — the buyer-side purchasing software (for example SAP Ariba, Coupa or Oracle Procurement) that opens the PunchOut session and later receives the cart.
- **OCI (Open Catalog Interface)** — the simpler, form-based PunchOut protocol; login credentials and the cart-return URL arrive as HTML form fields.
- **cXML (commerce XML)** — the richer XML-based PunchOut protocol; authentication, cart return and order documents are XML messages defined by the cXML standard (its structure rules are called the DTD).
- **Buyer profile** — the Magento Admin record that identifies one procurement account:

credentials, return URLs, mapping rules and operation policy.

- **BuyerCookie** — the session cookie a cXML buyer sends back with each request so Magento can continue the PunchOut session.
- **Extrinsic** — a named key/value pair inside a cXML request or cart line (for example a buyer's cost center); mapping profiles read extrinsics into Magento fields.
- **Level-2 data** — the per-line detail in a returned cart (SKU, quantity, unit price) that procurement systems need for approval; some platforms additionally expect tax and shipping breakdowns.
- **JIT (just-in-time)** — provisioning: creating the Magento customer account automatically when a buyer first logs in, instead of pre-creating accounts.
- **Idempotent** — safe to retry: processing the same request twice produces the same result once, so a repeated `OrderRequest` does not create duplicate orders.

Start here

- Installation
- Configuration
- Go-Live Checklist
- Buyer and Mapping Profiles

Protocol flows

- OCI Flow
- cXML Flow
- cXML Profile Discovery
- Protocol Compatibility
- Endpoint Reference

Storefront and operations

- Cart API and Session Lifecycle
- Storefront Operation Policy
- Document Loop and Queue
- Inbound Orders and Pricing
- Trace and Simulator
- Analytics
- Operations and CLI
- Extension Points

Mapping templates

The module ships editable starter mappings for generic OCI and cXML as well as SAP, SAP

Ariba, Coupa and Oracle-oriented projects. Start with the Template Selection Guide, then validate the copied rules against the buyer's current specification. A template name is not a certification for that platform or buyer.

Add-ons

- Iframe Compatibility prepares an authenticated PunchOut session for procurement platforms that embed the Magento shop in an iframe.
- Sample Data installs a connected OCI and cXML example workflow for local, demo and staging environments.

See the Add-Ons overview before installing either package. The normal top-level PunchOut flow needs neither add-on.

Important boundaries

- PunchOut transfers a shopping cart; it does not replace the buyer's approval or purchase-order process.
- Returning a cart does not place an order. Order creation begins only when an authenticated cXML `OrderRequest` reaches the inbound-order endpoint and the effective processing mode is **Create Magento Order**.
- Buyer-profile inbound-order values override the selected Store view. Leave an override empty when the buyer should inherit Store-view configuration.
- Confirmation, ASN and invoice documents use their own HTTPS endpoints. The browser cart-return URL is never reused for server-to-server documents.
- Fulfilment documents use durable buyer and order routing. They remain deliverable after the short-lived shopping session has expired.
- SAP Ariba, Coupa and Oracle requirements vary by tenant and buyer. Treat the supplied mappings as a starting point.
- Project-specific actor and permission behavior can be added through the public service contracts and Magento events without modifying the package.

For a failed request, begin with Common Issues and the Error Catalog.

Installation

Install the module first, then create one buyer profile and one outbound mapping profile. Do not connect a production procurement account until the login, cart return and retry behavior have been tested end to end.

Requirements

- Magento 2.4
- PHP `8.1` or newer within the PHP version supported by your Magento release
- HTTPS for Magento and every production return URL
- Magento cron running for cleanup and document-queue processing
- Magento Inventory (MSI) modules enabled; `MageB2B_Punchout` requires `Magento_InventoryApi` and `Magento_InventorySalesApi`, which supply the salable quantity for OCI `QUANTITYCHECK` responses

Install the package

Run these commands from the Magento project root:

```
composer require mageb2b/punchout
php bin/magento module:enable MageB2B_ExtensionManager
MageB2B_Punchout
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento cache:clean
```

Deploy static content when your release process requires it:

```
php bin/magento setup:static-content:deploy -f
```

Confirm the admin areas

Open **Punchout** in the main Magento Admin navigation. The menu contains:

- Punchout Buyers
- Punchout Mappings
- Punchout Trace
- Document Queue
- Inbound Orders
- Analytics
- Punchout Simulator
- Configuration

If an entry is missing, check the administrator role under **System > Permissions > User Roles**. PunchOut has separate permissions for buyers, mappings, traces, inbound orders, the document queue, analytics, the simulator and configuration.

Create the first working profile

1. Open **Punchout > Punchout Mappings**.
2. Add an outbound profile for the required protocol.
3. Select a matching starter template, save the copied rules, and keep the profile active.
4. Open **Punchout Buyers** and add the external buyer identity.
5. Enter the OCI credentials or cXML sender credentials supplied by the procurement team.
6. Select the customer-resolution mode and assign the new outbound mapping profile.
7. Keep the buyer inactive until the values have been reviewed.

Use a buyer identifier that belongs to the integration, such as `NORTHWIND-PROCUREMENT-EU`, rather than a person's name or an environment password.

Run the first test

Enable trace logging and use **Punchout Simulator** to build the request. A useful first test proves all of the following:

- the buyer is resolved in the intended store view;
- protocol and optional transport credentials are accepted;
- a context token and correlation ID are created;
- the storefront session opens with the expected guest or customer context;
- the cart is returned with the fields required by the buyer;
- the trace contains sanitized request and response information.

For cXML projects that create Magento orders, continue with one approved `OrderRequest`. Confirm the effective inbound processing mode, returned-cart price policy, customer and product resolution, shipping, payment and exact-retry behavior before go-live.

Uninstall the module

Uninstalling `MageB2B_Punchout` runs a destructive cleanup: every `punchout_*` table is dropped (buyer profiles, mappings, session contexts, traces, returned-cart snapshots, inbound orders, document-queue rows and analytics data) and all `core_config_data` entries under `punchout/` and `payment/punchout_purchaseorder/` are removed. The operation is not reversible, so export any data you still need before running `bin/magento module:uninstall`.

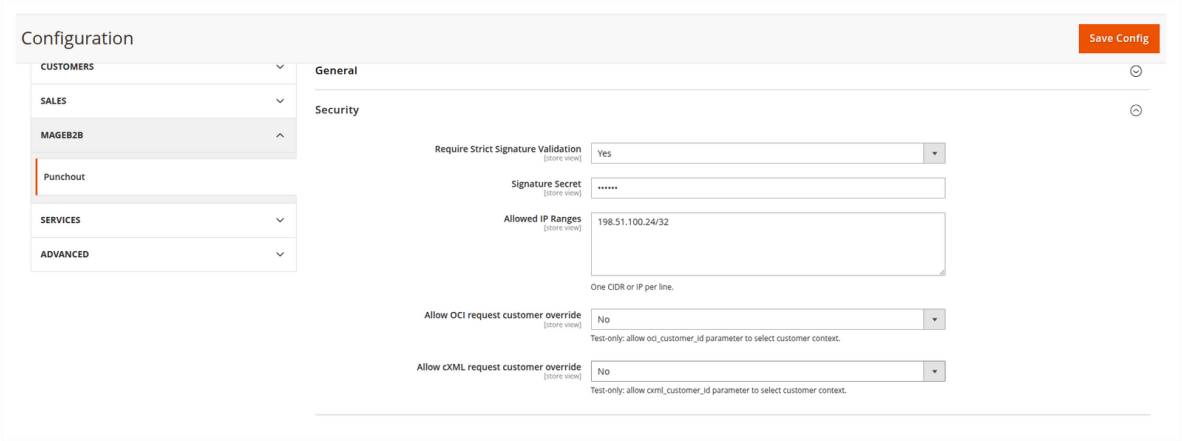
The `mageb2b/punchout-iframe` add-on ships its own uninstall step that drops the

`punchout_iframe_buyer_config` table and removes the `punchout/iframe/` configuration values.

Continue with Configuration before exposing the endpoints to a production buyer.

Configuration

Open **Punchout > Configuration** and choose the correct website or store view before saving. Store-specific settings are important when buyers use different catalogs, currencies or security policies.



General

Setting	Default	What it changes
Enable Punchout	Yes	Enables setup, cart, transfer, session and document processing for the selected scope.
Punchout return action label	Return to System	Replaces the checkout action during an active PunchOut storefront session.
Enable Sandbox Callback Mode	No	Exposes `/punchout/sandbox/callback` so a test return payload can be inspected. Never leave this on in production.

Security

Strict signature validation

Strict mode verifies an HMAC-SHA256 signature made with the configured signature secret. The caller sends the 64-character digest in `X-Punchout-Signature` or the `signature` parameter. A `sha256=` prefix is accepted.

This signature is a SoftwareSilo security layer, not a value that every OCI or cXML platform sends automatically. Enable it only after the procurement integration has implemented and verified the same signing rule. For a raw request body, sign that body. For parameter requests, the module removes the signature field, sorts associative keys and signs the normalized JSON representation.

Allowed IP ranges

Enter one IPv4/IPv6 address or CIDR range per line. An empty field permits requests from every source IP. In production, use the procurement provider's current outbound ranges and confirm that Magento sees the real client IP through your proxy or load balancer.

Request customer overrides

`Allow OCI request customer override` and `Allow cXML request customer override` allow request parameters to select a Magento customer. Both are disabled by default and labelled test-only. Keep them off in production; assign the customer on the buyer profile instead.

Outbound TLS

TLS certificate verification is enabled by default. `Allow insecure outbound TLS in sandbox mode` can disable certificate checks only while sandbox mode is active. It is intended for isolated test systems, not production.

Customer identity and JIT provisioning

The Store-view **Customer Identity** settings control whether customer-by-e-mail resolution can create an account when no Magento customer exists. Select the customer group used for those accounts before enabling JIT provisioning.

Each Buyer Profile selects one explicit resolution mode:

- **Guest** creates no Magento customer session;
- **Fixed Magento Customer** uses the selected customer;
- **Customer E-mail** resolves the e-mail extracted from the configured OCI or cXML source and can use JIT provisioning;
- **Optional Request Customer ID** uses a permitted request ID when present;
- **Required Request Customer ID** rejects a request without that ID.

Magento's configured account-sharing and account-confirmation policies remain authoritative. When confirmation is required, Magento sends its normal confirmation message and PunchOut cannot continue as that new account until it is confirmed.

Cart pricing

`Allow inbound cart item prices` is disabled by default. With that setting off, a price supplied to `/punchout/cart/add` is ignored and Magento calculates the product price. Enable buyer-supplied prices only when the integration contract requires them and the caller is trusted to set them.

cXML Profile Discovery

Profile Discovery is enabled by default for cXML 1.1 and 1.2 buyers. Magento builds the

response from the Store View and Buyer Profile instead of a separate transaction list.

Leave **Public PunchOut Base URL** empty when the Store View base URL is publicly correct. Set it only when a reverse proxy or external gateway exposes Magento under a different scheme or host. The value must not contain a query string or fragment.

The Buyer Profile's **Accept PunchOutSetupRequest** and **Accept OrderRequest** switches decide which transactions are advertised and accepted. See cXML Profile Discovery.

Inbound Orders

These Store-view settings control an approved cXML `OrderRequest`. They do not affect the earlier cart-return step.

Customer Identity

Cart

Inbound Orders

Processing Mode [store view]	Acknowledge Only Acknowledge Only records the request without creating a quote or order.
Authoritative Price Source [store view]	Current Magento Pricing Current Magento Pricing executes the normal customer-specific catalog and quote pricing pipeline.
Returned Cart Line Matching [store view]	Auxiliary ID, then SKU Used by snapshot-based pricing. Auxiliary ID is preferred when available.
Changed Quantity Action [store view]	Recalculate with Magento
New Item Action [store view]	Recalculate with Magento
Removed Item Action [store view]	Allow Removal
Missing/Expired Snapshot Action [store view]	Reject Applies when snapshot-based pricing cannot resolve the original returned cart.
Price Mismatch Action [store view]	Reject Order
Absolute Unit Price Tolerance [store view]	0.01 Difference in order currency allowed per unit.
Percent Unit Price Tolerance [store view]	0 Percentage difference allowed per unit. Either configured tolerance can permit the price.
Customer Resolution [store view]	Buyer Profile Customer
Product Resolution [store view]	Supplier Part ID
Address Source [store view]	Incoming Order Addresses
Shipping Method [store view]	[flatrate] Fixed The selected method must be available for the resolved quote and destination.
Payment Method [store view]	PunchOut Purchase Order Defaults to the included offline method punchout_purchaseorder.
Order Update Policy [store view]	Manual Review
Order Cancellation Policy [store view]	Manual Review
Send Customer Order Email [store view]	No
Send cXML ConfirmationRequest [store view]	No Sends a confirmation after Magento creates or cancels an order.
ASN URL [store view]	cXML endpoint that receives ShipNoticeRequest documents. This is not the PunchOut cart return URL.
Invoice URL [store view]	cXML endpoint that receives InvoiceDetailRequest documents. This is not the PunchOut cart return URL.
Supplier Credential Domain [store view]	cXML credential domain used for outbound ConfirmationRequest, InvoiceDetailRequest and ShipNoticeRequest messages.
Supplier Identity [store view]	
Supplier Shared Secret [store view]	
Maximum Processing Attempts [store view]	5
Retry Batch Size [global]	20
Retry Delay (minutes) [store view]	15
Outbound cXML Response Mode [store view]	Allow Empty HTTP 2xx Response Controls whether a successful HTTP response must include a valid cXML Response Status.
Inbound Order Payload Retention (days) [store view]	365 Removes retained request details after the configured period. Order identity and idempotency evidence are retained.
Returned Cart Snapshot Retention (days) [store view]	90 Must cover the longest expected requisition and approval workflow.

Processing and pricing

- **Processing Mode** chooses Acknowledge Only, Validate Only or Create Magento Order.
- **Authoritative Price Source** chooses current Magento pricing, the returned-cart snapshot, incoming prices or the hybrid original-if-unchanged policy.
- **Returned Cart Line Matching** uses auxiliary ID then SKU, or SKU only.
- **Changed Quantity**, **New Item**, **Removed Item** and **Missing Snapshot** settings decide whether Magento recalculates, allows, rejects or sends a request to manual review.
- **Price Mismatch Action** works with the absolute and percentage unit-price tolerances.

For most approval workflows, start with **Original Price if Unchanged; Otherwise Current Magento Price**. It keeps the price Magento returned for an unchanged line and sends changed quantities or new products through current customer-specific pricing.

Magento order context

Choose how Magento resolves the customer, product and addresses. Then select a shipping method and payment method available to the resulting quote. The included **PunchOut Purchase Order** method is intended for approved procurement orders.

Update and cancellation policies can reject a request, send it to manual review or perform their supported automatic action. **Send Customer Order Email** controls Magento's normal new-order email independently from cXML confirmation delivery.

Confirmation, document routing, retries and retention

Enable `ConfirmationRequest` only after the buyer has supplied a reachable HTTPS endpoint and supplier credentials are complete. Configure the ASN and Invoice URLs separately; neither document is sent to the browser cart-return URL. Buyer Profile URLs override the Store-view defaults. Retry settings control inbound processing attempts. Keep the returned-cart snapshot longer than the longest expected requisition and approval cycle.

Buyer-profile overrides take precedence over these Store-view values. Leave an override empty to inherit the selected Store view. See Inbound Orders and Pricing for the decision flow.

Trace and session

- **Enable Trace Logging** defaults to Yes. Payloads are sanitized before storage, but trace access should still be limited to integration and support administrators.
- **Trace Retention** defaults to 30 days. The daily cleanup job removes older traces.
- **Token Expiration** defaults to 2 hours and accepts decimal values such as `0.5`. Choose a duration long enough for the buyer's shopping flow. An invalid or non-positive value prevents context creation.

Analytics

Enable PunchOut Analytics controls the daily aggregates shown under **Punchout > Analytics**. The setting is enabled by default and follows Magento scope inheritance. The aggregates are independent of trace retention, so clearing old traces does not remove the reporting history.

Document Queue

The queue is enabled by default. Magento adds shipments and invoices belonging to PunchOut-created orders and delivers them to the effective Buyer Profile or Store-view ASN and Invoice URLs. Encrypted routing and payload data are retained with the order and queue row, so delivery does not depend on the expired browser session.

Setting	Default	Effect
Enable Queue Processing	Yes	Allows the five-minute cron job to process due documents.
Batch Size per Cron Run	20	Maximum due rows processed per store in one run.
Maximum Retry Attempts	5	Attempts recorded on a newly queued document.
Retry Delay	15 minutes	Delay before a failed row becomes eligible again.
Completed Queue Retention	365 days	Removes completed and failed operational queue rows after the selected period.

Definite HTTP failures follow the retry policy. A transport interruption after the request may have reached the receiver becomes **Delivery Unknown** and is not retried automatically. Verify the procurement system first, then use the separately protected Retry Unknown Delivery action only when another delivery is safe.

Retention and debug logging

Request replay retention is configurable and preserves completed results for safe idempotent retries. Inbound payload, returned-cart snapshot, trace and completed queue retention are configured independently. The cleanup batch size is a Default Config setting used by the daily retention job.

Store-scoped debug logging writes additional non-secret diagnostics to `var/log/punchout.log`. Enable it only during a controlled investigation and disable it afterward.

Frontend operation policy

The policy is enabled by default and applies different route lists to `create`, `edit`, `inspect`, OCI `DETAIL` and OCI `DETAILADD` sessions. The supplied defaults allow normal catalogue and cart editing for `create` and `edit`. Inspect reconstructs the supplied cart as read-only. DETAIL opens only the selected product, while DETAILADD allows that product's Add to Cart action and immediately returns the one selected line. Read Storefront Operation Policy before changing these lists; an incorrect full-action name can block a valid buyer journey.

For the literal configuration paths and field types, use Admin Field Reference.

Go-Live Checklist

Complete this checklist with the Magento owner and the buyer's procurement team. A successful simulator request alone is not a production acceptance test.

Buyer and store context

- The active buyer ID exactly matches the inbound OCI or cXML identity.
- OCI username/password or cXML sender domain, identity and shared secret have been exchanged securely.
- Optional transport authentication is configured on both sides, if used.
- The buyer points to the intended Magento Store View and uses the agreed customer resolution mode.
- E-mail identity sources, JIT customer group and Magento account-confirmation behavior have been tested when JIT provisioning is used.
- OCI buyers use the agreed OCI version and external product-ID source.
- Catalog visibility, customer-specific prices, tax mode and currency match the buyer's expectations.
- The assigned mapping is an active outbound runtime profile, not a synced template profile.

Network and request security

- Magento and the buyer return URL use valid HTTPS certificates.
- The return hostname resolves to public IP addresses in production.
- The IP allowlist contains the buyer's current egress addresses; it is not left empty unintentionally.
- If strict signatures are enabled, the caller has proved the exact HMAC-SHA256 signing behavior with both form and raw-body requests it will send.
- Test-only customer overrides and insecure TLS are disabled.
- Sandbox callback mode is disabled.

End-to-end workflow

- The procurement system starts an OCI login or cXML setup request successfully.
- The returned token opens the expected Magento customer session.
- Search, product visibility and cart pricing are correct.
- `create`, `edit`, `inspect`, `DETAIL` and `DETAILADD` allow only their intended storefront actions when those operations are part of the integration.
- Return to System posts the expected field names, quantities, units, prices and currency.
- The procurement system accepts the returned cart and preserves the correlation or

buyer-cookie value it needs.
- A repeated transfer with the same `X-Punchout-Request-Key` and body returns the saved response; reusing the key with changed input returns HTTP 409.

Approved OrderRequest, when included

- The procurement system posts to `/punchout/document_po/inbound` only after its approval process is complete.
- The processing mode is explicitly set to Acknowledge Only, Validate Only or Create Magento Order for the selected Store view.
- Buyer-profile overrides are documented and empty fields intentionally inherit Store-view configuration.
- The procurement system preserves the original BuyerCookie through cart return and the later OrderRequest.
- The snapshot retention period is longer than the longest requisition and approval cycle.
- Unchanged, changed, added and removed lines have each been tested against the selected returned-cart policy.
- Customer, product, address, shipping and payment resolution produce the intended Magento quote.
- Exact OrderRequest retries do not create duplicate inbound records or Magento orders.
- Update and cancellation requests follow the selected manual-review, reject or automatic policy.

Outbound document loop, when included

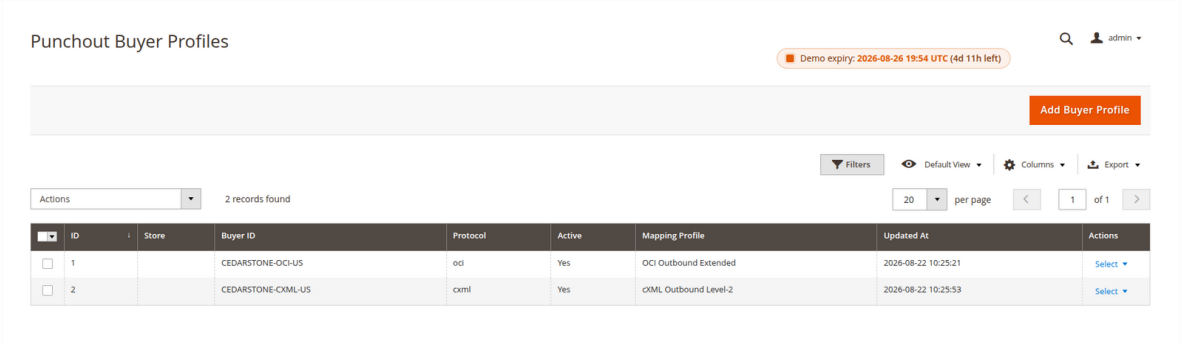
- Shipment and invoice creation produce one queue row per Magento entity.
- The buyer accepts generated ASN and invoice cXML at the dedicated HTTPS endpoints; the cart-return URL is not used.
- Supplier credentials are configured before ConfirmationRequest, InvoiceDetailRequest or ShipNoticeRequest delivery is enabled.
- Cron processes due rows every five minutes.
- A controlled delivery failure follows the configured retry delay and reaches `failed` after the configured maximum.
- A simulated uncertain transport outcome reaches **Delivery Unknown** and is never retried automatically.
- Support staff understand View, Reset, Force Run and the separately authorized Retry Unknown Delivery action without exposing payload secrets.
- Shipment and invoice delivery still works after the browser PunchOut session has expired.

Support handoff

- Trace logging and an appropriate retention period are enabled for the launch window.
- Support staff know how to find a request by correlation ID.
- Buyer secrets are not copied into tickets or trace exports.
- An owner is assigned for Magento, networking and the procurement platform.
- The rollback action is known: deactivate the affected buyer profile or disable PunchOut only in the affected scope.

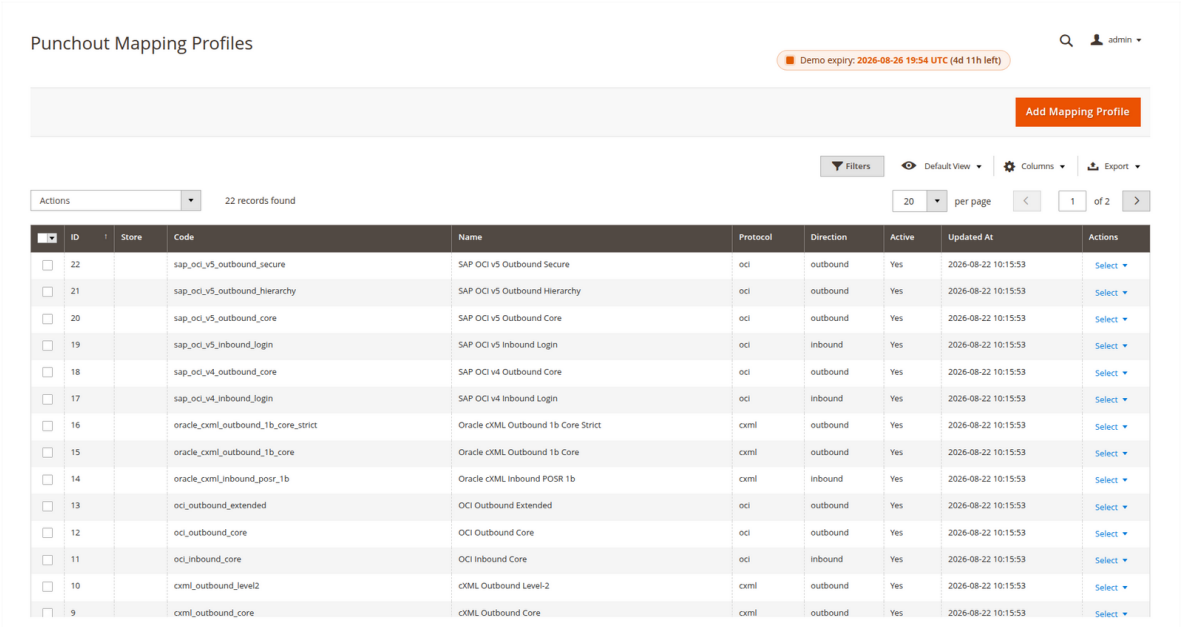
Buyer and Mapping Profiles

A buyer profile answers **who may start a PunchOut session**. A mapping profile answers **which Magento values are returned to that buyer**. Keep those responsibilities separate: one buyer can point to one active outbound mapping, while a copied mapping can be reused by several buyers with the same protocol and field contract.



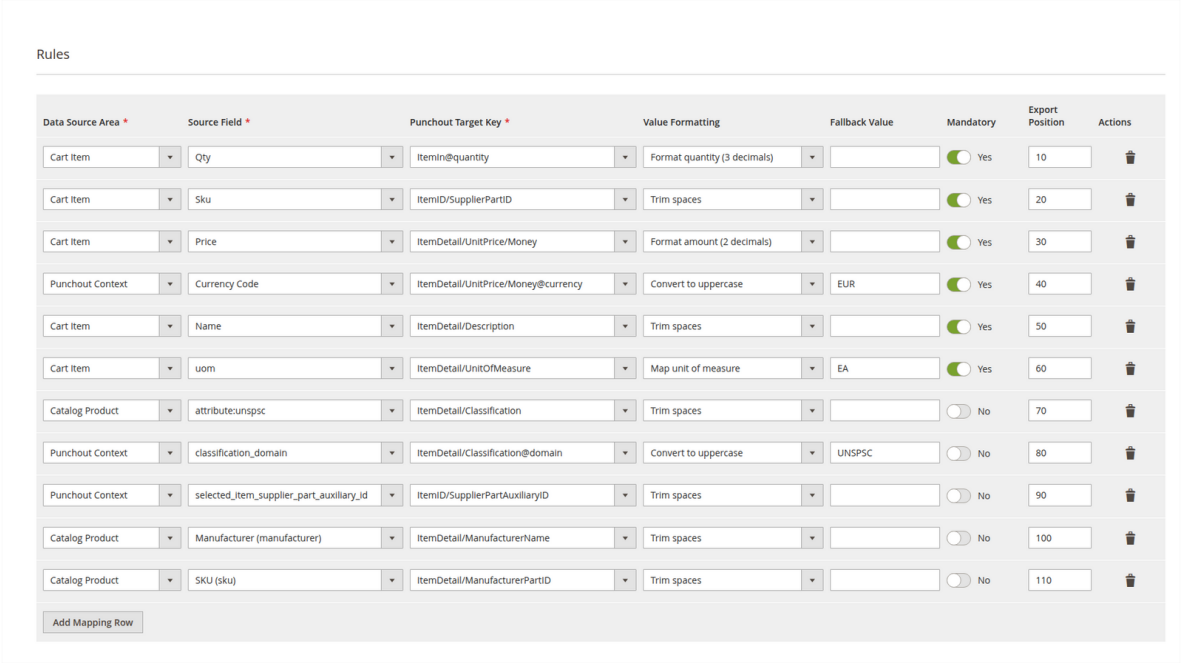
Create the outbound mapping first

Open **Punchout > Punchout Mappings** and add a profile.



1. Select the store scope. A global profile is available to all stores; a store-specific profile is available only in that store.
2. Enter a stable lowercase code using letters, numbers, hyphens or underscores.
3. Choose `oci` or `cxml` and set the direction to `outbound`.
4. Select **Start From Template** while creating the profile.
5. Review every copied rule and save the profile as active.

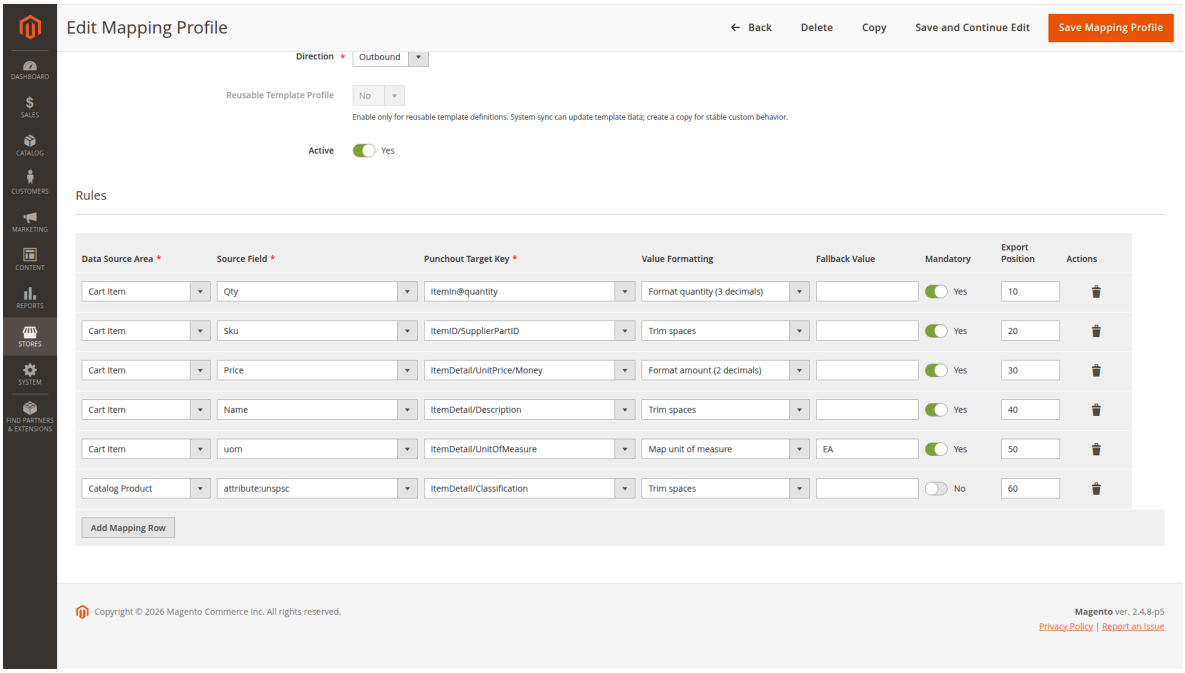
The template selector is available only for a new profile. Magento copies the rules into the profile; later template synchronization does not overwrite that runtime copy. Profiles marked **Reusable Template Profile** are definitions for future copies and cannot be assigned to buyers.



Understand one mapping rule

Each row performs one small conversion:

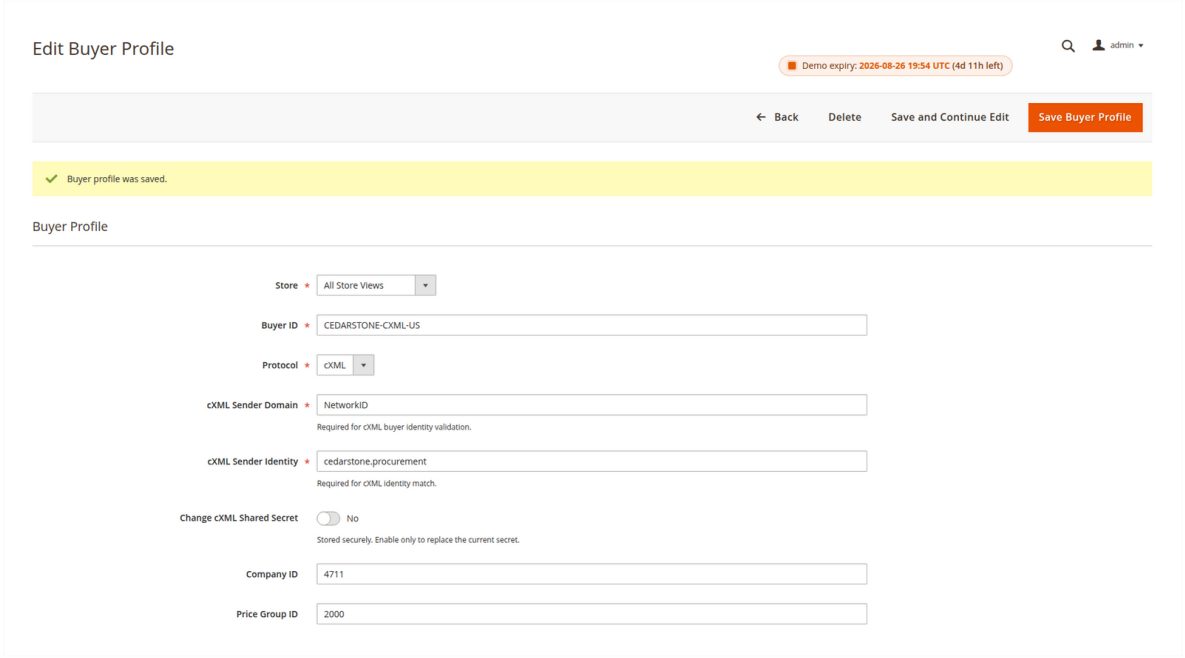
- **Data Source Area** selects cart item, product, customer, address or PunchOut-session data.
- **Source Value** selects the exact field in that area.
- **Punchout Target Key** is the outgoing OCI or cXML field.
- **Custom Target Key** allows a buyer-specific key not present in the supplied list.
- **Value Formatting** trims, formats or normalizes the source value.
- **Fallback Value** is used only when the source is empty.
- **Mandatory** stops the transfer when the resulting value is empty.
- **Export Position** keeps output evaluation deterministic.



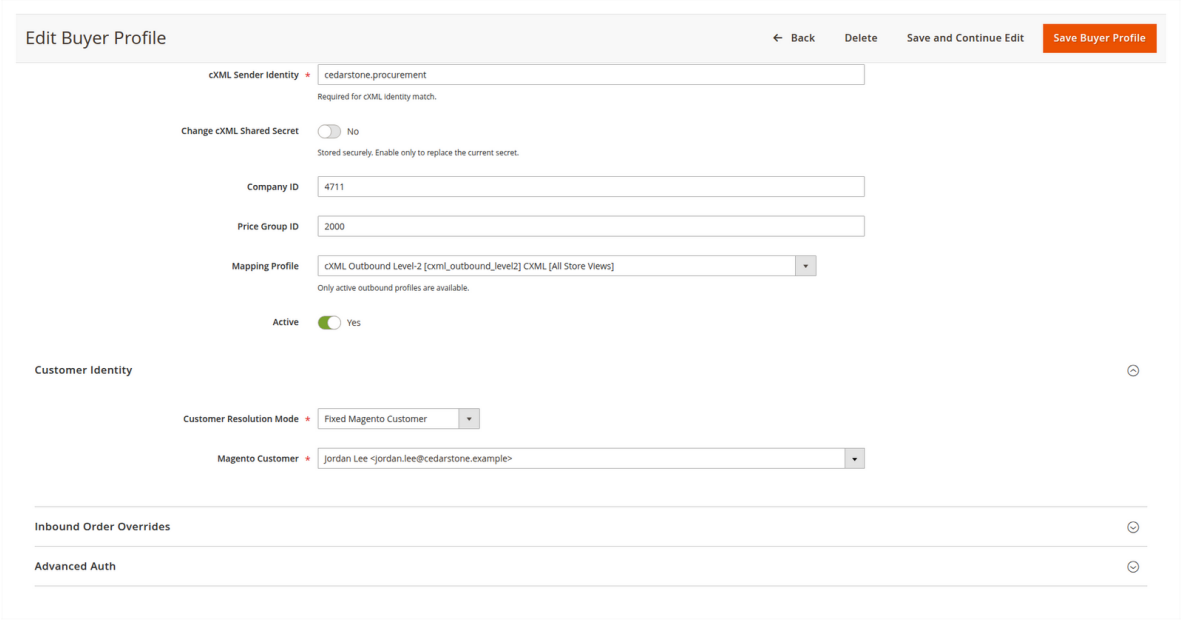
Use mandatory rules for fields the buyer truly rejects when absent. A strict profile can make a transfer fail because optional Magento data such as manufacturer or classification has not been maintained.

Create the buyer profile

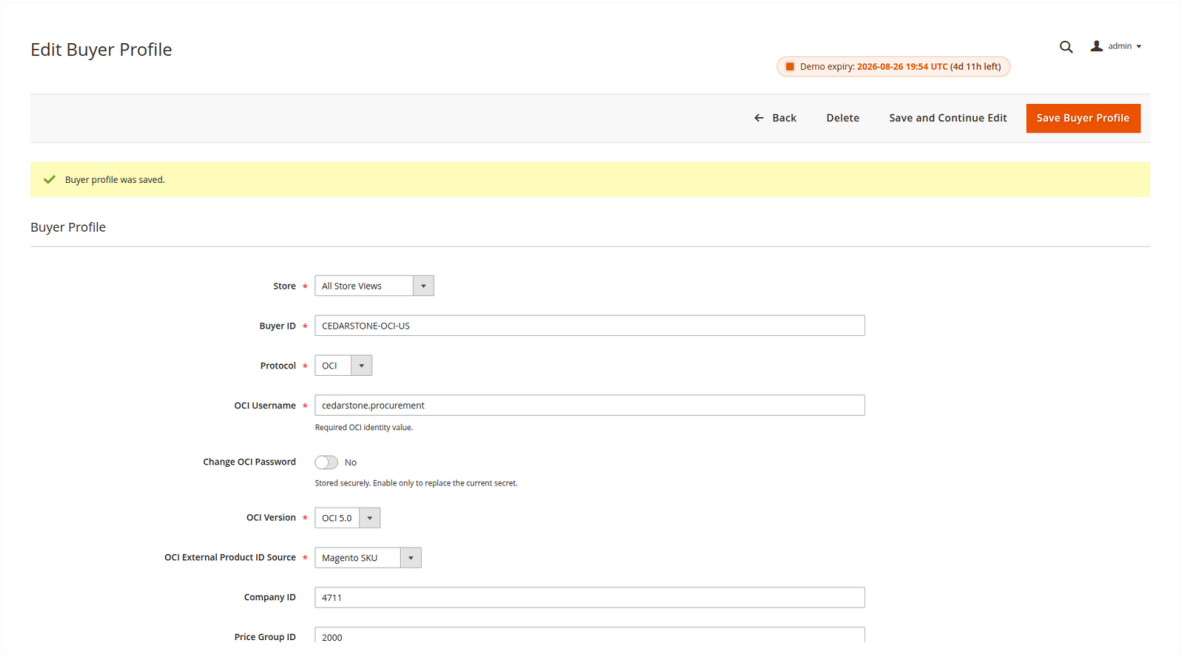
Open **Punchout Buyers** and add the procurement account.

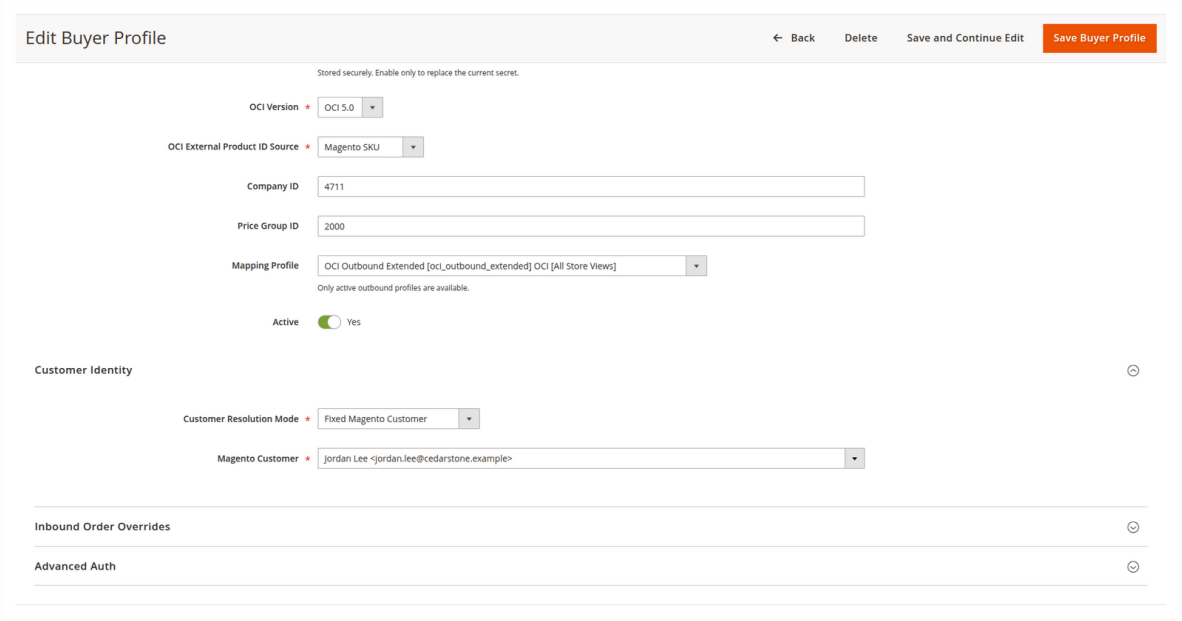


The lower part of the same profile binds the buyer to its outbound mapping and fixed Magento customer without exposing the stored secret:



The OCI form exposes the protocol-owned version and product identifier alongside its protected password control:





Identity and scope

- **Store** controls which Magento store, catalog and scoped configuration apply. A store-specific buyer wins over a matching global buyer.
- **Buyer ID** is the external identifier used to resolve the account.
- **Protocol** must match the inbound request and assigned mapping.
- **Customer Resolution Mode** chooses guest access, one fixed Magento customer, customer e-mail mapping, or optional/required request customer ID mapping.
- **Magento Customer** is selected only for fixed-customer mode. Magento verifies that the customer belongs to the buyer profile's website.
- **Identity Source** identifies the OCI parameter or cXML data field used for dynamic customer resolution.
- **JIT Provisioning** can create a customer when e-mail mapping does not find an existing account. The new account uses the configured customer group and Magento's normal account-confirmation policy.
- **Company ID** and **Price Group ID** are context values for compatible integrations and mapping rules; the core package does not create a company or pricing model from them.

Protocol credentials

For OCI, enter the configured username and password. The request may supply `USER` or `username`, and `PASSWORD`, `password` or `pwd`.

An OCI buyer profile also owns its OCI version and external product identifier. Use Magento SKU unless the procurement system requires a different globally unique product attribute. The configured identifier is used consistently for `PRODUCTID` input and `NEW_ITEM-EXT_PRODUCT_ID` output.

The profile-owned values win over request and mapping values. A request may select a

function, but it cannot change the profile's OCI version or external product-ID source. A normal mapping row also cannot replace `NEW_ITEM-EXT_PRODUCT_ID`.

For cXML, enter the sender domain, sender identity and shared secret. These values must match the cXML `Header > Sender > Credential` block.

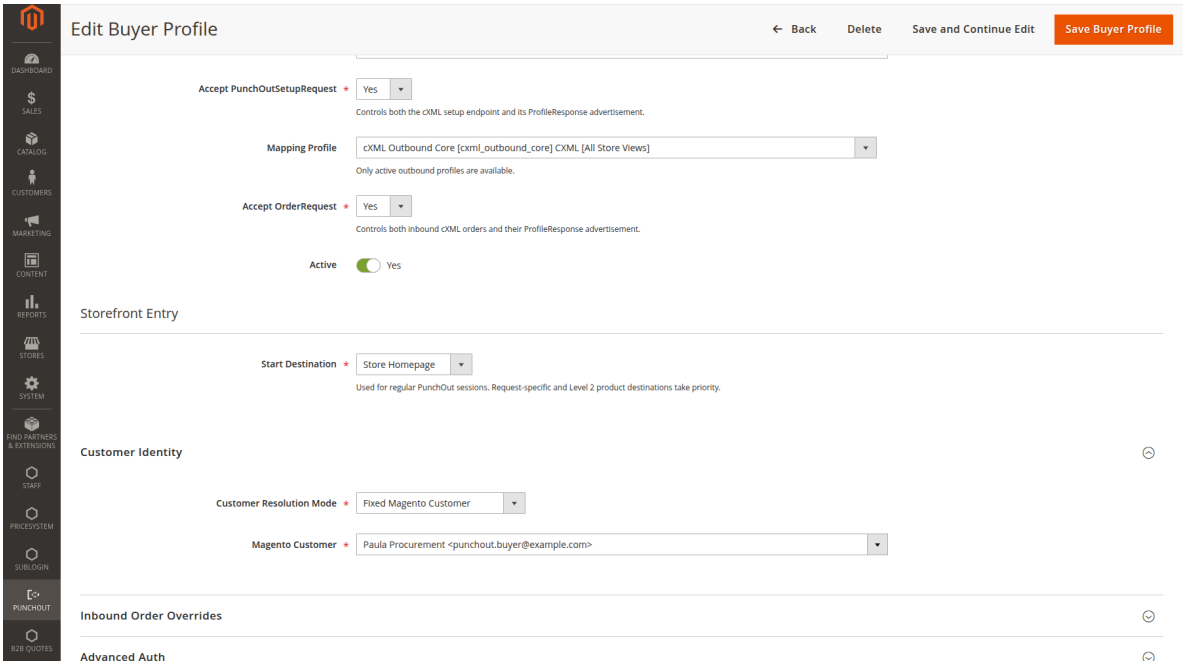
Stored secrets are encrypted. On an existing buyer, use the corresponding **Change** switch only when replacing a secret; leaving the password field blank preserves the saved value.

Storefront entry

The **Storefront Entry** fieldset controls where a regular PunchOut session opens after Magento activates the customer context:

Destination	Result
Shopping Cart	Opens the isolated PunchOut cart.
Store Homepage	Opens the selected Store View homepage.
CMS Page	Opens one CMS page available in the selected Store View.
Category	Opens one category from the selected Store View catalog.
Store-relative Path	Opens a validated internal path such as `procurement/catalog`.

The default is **Store Homepage**. A request-specific cXML Level-2 product target or OCI `DETAIL`/`DETAILADD` product has priority over this general destination. External URLs, another Store View and paths that cannot be resolved safely are rejected.



Use a buyer-specific destination when the procurement account should begin in a curated category or information page. Do not use it as a replacement for the buyer's cart-return URL.

Optional transport authentication

Protocol credentials are always checked. Advanced transport authentication can add one more independent check:

- HTTP Basic authorization;
- bearer token through `Authorization: Bearer` or `X-Punchout-Auth-Token`;
- shared secret through `X-Punchout-Shared-Secret`;
- none.

Do not put the OCI password in HTTP Basic unless that is also the intentionally configured transport credential. The module treats protocol authentication and transport authentication separately.

Assignment rules

A buyer can select only an active, outbound, non-template profile with the same protocol and a compatible store scope. If the profile is missing from the selector, check those four conditions before recreating it.

Deactivate a buyer to stop new sessions without deleting its mappings or trace history.

Inbound-order overrides

The second buyer fieldset can override Store-view defaults for approved cXML orders. Use it only for differences that belong to this procurement account, such as create-order mode, returned-cart pricing, tolerance, customer/address resolution, shipping, payment or update policy.

An empty override means **Use Store Configuration**. This keeps common behavior in one Store-view configuration while allowing a named buyer to follow a different approval contract. The effective values are stored with each inbound-order record for later diagnosis.

Effective configuration and precedence

The runtime resolves configuration in a fixed order:

1. The request identifies the protocol and buyer. The resolver first prefers a buyer profile scoped to the requested Store View and then considers an active `All Store Views` profile as the fallback.
2. The buyer profile selects its Store View. That scope supplies catalog, website, currency, tax and PunchOut configuration values.
3. For inbound-order settings, a non-empty buyer override wins. The empty **Use Store**

Configuration value does not mean disabled; it means that the Store-view setting is inherited.

4. An assigned mapping must be active, outbound, non-template, protocol-matching and store-compatible. The buyer assignment does not make an inbound template executable.

This precedence is also why a global mapping or buyer profile must not be used as an accidental cross-store default. If two profiles could match, test the resolved Store View in the simulator and confirm the resulting store, customer and mapping in **Punchout Trace** before enabling the buyer.

What is copied, merged and protected

`Start From Template` is a create-form action. It copies the template's mapping rows into the new runtime profile; it does not retain a live template relationship. Editing or synchronizing the reusable template later does not merge new rows into an existing runtime profile. Use **Duplicate** for a deliberate second runtime profile and compare the copied rows before activation.

At runtime, each row resolves its source, applies its formatter, then uses its fallback only when the formatted value is empty. A mandatory row stops the transfer when the resulting value is empty. Export position provides deterministic rule evaluation. Protocol-owned fields are applied after customer rules: for example, `NEW_ITEM-EXT_PRODUCT_ID` is derived from the buyer profile's OCI product-ID source and cannot be overwritten by an ordinary mapping row. This is an overwrite protection rule, not a merge of two arbitrary values.

Protocol targets are not interchangeable. OCI targets are indexed fields such as `NEW_ITEM-DESCRIPTION[1]` and `NEW_ITEM-QUANTITY[1]`; cXML targets are XML paths within the generated `ItemIn` structure. Select the protocol's target list when the row is created, and use a custom key only when the buyer has documented that key. The runtime validates special OCI targets such as unit of measure, lead time and manufacturer code. A value that does not meet the target's expected shape stops the transfer rather than producing a misleading field. Lead time and manufacturer code are not invented when Magento has no maintained source value.

For store scope, a profile assigned to the requested Store View wins over an active `All Store Views` profile with the same buyer identity. A mapping must still be active, outbound, non-template, protocol-matching and store-compatible. When a new mapping is created without an explicit scope, the form uses the current default store. Review that scope before saving; it is not a safe reason to make a profile global.

Confirmation, shipment notice and invoice documents have separate endpoint overrides. Never enter the browser cart-return URL in the ASN or Invoice URL fields.

Next, verify the Authentication and Security Model and run the Trace and Simulator workflow.

Authentication and Security Model

PunchOut does not accept a request merely because it contains a known buyer ID. Setup requests pass through store resolution, request security, protocol authentication and return-URL validation before a session context is created.

Validation sequence

1. Read the OCI parameters, JSON body or cXML body.
2. Resolve an active buyer by protocol, buyer identity and store scope.
3. Re-check that PunchOut is enabled in the resolved buyer's store.
4. Apply the IP allowlist and optional global HMAC signature.
5. Verify the buyer's OCI or cXML credentials.
6. Apply optional Basic, bearer or shared-secret-header transport authentication.
7. Validate the return URL.
8. Resolve the customer according to the buyer profile and create the expiring context token. E-mail mapping can provision a missing customer when JIT is enabled.

OCI identity

The buyer resolver uses `buyer_id` or `USER`. Authentication then compares the request username with the OCI username on the profile and checks the configured password. OCI protocol fields are independent of an optional HTTP `Authorization` header.

cXML identity

The request extractor reads sender identity, credential domain and shared secret from the cXML header. All three values must match the selected buyer profile. The sender identity also acts as the buyer ID used for profile resolution.

Magento customer identity

The buyer profile selects one customer-resolution mode:

- guest session;
- fixed Magento customer;
- customer e-mail from a configured OCI parameter or cXML field;
- optional request customer ID;
- required request customer ID.

Request customer IDs are accepted only when the corresponding security option is enabled. E-mail mapping first resolves an existing account in the buyer profile's website. If none exists and JIT provisioning is enabled, Magento creates the account in the configured customer group and applies Magento's normal account-confirmation policy.

IP allowlist

The allowlist accepts individual IPv4/IPv6 addresses and CIDR ranges. When the field is empty, every valid source IP is allowed. If Magento runs behind a proxy, verify the resolved client address before enforcing the list; otherwise a correct buyer may be rejected or the proxy address may be trusted too broadly.

Global request signature

Strict signature mode expects an HMAC-SHA256 digest in `X-Punchout-Signature` or `signature`. It is an additional SoftwareSilo check and must be implemented by the calling integration. Missing configuration, a malformed digest and a mismatched digest produce different nested security codes.

Return URL policy

The OCI hook URL or cXML `BrowserFormPost/URL` must:

- be an absolute HTTPS URL;
- contain a resolvable host;
- resolve only to public addresses in production;
- avoid `localhost` and private/reserved IP ranges outside sandbox mode.

Sandbox mode permits local or private targets for controlled testing, but it does not permit plain HTTP. The optional insecure-TLS setting affects certificate verification for outbound document delivery; it does not turn an HTTP URL into a valid one. Confirmation, ASN and invoice delivery use their own configured endpoints, not the browser return URL.

Context token

Successful setup creates a random 64-character token with the configured lifetime. The token binds buyer, store, customer, mapping, protocol, operation and cart context. Treat it as a credential: pass it only over HTTPS, never place it in documentation or tickets, and expire it through `/punchout/session/logout` when the flow is abandoned.

The token is intentionally short-lived and may be cleaned up after expiry. An accepted OrderRequest retains only the encrypted routing and identity data needed for later confirmation, shipment and invoice documents. Fulfilment does not require the original browser session.

Trace payloads are sanitized before storage, but access to Punchout Trace should still be restricted through the dedicated ACL permission.

Cart API and Session Lifecycle

After OCI login or cXML setup, the returned token can drive a browser shopping session, the cart API, or both. Every cart call is bound to the Magento quote stored on that PunchOut context.

Open the storefront session

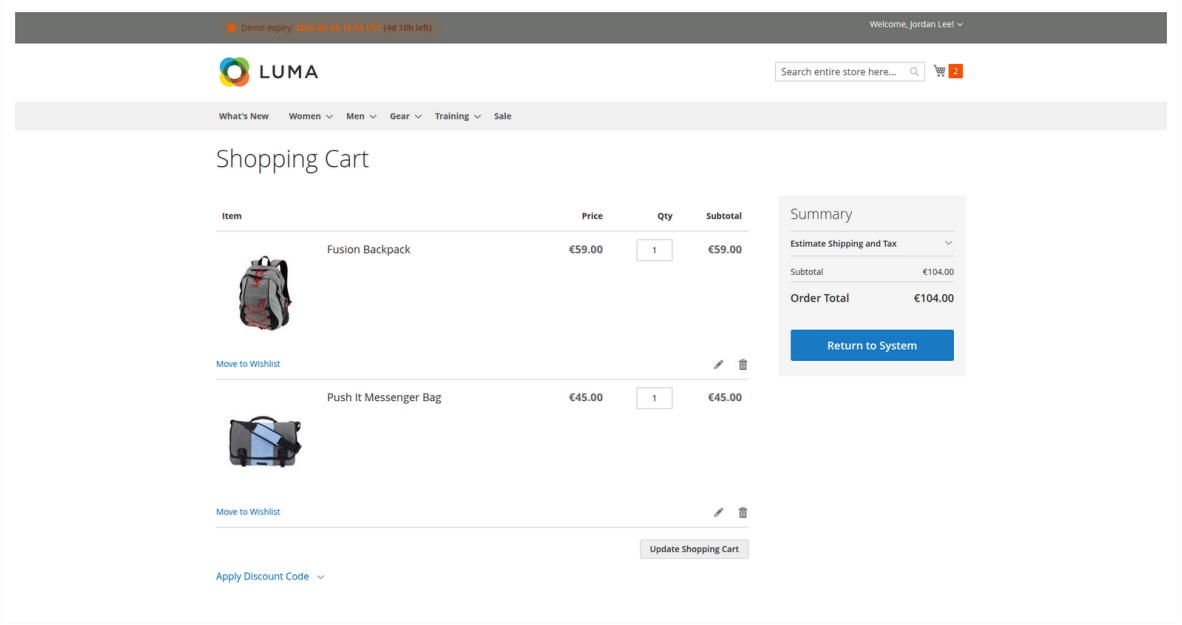
Send the browser to:

```
/punchout/session/activate?token=<token>&target_url=/catalogsearch/result/?q=chair
```

The activation route validates the token, stores it in the Magento session and applies the customer context resolved during setup. Depending on the buyer profile, this can be a guest, a fixed customer, an account resolved from the request, or a customer created through JIT provisioning. `target_url` may be relative or use the same storefront host; an external host is discarded.

PunchOut creates its own active quote. It never reuses the mapped customer's ordinary checkout quote. On every cart load, Magento checks that the quote is active, belongs to the selected Store view and has the same customer association as the context. A quote from another store or customer, or a quote that has already been submitted, is rejected. The context's currency and conversion rate are restored before totals are collected.

During the active session, the cart shows the configured return action instead of the normal checkout action.



Tokenized cart endpoints

Send the token in the request body, query parameters or `X-Punchout-Token` header.

Endpoint	Method	Required business input
`/punchout/cart/add`	POST	`sku`, positive `qty`; or an `items` array
`/punchout/cart/update`	POST	`item_id` or `sku`, plus fields to change; or an `items` array
`/punchout/cart/remove`	POST	`item_id`, `sku`, `item_ids`, `skus` or `items`
`/punchout/cart/get`	GET or POST	token only

Example add request:

```
{
  "token": "<session-token>",
  "items": [
    {"sku": "24-WG085", "qty": 4}
  ]
}
```

Magento remains the pricing authority by default. A `price` value is applied only when **Allow inbound cart item prices** is enabled for the context store. Accepted custom prices remain locked during later total recollection.

A successful `/punchout/cart/get` response uses HTTP 200, sets `success` to `true`, and returns the current quote summary under `cart`. Keep the token and request signature out of screenshots, exported traces and support tickets. A normal storefront cart image does not prove that the tokenized API contract was called.

Return the cart

The storefront action `/punchout/session/back` attaches the active Magento quote to the context and prepares the mapped payload.

- OCI returns an HTML form that posts indexed `NEW_ITEM-*` fields to the hook URL.
- cXML returns an HTML form that posts the generated `PunchOutOrderMessage` as `cXML-urlencoded`.

Direct integrations can call `/punchout/oci/transfer` or `/punchout/cxml/transfer` and receive the generated HTML or XML payload inside a JSON response instead.

Status and expiry

A new context starts in `AUTHENTICATED`. A successful direct transfer changes it to `TRANSFERRED`. Logout changes it to `EXPIRED`; time expiry also prevents further browser or cart API use. Expired contexts are removed by cron every 15 minutes according to the cleanup rules. A transferred context cannot be used again, even when its token has not reached its time limit.

The cart API follows the same operation policy as the storefront. An `inspect` context may be read but cannot update or remove lines. `DETAIL` does not initialise a normal shopping quote. `DETAILADD` accepts only the resolved product and closes the one-item handoff after Add to Cart.

The configured lifetime must cover browsing and cart return. A later, independently authenticated cXML `OrderRequest` can still use its retained BuyerCookie snapshot after browser-session expiry, provided the snapshot has not expired. Configure snapshot retention separately for the approval window.

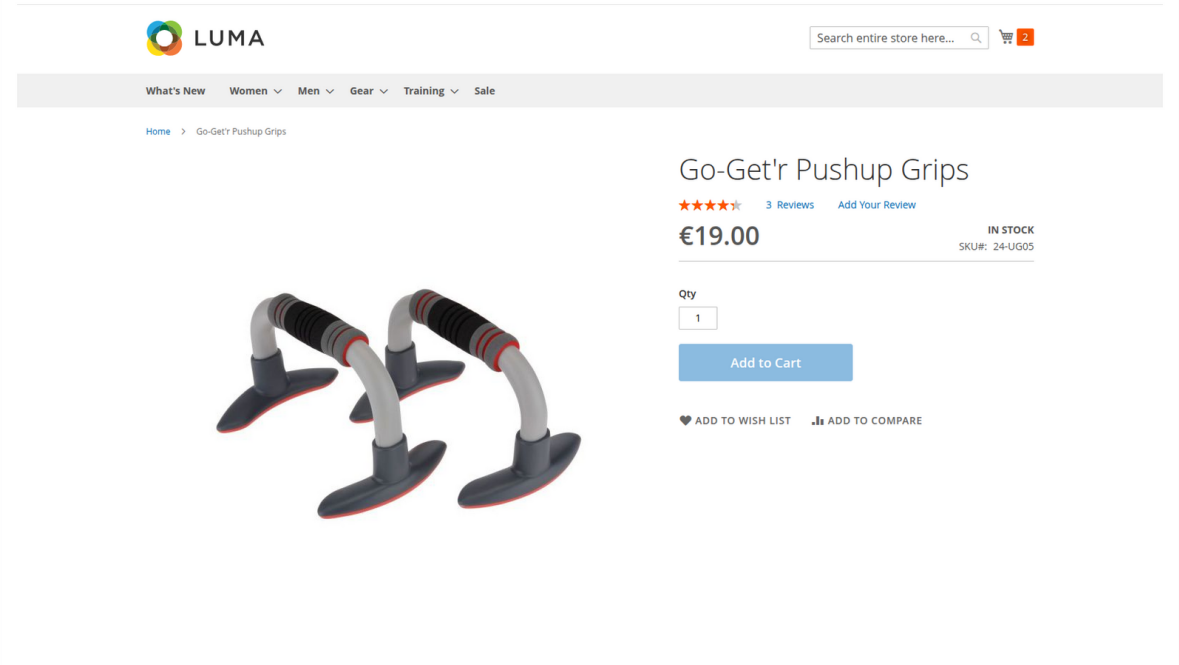
Storefront Operation Policy

cXML setup and OCI requests can describe the buyer's intent as `create`, `edit`, `inspect`, `detail` or `detail_add`. The frontend operation policy uses that value to decide which Magento full-action names may run while the PunchOut token is active.

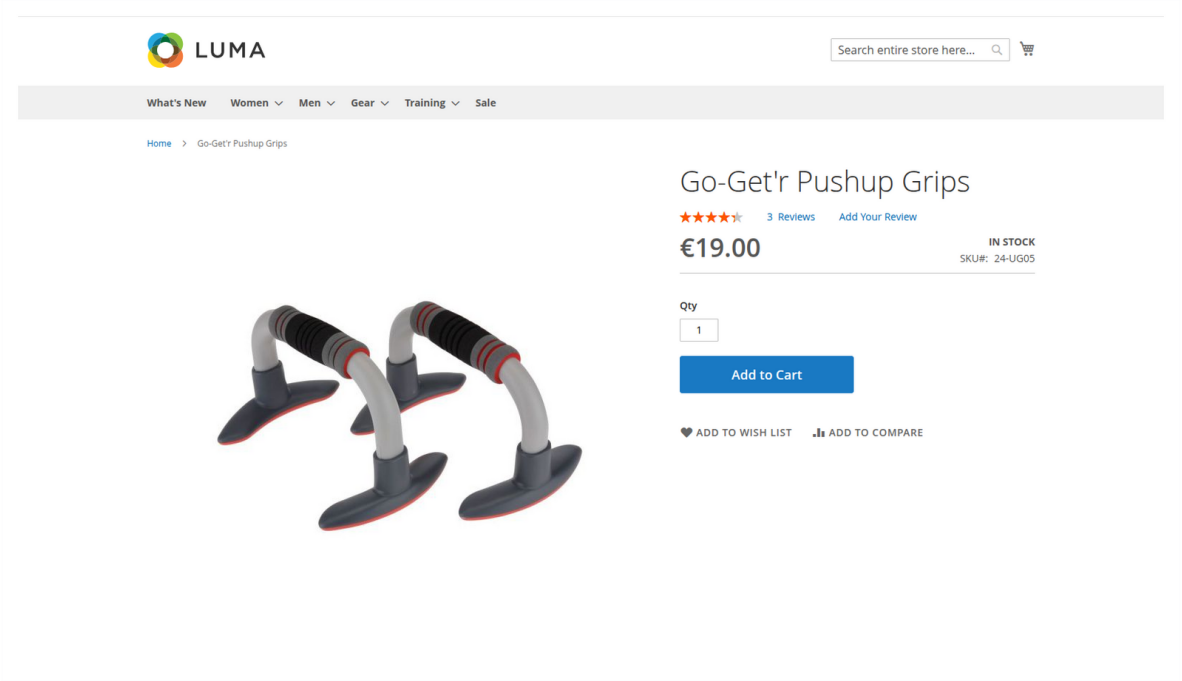
Supplied behavior

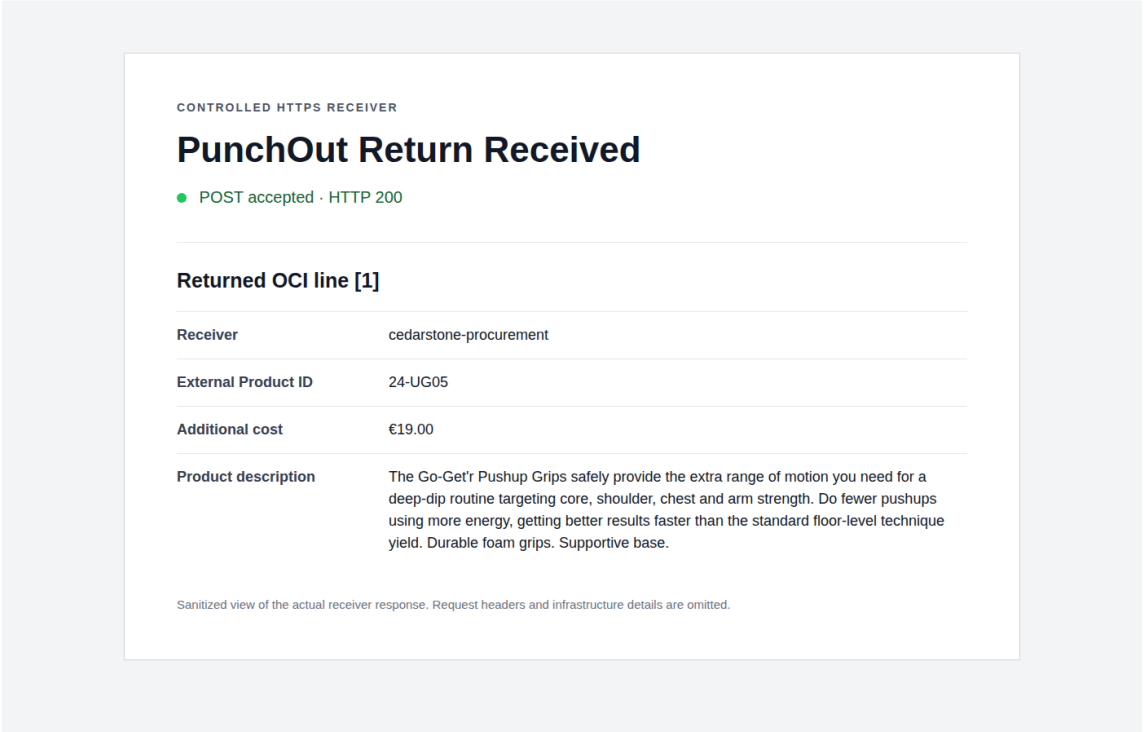
- **create** permits every action by default (`*`), so the buyer can browse and build a new cart.
- **edit** permits normal catalogue and cart editing by default. Magento reconstructs an isolated quote from the supplied `ItemOut` lines and recalculates them in the mapped customer context.
- **inspect** reconstructs the supplied cart but presents it as read-only. Quantity, edit, remove, wishlist, coupon, shipping-estimate, cross-sell and checkout controls are not rendered. The cart and minicart retain viewing and Return to System actions.
- **detail** opens the product resolved from OCI `PRODUCTID` without starting a normal cart-building flow.
- **detail_add** opens the resolved product, permits one successful Add to Cart action and immediately returns that single line to the procurement system.

In a DETAIL session the exact product remains visible for inspection, while the normal Add to Cart action is unavailable.

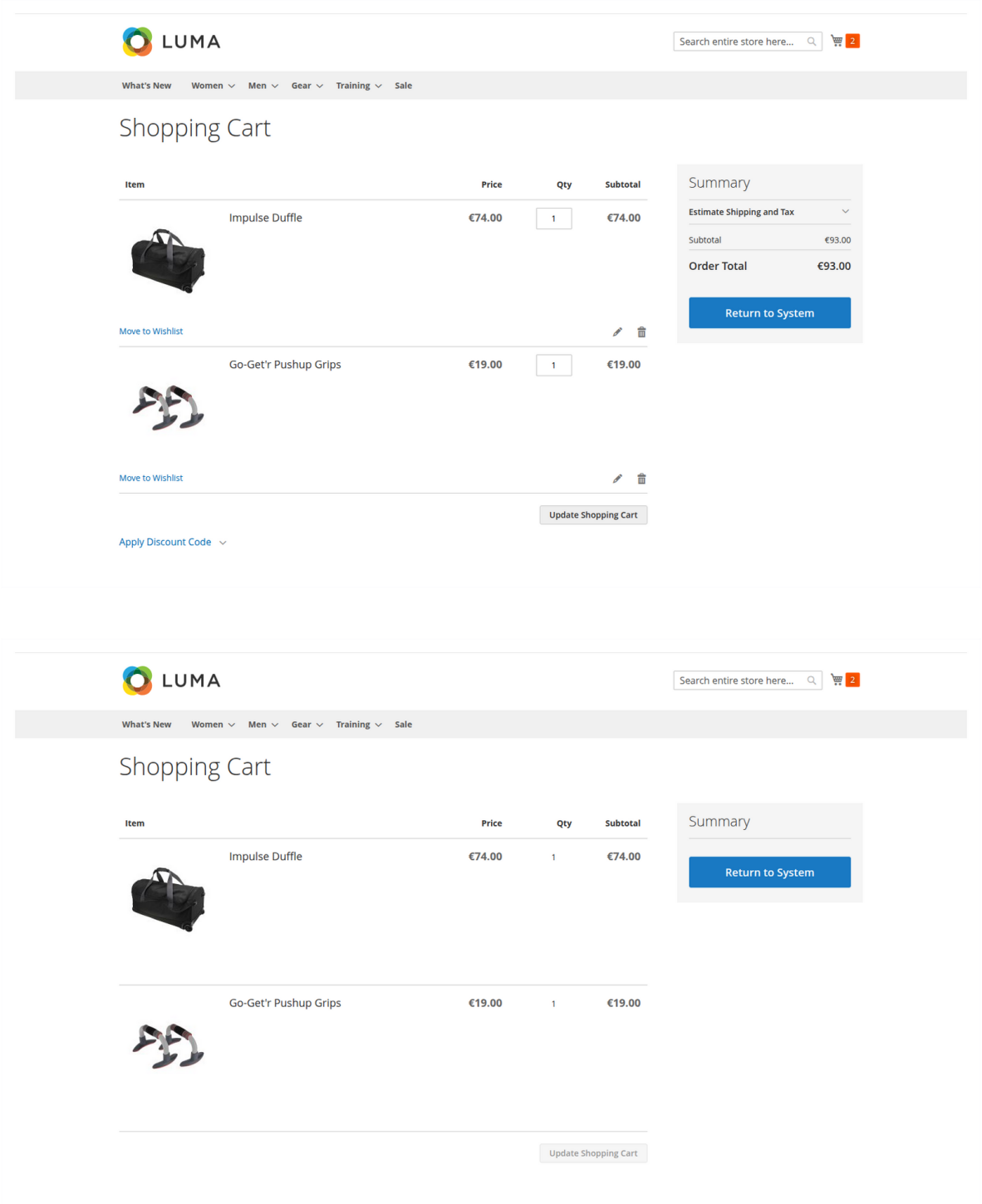


DETAILADD enables that action only for the resolved product and completes the return after the first successful add.





The difference is visible in the storefront: EDIT retains quantity and item controls, while INSPECT displays the same reconstructed lines without mutation or checkout actions.



When a route is not allowed, Magento redirects to the configured landing route, which defaults to `checkout/cart`.

Change a policy safely

Open **Punchout > Configuration > Operation Policy**.

1. Capture the exact Magento full-action name, such as `catalog_product_view` or `checkout_cart_index`.
2. Add it only to the operations that need it.

3. Separate entries with semicolons, commas or new lines.
4. Test direct navigation, product search, cart changes and Return to System for every enabled operation.

Do not enter URL paths such as `/catalog/product/view`. The guard compares normalized full-action names. An incorrect list can redirect a buyer away from a required page without changing the underlying role or customer permission.

The policy is a storefront-flow control. It does not replace buyer authentication or Magento catalog permissions. Integrations can extend actor and permission resolution through the module's public service contracts and events.

cXML Profile Discovery

Profile Discovery lets a procurement platform ask Magento which cXML transactions are available and where to send them. The response is generated from the active Store View and Buyer Profile, so it cannot advertise a service that the selected buyer is not allowed to use.

Configure discovery

Open **Punchout > Configuration > cXML Profile Discovery** for the required Store View.

- **Enable cXML Profile Discovery** controls the endpoint for that scope.
- **Public PunchOut Base URL** is needed only when the public URL used by the procurement platform differs from the Magento Store View base URL. Enter the scheme and host, without a query string or fragment.

Then open **Punchout > Punchout Buyers** and edit the cXML buyer:

- **Accept PunchOutSetupRequest** controls the setup transaction.
- **Accept OrderRequest** controls inbound purchase orders.

These switches affect both the actual endpoints and the transactions listed in `ProfileResponse`.

Send a ProfileRequest

Post the cXML document to:

```
POST /punchout/cxml/profile
Content-Type: application/xml
```

The request must contain a complete cXML Header. Magento resolves the Buyer Profile from the Sender credential domain and identity, then checks the shared secret, Store-scoped security settings and optional transport authentication.

Profile transactions require cXML 1.1 or 1.2. A cXML 1.0 request receives an error response because that version does not support Profile Discovery.

Read the response

A successful `ProfileResponse` lists only the buyer's enabled transactions:

- `PunchOutSetupRequest` points to the Store-specific setup endpoint.
- `OrderRequest` appears only when inbound orders are enabled for the Buyer Profile.
- the OrderRequest `changes` option reflects the effective update and cancellation policies.

Magento adds the selected Store code to generated endpoint URLs. The response uses the cXML version from the request and does not rely on a manually maintained transaction list.

If discovery fails, verify the Store View, Buyer Profile credentials, source IP restrictions and Profile Discovery setting before changing the public URL.

Security and Idempotency

Use buyer authentication, network restrictions and request signing to decide who may call PunchOut. Use idempotency to make a permitted retry safe. They solve different problems and should not be treated as substitutes.

Security layers

- buyer identity and protocol credentials select and authenticate the procurement account;
- optional Basic, bearer or shared-secret-header authentication protects the HTTP transport;
- the IP allowlist limits inbound network sources;
- optional HMAC-SHA256 signatures detect a missing or altered signed request;
- return-URL validation blocks non-HTTPS and production-local targets;
- context tokens bind browser activation, cart access and cart transfer to one accepted setup;
- Magento pricing remains authoritative unless inbound prices are explicitly enabled.

See Authentication and Security Model for the exact order and request fields.

Request replay keys

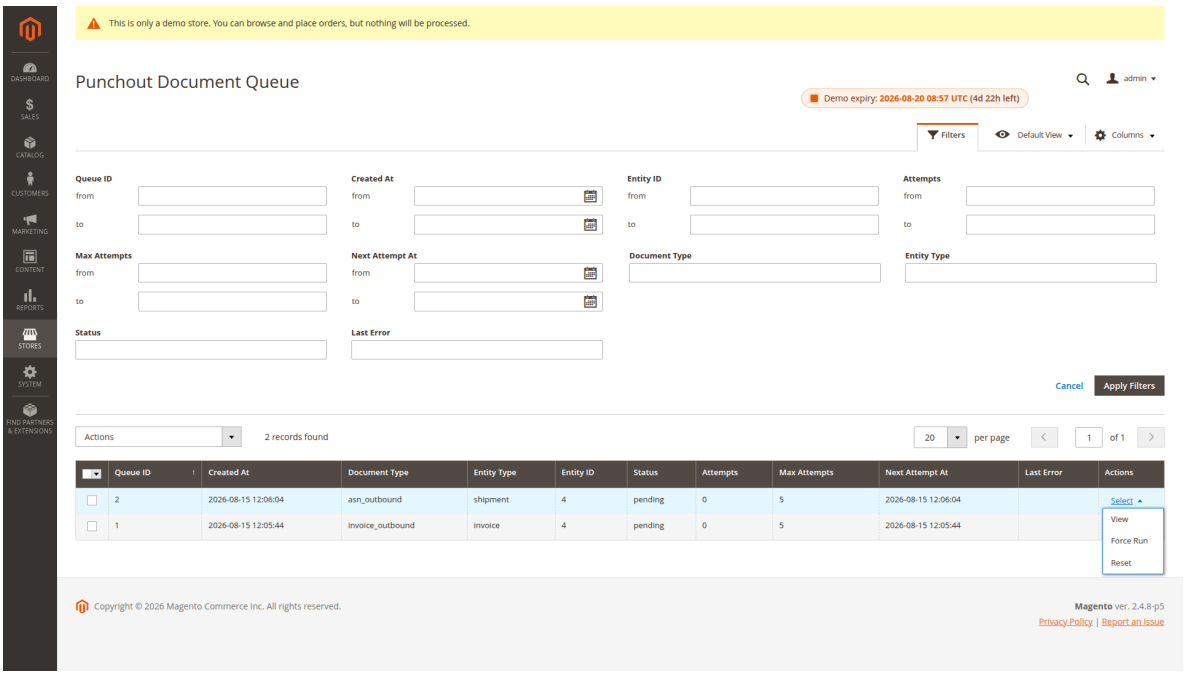
Transfer and document endpoints accept a client-generated key only in the `X-Punchout-Request-Key` header.

The first request reserves the key for that operation. Later calls behave as follows:

- same operation, key and request hash after completion: the saved response and its original HTTP status are replayed;
- same operation and key with a different request hash: HTTP 409 with `PUNCHOUT_REQUEST_REPLAY_CONFLICT`;
- same operation and key while the first call is still running: HTTP 409 with `PUNCHOUT_REQUEST_REPLAY_IN_PROGRESS`.

Generate one key for one semantic action. Reuse it only when retrying the identical request. A buyer must not recycle a key for a changed cart, another shipment or another invoice.

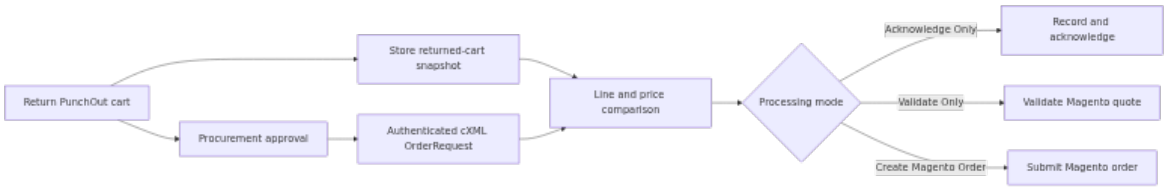
Replay records are isolated by Store view, operation and request key. They are retained according to **Punchout > Configuration > Retention > Request Replay Retention** and removed in bounded batches by the cleanup cron.



Later shipment and invoice documents do not depend on the browser token remaining active. Their encrypted payload and routing information are retained with the order workflow so fulfilment can happen days or weeks after the shopping session.

Inbound Orders and Pricing

Cart return and order creation are separate steps. **Return to System** sends the selected products back to the procurement platform. After its own approval process, that platform can submit a cXML `OrderRequest` to Magento.



Customer Identity

Cart

Inbound Orders

Processing Mode [store view]
Acknowledge Only
Acknowledge Only records the request without creating a quote or order.

Authoritative Price Source [store view]
Current Magento Pricing
Current Magento Pricing executes the normal customer-specific catalog and quote pricing pipeline.

Returned Cart Line Matching [store view]
Auxiliary ID, then SKU
Used by snapshot-based pricing. Auxiliary ID is preferred when available.

Changed Quantity Action [store view]
Recalculate with Magento

New Item Action [store view]
Recalculate with Magento

Removed Item Action [store view]
Allow Removal

Missing/Expired Snapshot Action [store view]
Reject
Applies when snapshot-based pricing cannot resolve the original returned cart.

Price Mismatch Action [store view]
Reject Order

Absolute Unit Price Tolerance [store view]
0.01
Difference in order currency allowed per unit.

Percent Unit Price Tolerance [store view]
0
Percentage difference allowed per unit. Either configured tolerance can permit the price.

Customer Resolution [store view]
Buyer Profile Customer

Product Resolution [store view]
Supplier Part ID

Address Source [store view]
Incoming Order Addresses

Shipping Method [store view]
[flatrate] Fixed
The selected method must be available for the resolved quote and destination.

Payment Method [store view]
PunchOut Purchase Order
Defaults to the included offline method punchout_purchaseorder.

Order Update Policy [store view]
Manual Review

Order Cancellation Policy [store view]
Manual Review

Send Customer Order Email [store view]
No

Send cXML ConfirmationRequest [store view]
No
Sends a confirmation after Magento creates or cancels an order.

ASN URL [store view]
cXML endpoint that receives ShipNoticeRequest documents. This is not the PunchOut cart return URL.

Invoice URL [store view]
cXML endpoint that receives InvoiceDetailRequest documents. This is not the PunchOut cart return URL.

Supplier Credential Domain [store view]
cXML credential domain used for outbound ConfirmationRequest, InvoiceDetailRequest and ShipNoticeRequest messages.

Supplier Identity [store view]

Supplier Shared Secret [store view]

Maximum Processing Attempts [store view]
5

Retry Batch Size [global]
20

Retry Delay (minutes) [store view]
15

Outbound cXML Response Mode [store view]
Allow Empty HTTP 2xx Response
Controls whether a successful HTTP response must include a valid cXML Response Status.

Inbound Order Payload Retention (days) [store view]
365
Removes retained request details after the configured period. Order identity and idempotency evidence are retained.

Returned Cart Snapshot Retention (days) [store view]
90
Must cover the longest expected requisition and approval workflow.

Choose the processing mode

Configure the default under **Punchout > Configuration > Inbound Orders**. A buyer profile can override the Store-view value.

Mode	What Magento does	When to use it
Acknowledge Only	Authenticates and records the request, then returns an acknowledgement. It does not build a quote or order.	The external system owns fulfilment or another integration consumes the record.
Validate Only	Resolves the customer, products, addresses, shipping, payment and totals without submitting the quote.	Acceptance testing or an approval stage that needs Magento validation.
Create Magento Order	Validates the request and places a Magento sales order.	The approved procurement PO should become the fulfilment order in Magento.

The inbound-order grid records the effective mode with the request. This matters when one buyer overrides a Store-view default.

Preserve the returned cart references

The procurement system chooses the opaque BuyerCookie during `PunchOutSetupRequest`. Magento returns the same value in `PunchOutOrderMessage`. The immutable returned cart is stored against Magento's internal PunchOut session; the BuyerCookie may be shared by independent sessions.

For later original-cart pricing and product options, the approved `OrderRequest` must return Magento's `SupplierPartAuxiliaryID` in the corresponding `ItemOut/ItemID`:

```
<SupplierPartAuxiliaryID>exact-value-from-the-returned-cart</SupplierPartAuxiliaryID>
```

The snapshot is bound to the original buyer profile, Magento customer, Store view and currency. A BuyerCookie alone cannot select a snapshot, even when only one matching retained cart remains. If a BuyerCookie is supplied in the PO, it must agree with the cart selected by the auxiliary identifiers. Missing or unmatched cart references follow the configured missing-snapshot policy without attaching a different cart. Set **Returned Cart Snapshot Retention** longer than the customer's longest approval cycle.

Select the price authority

Price source	Behavior
Current Magento Pricing	Magento prices every submitted line through the resolved customer quote.
Returned Quote Snapshot	Every submitted line must match the returned cart. Magento uses the captured returned price.
Incoming Order Prices	Magento accepts submitted unit prices subject to the configured mismatch handling. Use this only when the buyer is contractually authoritative.
Original Price if Unchanged; Otherwise Current Magento Price	Unchanged lines keep their returned Magento price. Changed quantities and new products use current Magento pricing.

The hybrid option is the normal choice when approval may take days or weeks. It preserves the price Magento originally returned without trusting changed order content.

The four sources are not a merge of arbitrary values. The selected source defines the authoritative unit price for each line; the mismatch action then decides what to do when the submitted price differs. The hybrid source is the only source that branches on line identity: an unchanged SKU/quantity/configuration can use the immutable returned snapshot, while a changed quantity or new line goes through the current Magento quote calculation.

Compare the approved lines

Magento first identifies a retained cart through an exact returned auxiliary identifier. Within that identified cart, **Auxiliary ID then SKU** matches lines by auxiliary ID, then by an echoed `lineNumber` and finally by a unique SKU. Line-number matching requires the submitted SKU to agree. **SKU Only** changes line matching within the selected cart; it does not allow a BuyerCookie to identify the cart. A supplied but unknown auxiliary identifier cannot borrow another line's original data through either strategy. Snapshot-only pricing rejects that line; hybrid pricing applies the configured new-item policy. Lines without an auxiliary identifier can still use the configured line-number/SKU matching after the cart has been identified.

The remaining policies answer four separate questions:

- What happens when an approved quantity differs from the returned cart?
- May the buyer add a product after cart return?
- May the approved order omit a returned line?
- What happens when the snapshot is missing or expired?

Each case can recalculate, allow, reject or enter manual review where that choice is applicable. Do not select a permissive fallback until the procurement team has supplied a real example of that change.

The policy is evaluated in this order:

1. Magento authenticates the sender and resolves the effective buyer and Store view configuration.
2. Magento selects the immutable snapshot by an exact auxiliary identifier inside the authenticated buyer/customer/Store/currency scope, then applies the configured line-matching strategy within that cart. Conflicting references to multiple retained carts are rejected.
3. The configured actions handle a changed quantity, a new item and a removed returned item independently.
4. A missing or expired snapshot follows **Use Current Magento Price**, **Manual Review** or **Reject**. Manual approval changes only the review gate; it does not bypass product, address, shipping, payment or total validation.
5. Magento applies the selected price source and mismatch tolerance, then runs the processing mode.

For **Returned Quote Snapshot**, a new item or an unmatched line is rejected because every submitted line must exist in the snapshot. For the hybrid policy, unchanged lines use the captured value and changed or new lines use current Magento pricing unless the configured action sends the request to review or rejects it. For **Current Magento Pricing** and **Incoming Order Prices**, snapshot matching is still available for line metadata, but it does not make the snapshot price authoritative.

Validate prices and totals

Magento compares the submitted unit price with the selected authoritative price. A difference is accepted when either the absolute or percentage tolerance allows it. The mismatch action can reject the order, send it to manual review, log and continue, or ignore the difference.

Quote creation still uses Magento's customer, product, address, shipping, payment, tax and inventory rules. Price tolerance does not bypass those checks.

The session `tax_mode` decides which completed Magento quote value is sent back. `NET` uses Magento's calculation price. `GROSS` uses Magento's price including tax. Magento still calculates taxes and totals from the Store view, customer, addresses, tax classes and installed pricing or tax extensions.

Preserve product options and line data

The returned-cart snapshot stores Magento's safe buy-request data for configurable products, custom options, bundles and downloadable links. Every returned line receives an opaque `SupplierPartAuxiliaryID`, so Magento can distinguish equal SKUs with different configurations when the approved OrderRequest arrives.

The OrderRequest quantity remains authoritative. Every `ItemOut` must carry a unique `lineNumber`, which Magento stores together with `UnitOfMeasure` on the quote and

order item. Confirmation, shipment and invoice documents reuse those values. Configurable and bundle children are reconciled through their visible parent line; Magento rejects inconsistent partial quantities or invoice amounts instead of sending misleading document data.

Resolve the Magento context

- **Customer Resolution** follows the buyer profile: guest, fixed Magento customer, e-mail mapping, or an optional/required request customer ID. E-mail mapping can provision a missing account when JIT is enabled.
- **Product Resolution** treats `SupplierPartID` or `SupplierPartAuxiliaryID` as the Magento product reference.
- **Address Source** uses incoming `ShipTo` and `BillTo` values or the customer's default addresses.
- **Shipping Method** must be available for the resolved quote and destination.
- **Payment Method** defaults to the included **PunchOut Purchase Order** method.

When **Create Magento Order** is active, empty or unavailable shipping and payment choices stop order creation.

Updates, cancellations and manual review

`OrderRequestHeader@type` selects `new`, `update` or `delete`.

- Updates can enter manual review, be rejected, or cancel and recreate the Magento order.
- Cancellations can enter manual review, be rejected, or cancel the linked Magento order.
- A request held for review appears in **Punchout > Inbound Orders**. Authorized administrators can inspect the normalized payload and then process or reject it.

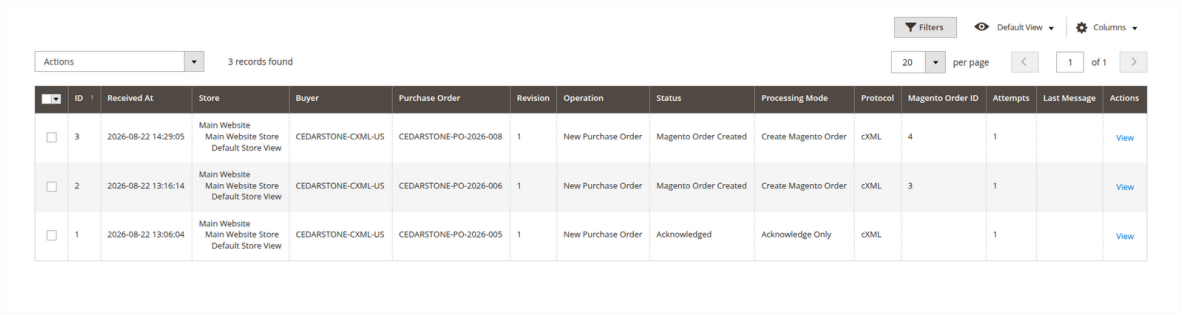
Use a new `orderVersion` for a genuine revision. An exact network retry keeps the same PO identity, request body and `X-Punchout-Request-Key`; Magento returns the existing result instead of creating a duplicate.

An update is a business revision, not a transport retry. The supported automatic update policy is **Cancel and Recreate**; it uses a recoverable replacement-order workflow so an interrupted update does not silently lose the original order or create an untracked duplicate. **Manual Review** and **Reject** keep the existing order untouched. A delete follows its separate cancellation policy and can be manually reviewed, rejected or cancel the linked Magento order.

The replacement workflow records the original order, the replacement attempt and the resulting inbound-order state. If a worker stops between cancellation and replacement, the record remains available for an authorized retry instead of starting an unrelated order. A transport retry of the same request still uses the original idempotency key and replays the stored result; a new order revision needs a new `orderVersion` and a new

semantic request key.

Find the resulting records

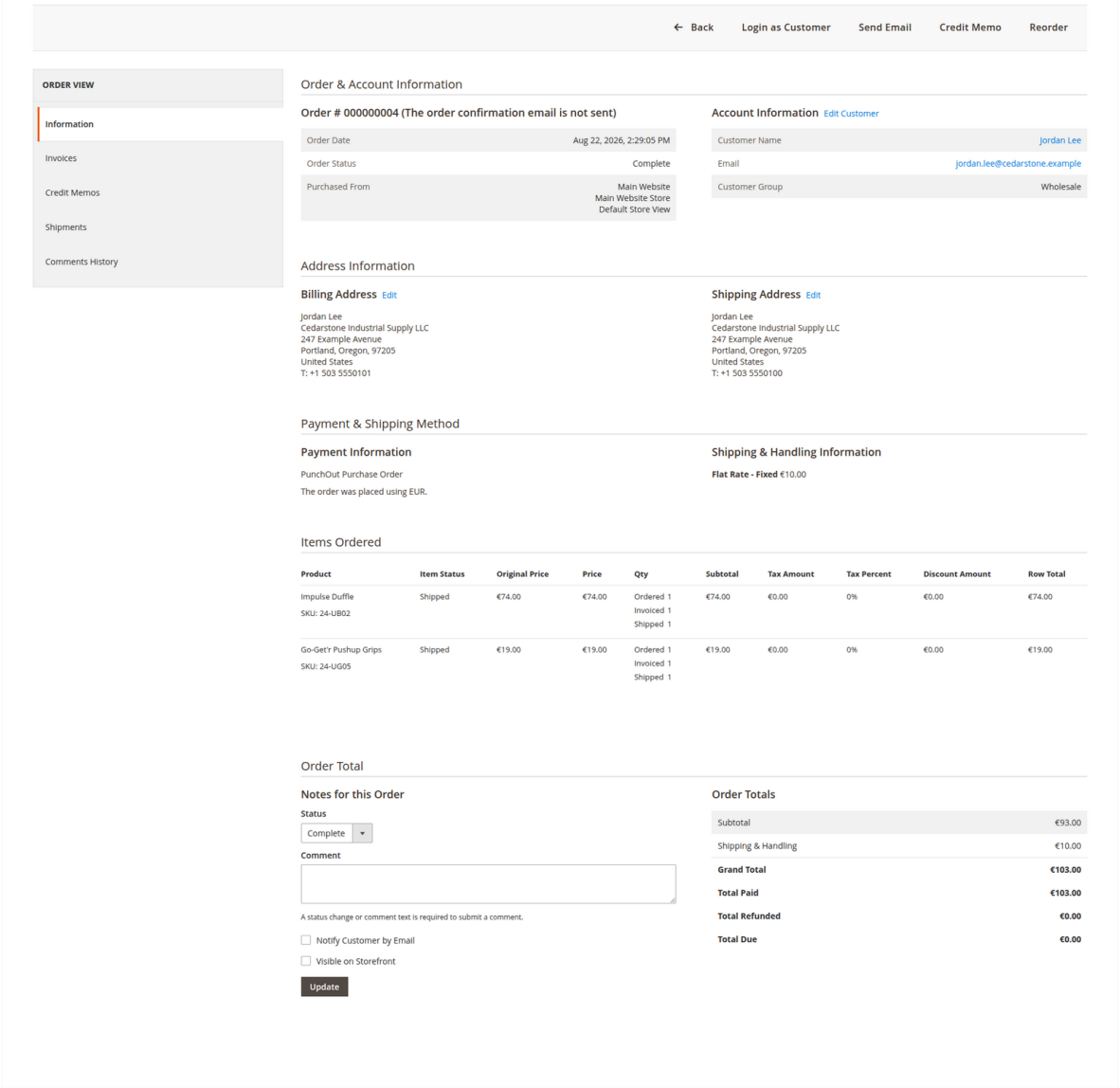


	ID	Received At	Store	Buyer	Purchase Order	Revision	Operation	Status	Processing Mode	Protocol	Magento Order ID	Attempts	Last Message	Actions
☐	3	2026-08-22 14:29:05	Main Website Main Website Store Default Store View	CEDARSTONE-CXML-US	CEDARSTONE-PO-2026-008	1	New Purchase Order	Magento Order Created	Create Magento Order	cXML	4	1		View
☐	2	2026-08-22 13:16:14	Main Website Main Website Store Default Store View	CEDARSTONE-CXML-US	CEDARSTONE-PO-2026-006	1	New Purchase Order	Magento Order Created	Create Magento Order	cXML	3	1		View
☐	1	2026-08-22 13:06:04	Main Website Main Website Store Default Store View	CEDARSTONE-CXML-US	CEDARSTONE-PO-2026-005	1	New Purchase Order	Acknowledged	Acknowledge Only	cXML		1		View

The inbound-order detail preserves the buyer, external PO, revision, operation, Store view, customer, working quote, original PunchOut quote, context, cXML payload ID and Magento order ID. The technical payload section contains sanitized copies of the original request, normalized order and processing response.



After order creation, open the linked record under **Sales > Orders** to verify the customer PO number, customer, addresses, PunchOut payment method, products and totals.


Back
Login as Customer
Send Email
Credit Memo
Reorder
ORDER VIEW
Information
Invoices
Credit Memos
Shipments
Comments History
Order & Account Information
Order # 000000004 (The order confirmation email is not sent)
Order Date
Aug 22, 2026, 2:29:05 PM
Order Status
Complete
Purchased From
Main Website
Main Website Store
Default Store View
Account Information
Edit Customer
Customer Name
Jordan Lee
Email
jordan.lee@cedarstone.example
Customer Group
Wholesale
Address Information
Billing Address
Edit
Jordan Lee
Cedarstone Industrial Supply LLC
247 Example Avenue
Portland, Oregon, 97205
United States
T: +1 503 5550101
Shipping Address
Edit
Jordan Lee
Cedarstone Industrial Supply LLC
247 Example Avenue
Portland, Oregon, 97205
United States
T: +1 503 5550100
Payment & Shipping Method
Payment Information
PunchOut Purchase Order
The order was placed using EUR.
Shipping & Handling Information
Flat Rate - Fixed €10.00
Items Ordered
Product
Item Status
Original Price
Price
Qty
Subtotal
Tax Amount
Tax Percent
Discount Amount
Row Total
Impulse Duffle
SKU: 24-UB02
Shipped
€74.00
€74.00
Ordered 1
Invoiced 1
Shipped 1
€74.00
€0.00
0%
€0.00
€74.00
Go-Get'r Pushup Grips
SKU: 24-UG05
Shipped
€19.00
€19.00
Ordered 1
Invoiced 1
Shipped 1
€19.00
€0.00
0%
€0.00
€19.00
Order Total
Notes for this Order
Status
Complete
Comment
A status change or comment text is required to submit a comment.
Notify Customer by Email
Visible on Storefront
Update
Order Totals
Subtotal
€93.00
Shipping & Handling
€10.00
Grand Total
€103.00
Total Paid
€103.00
Total Refunded
€0.00
Total Due
€0.00

Document Loop and Queue

The document loop starts with an approved cXML `OrderRequest`. Magento can record, validate or create an order from that request. When fulfilment begins, Magento can send a `ConfirmationRequest`, `ShipNoticeRequest` and `InvoiceDetailRequest` to the procurement system.

Order creation and returned-cart pricing are covered in Inbound Orders and Pricing. This page focuses on outbound documents and their queue.

Configure outbound cXML identity

Generated cXML uses the effective supplier credential domain, identity and shared secret retained for the inbound order. Configure those values before enabling ConfirmationRequest delivery or testing shipment and invoice callbacks. Buyer-profile settings override Store-view defaults.

Configure a separate endpoint for each document type: Confirmation URL, ASN URL and Invoice URL. The browser cart-return URL is used only to return the shopping cart and is never reused for later server-to-server documents.

Magento generates cXML 1.2.070 envelopes for confirmation, shipment and invoice documents. The destination must be a validated HTTPS URL.

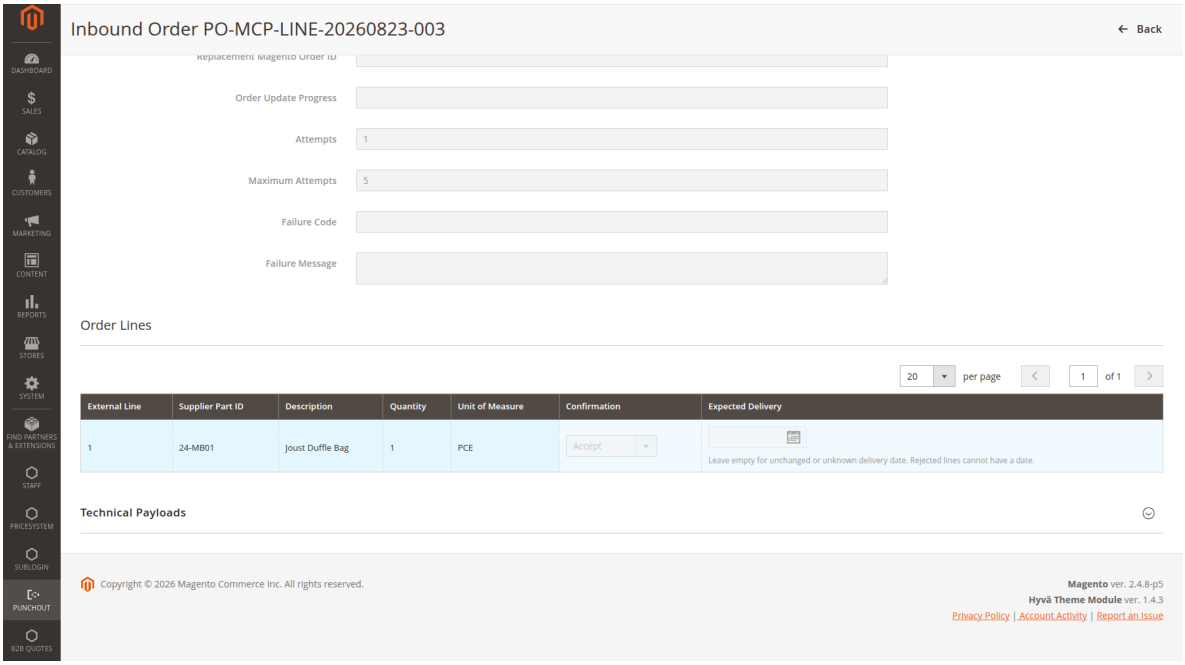
ConfirmationRequest

When **Send cXML ConfirmationRequest** is enabled, the effective Confirmation Mode decides how the document is prepared.

- **Automatic Whole-Order Confirmation** confirms a successfully created order without waiting for an administrator. A cancellation produces the corresponding rejection.
- **Manual Line Confirmation** waits for an administrator to decide every order line in **Punchout > Inbound Orders**.

For a manual confirmation, open the inbound order and select **Accept**, **Reject** or **Backordered** for every line. Accepted and backordered lines may include an expected delivery date. Rejected lines cannot have a delivery date. Every inbound line must have exactly one decision before the confirmation can be submitted.

Submitting the form creates one immutable confirmation document in the queue. The same inbound order cannot create a second confirmation. Its expected-delivery values are shown in the Admin user's locale, while the cXML document uses the protocol's ISO date format.



A delivery failure remains visible with the inbound order and queue entry so an administrator can diagnose the receiver before retrying.

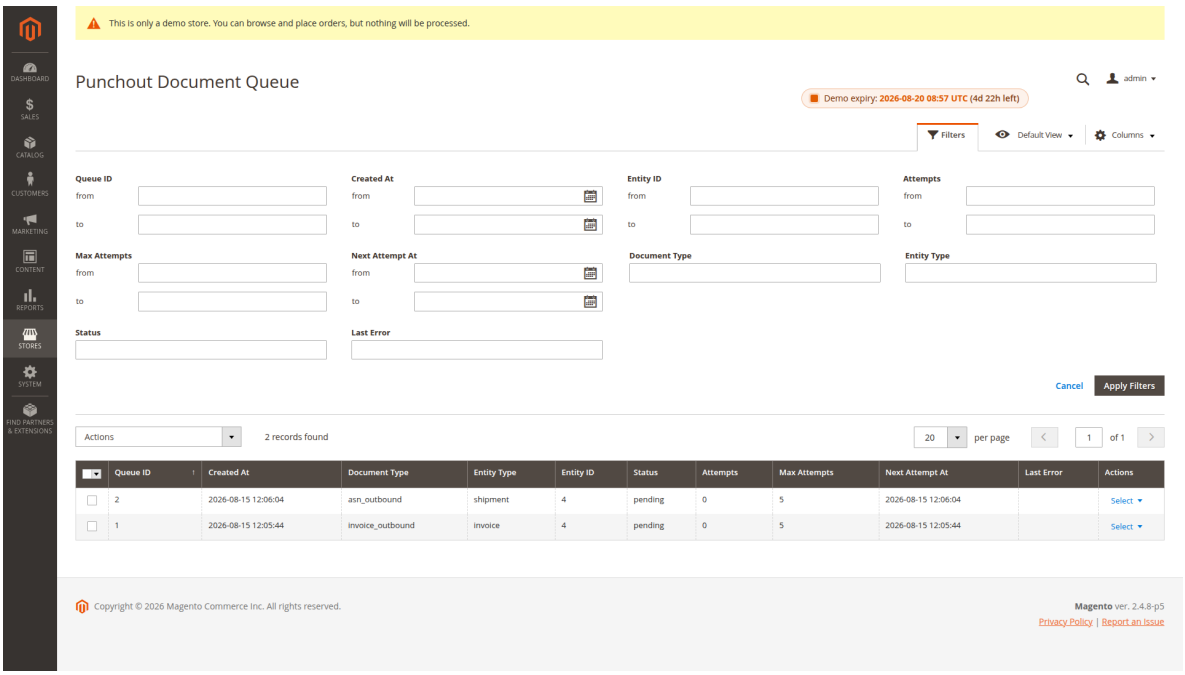
Automatic outbound documents

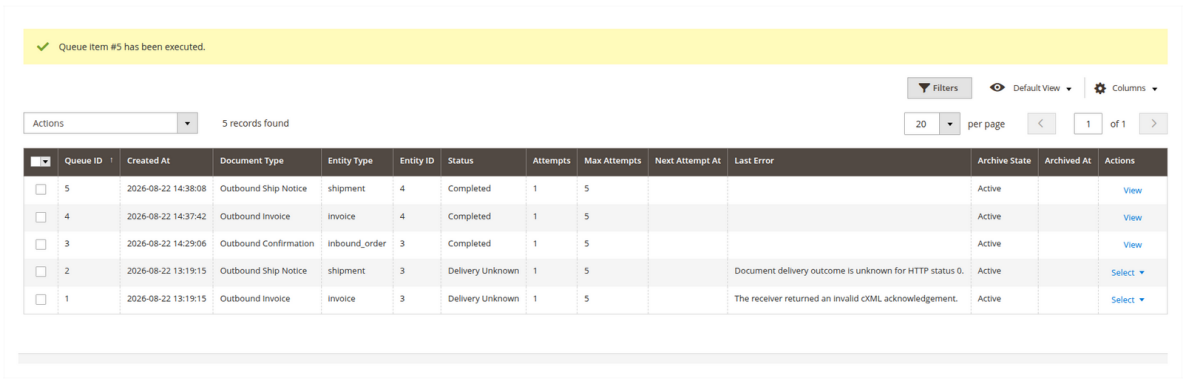
When Magento saves a shipment or invoice, the module checks whether its order quote is linked to a PunchOut context.

- A shipment creates one `asn_outbound` queue row containing the shipment number, tracking information and shipped SKU quantities.
- An invoice creates one `invoice_outbound` queue row containing the invoice number, order references, item details, summary amounts and order currency.
- The same Magento entity is not queued twice for the same document type.

The queue builds the outbound document once, encrypts the exact payload at rest and sends it to the document-specific HTTPS endpoint. Retries reuse the same retained payload and routing data. Shipment and invoice delivery therefore remains independent of the short-lived browser context.

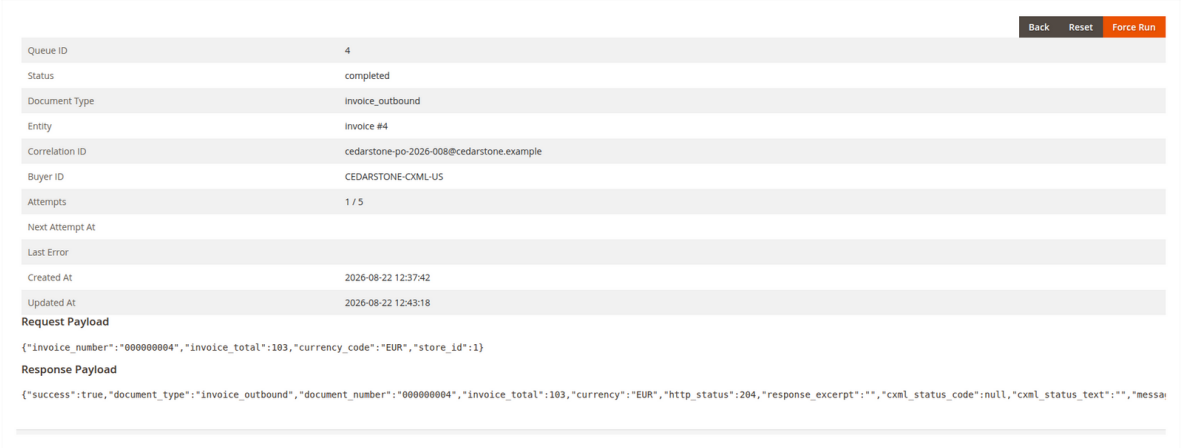
Queue processing

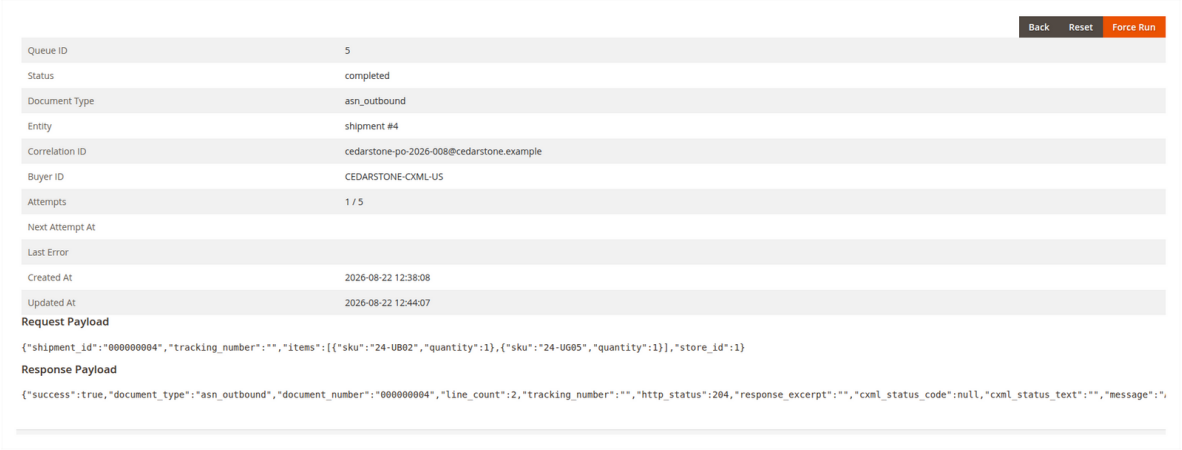




Each queue detail keeps the business entity, buyer, attempt count, safe request summary and delivery result together. The examples below show the three outbound document types from one completed order lifecycle; each receiver returned HTTP 204.







The `punchout_document_queue_process` job runs in Magento's `default` cron group every five minutes. It processes due `pending` rows per store, using the store-scoped batch size and retry settings.

For queue inspection and controlled document-type processing, see Operations and CLI.

Statuses are:

- `pending`: ready now or waiting until `next_attempt_at`;
- `processing`: claimed by a worker or Force Run;
- `completed`: the document endpoint answered successfully;
- `failed`: the maximum attempts were used.
- `delivery_unknown`: Magento cannot determine whether the remote system received the document, so automatic retry is stopped to avoid a duplicate business document.

Network exceptions and non-2xx responses count as failures. A row below its maximum returns to `pending` with the configured delay.

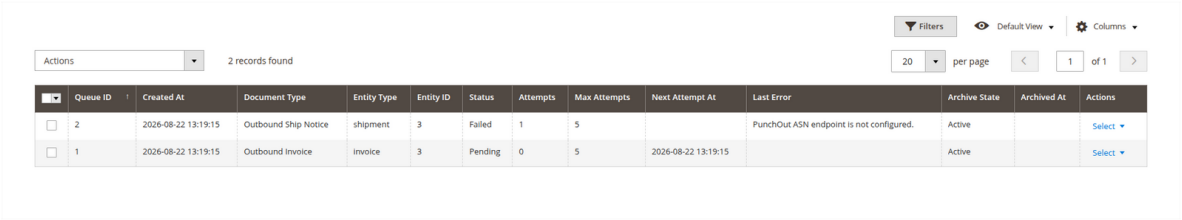
An interrupted delivery can produce `delivery_unknown` when the remote outcome is ambiguous. Review the receiving system first. Only an administrator with the dedicated permission can deliberately schedule that row again.



Confirmation delivery is durable. When the effective configuration requires a confirmation, the inbound-order record retains the immutable supplier routing and the queue can recover a missing confirmation intent after a worker or deployment interruption. The synchronous `OrderRequest` response does not wait for a remote confirmation endpoint. This keeps order acknowledgement separate from later server-to-server document delivery while still leaving the intent visible in **Document Queue**.

Work with a failed row

Open **Punchout > Document Queue**, then View the row.



- **Force Run** schedules the row immediately and processes it once. It does not erase the existing attempt count.
- **Reset** clears attempts, response and error data and returns the row to `pending`. Use it only after the underlying configuration or receiver problem has been corrected.
- **Retry Unknown Delivery** deliberately reschedules a `delivery_unknown` row after the receiving system has been checked.
- **Delete** removes the operational record and should not be used as a substitute for diagnosing delivery.

The module and queue must be enabled for the row's Store view, the retained supplier identity must be complete, and the document-specific URL must accept HTTPS POST requests. Read the saved HTTP response or transport error before resetting the attempt count.

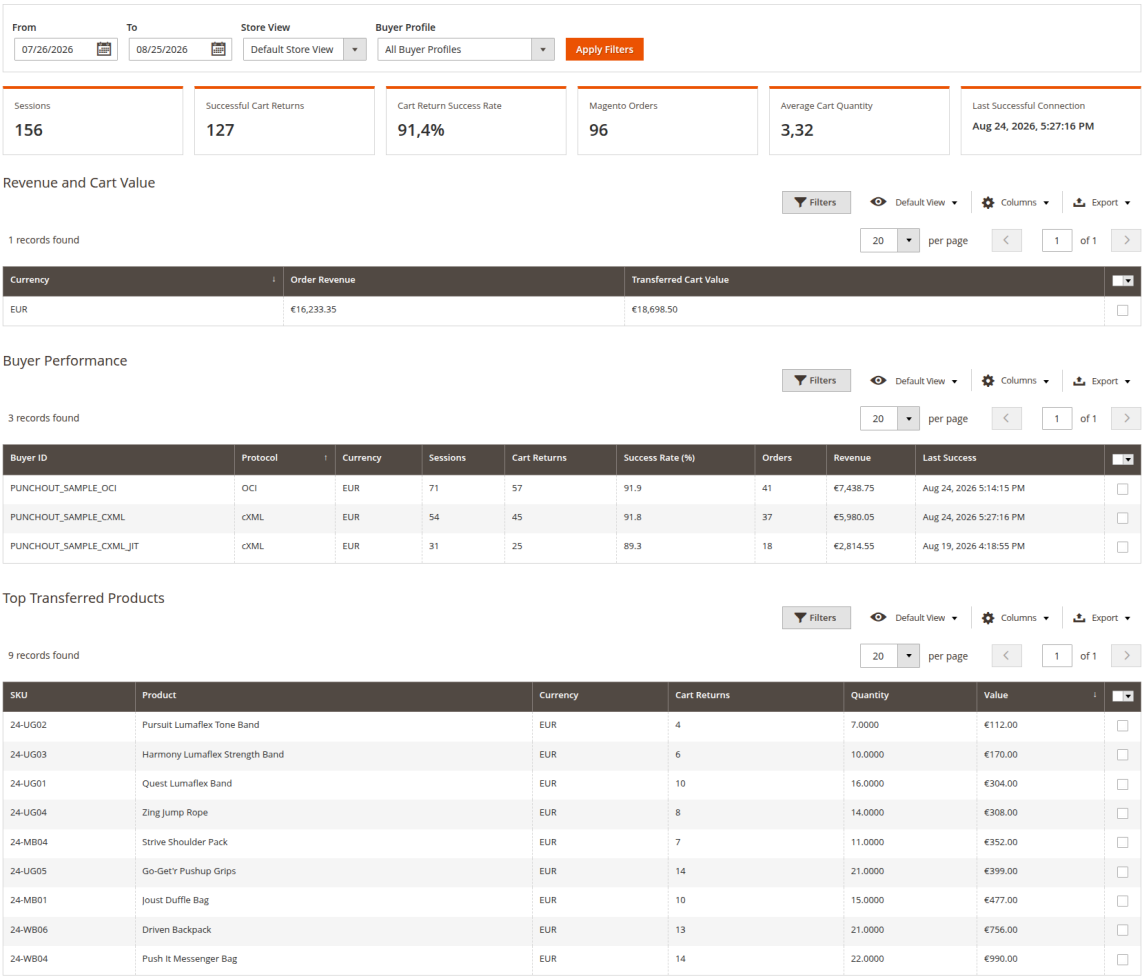
Direct document endpoints

`/punchout/document_asn/outbound` and `/punchout/document_invoice/outbound` execute the same document delivery handlers immediately and return their result. They use the effective document-specific endpoint and do not create a Magento shipment or invoice.

Send a client-generated `X-Punchout-Request-Key` for OrderRequest, ASN and invoice calls. Reuse it only for the same semantic request. Magento replays the saved result for an exact retry and rejects changed input that reuses the key.

Analytics

Open **Punchout > Analytics** to review daily PunchOut activity. Access is controlled by the PunchOut Analytics permission.



The summary gives you a quick view of:

- sessions;
- successful cart returns and their success rate;
- Magento orders;
- the average returned-cart quantity;
- the most recent successful connection.

Filter the report

Use the date range, Store View and Buyer Profile fields at the top of the page, then select **Apply Filters**. The date fields use the locale of the Magento Admin user. Day boundaries and displayed timestamps use the selected Store View timezone.

The Buyer Profile field lists profiles that already have analytics data. Select **All Store**

Views to include both global and store-specific activity in the same report.

Read and export the grids

The page contains three Magento grids:

- **Revenue and Cart Value** groups Magento order revenue and transferred cart value by currency.
- **Buyer Performance** shows sessions, cart returns, success rate, orders, revenue and the last successful connection for each Buyer Profile.
- **Top Transferred Products** shows returned SKUs, quantities, values and cart-return counts.

Use **Filters** to narrow a grid by its columns, and select a column heading to change the sort order. The **Export** menu downloads the filtered result while preserving the date, Store View and Buyer Profile filters from the page.

Money values remain separated by currency. The report does not combine different currencies into one total.

Enable or disable collection

Open **Punchout > Configuration > Analytics** and set **Enable PunchOut Analytics** for the required scope. New installations enable collection by default. Disabling it stops new analytics records for that scope; it does not delete existing aggregates.

Analytics data is stored separately from the technical trace. Trace retention therefore has no effect on the report.

Operations and CLI

Magento cron handles normal PunchOut cleanup, inbound-order retries and document delivery. The CLI commands are intended for monitoring, controlled recovery and deployments where an operator needs to process one document type explicitly.

Run all commands from the Magento project root.

Inspect the document queue

```
php bin/magento punchout:queue-status
```

The command groups queue entries by document type and status. It does not change any records.

Reconcile and send invoices

```
php bin/magento punchout:send-invoices --batch-size=100
```

Magento first recreates missing invoice queue entries for eligible PunchOut orders, then processes due invoice entries. Use a smaller batch during an investigation.

Reconcile and send ship notices

```
php bin/magento punchout:send-ship-notices --batch-size=100
```

This performs the same workflow for shipment-based `ShipNoticeRequest` documents.

Both send commands accept:

Option	Effect
`--batch-size`, `-b`	Limits reconciliation and processing to a positive number of documents. The default is 100.
`--skip-reconcile`	Processes existing due queue entries without reconstructing missing entries.

Understand the cron jobs

Job	Schedule	Purpose

Job	Schedule	Purpose
`punchout_cleanup_expired_contexts`	Every 15 minutes	Removes expired browser-session contexts.
`punchout_document_queue_process`	Every 5 minutes	Processes due confirmation, ship-notice and invoice entries.
`punchout_inbound_order_process`	Every 5 minutes	Retries eligible inbound orders.
`punchout_cleanup_retention`	Daily at 02:00	Applies configured trace, replay, snapshot, inbound-order and queue retention.

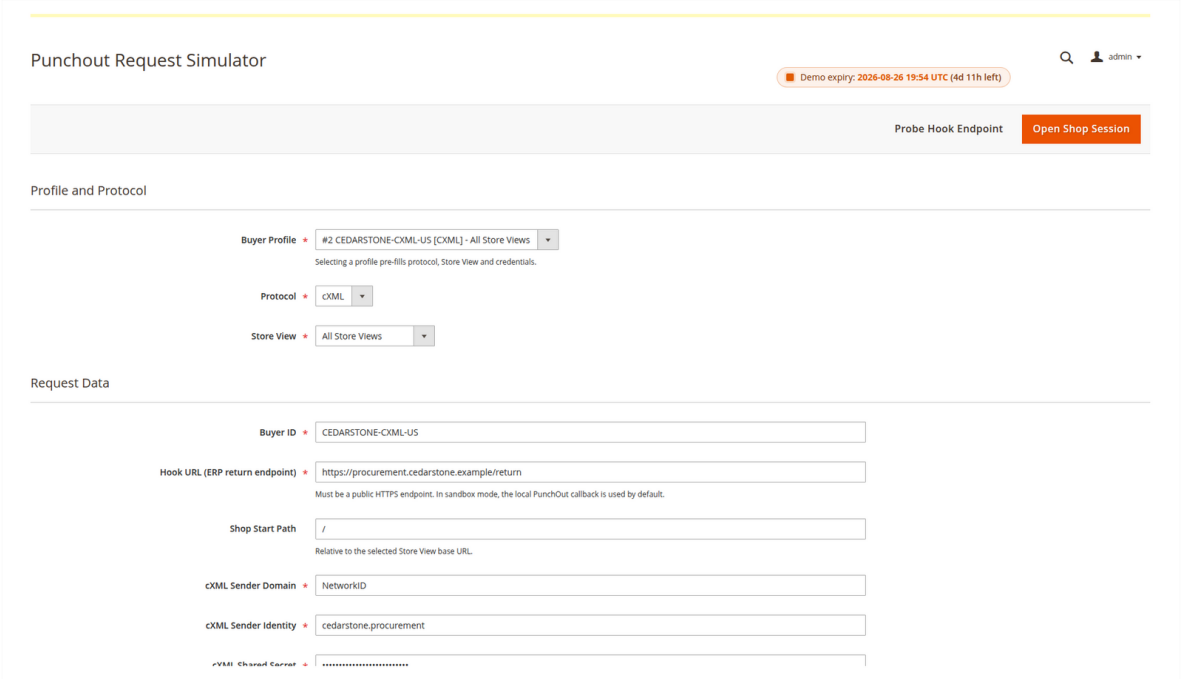
Do not run a send command repeatedly to bypass a `delivery_unknown` state. Check the receiving platform first, then use the protected Admin action when a deliberate retry is safe.

Trace and Simulator

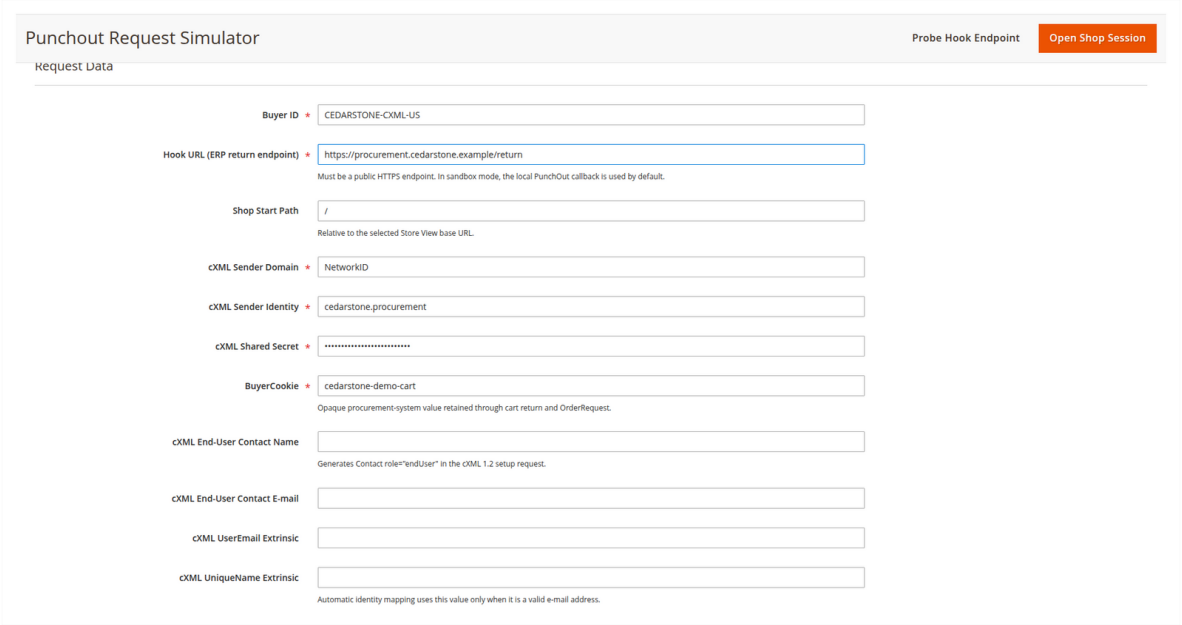
Use the simulator to build a controlled setup request. Use the trace viewer to follow a real request by correlation ID. Together they show whether a failure happened during authentication, setup, cart return, mapping or document delivery.

Build a simulator request

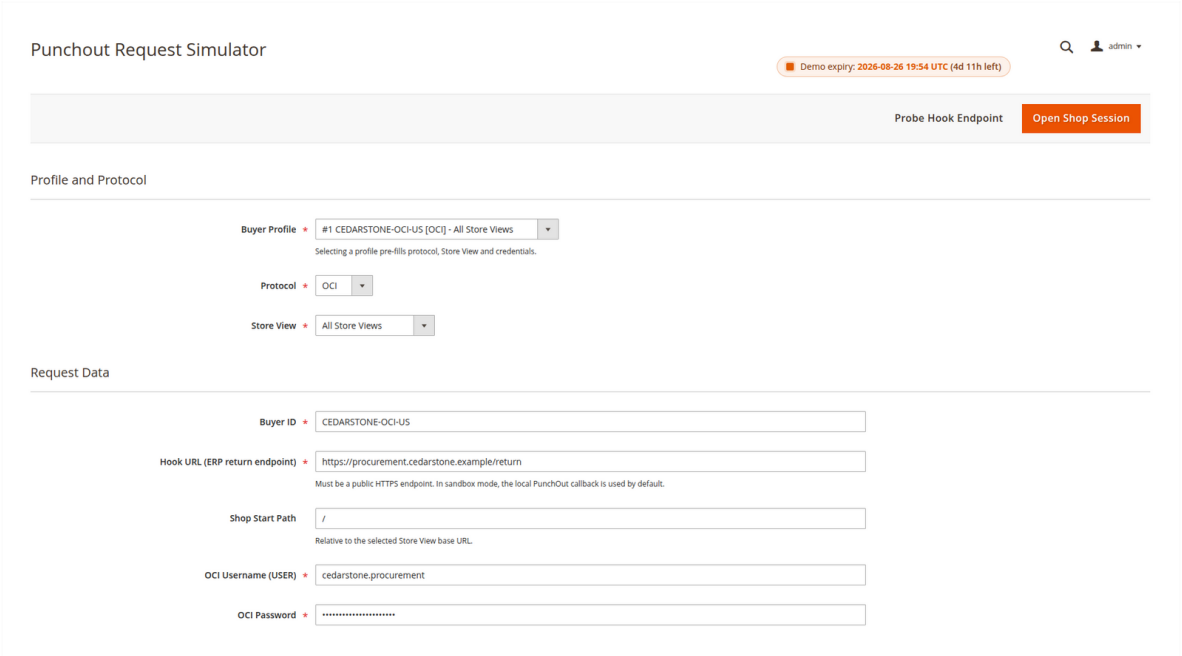
Open **Punchout > Punchout Simulator** and choose OCI or cXML. Use a configured buyer profile and a privacy-safe test customer in the local or buyer-provided sandbox.



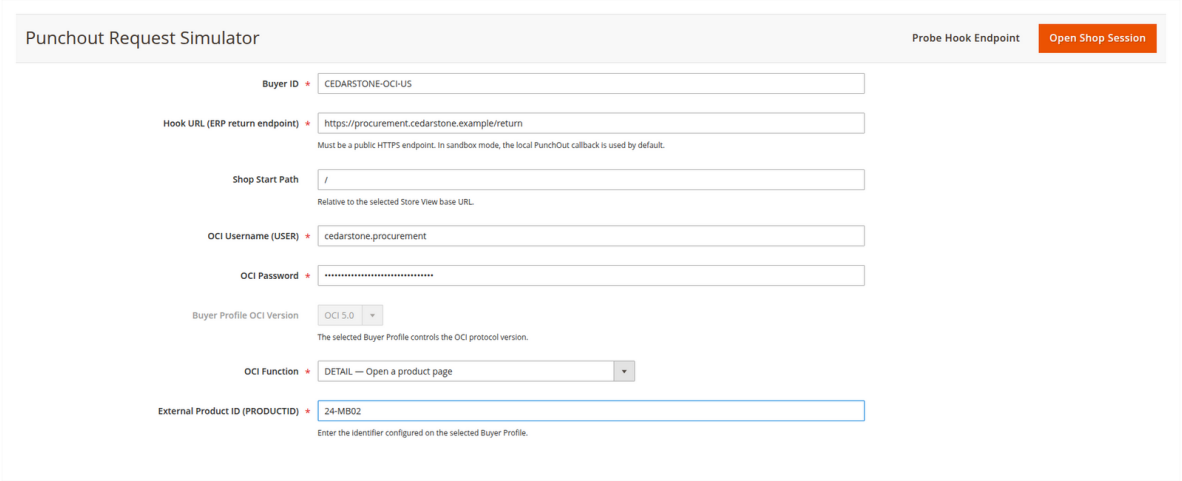
The request section keeps the stored shared secret masked while exposing the sender identity and the fictional BuyerCookie used to correlate the later cart return:



The same page can launch an OCI session with its own buyer identity and credentials:



For product-specific operations, the selected profile fixes the OCI version and the form requires the external product identifier expected by that profile:



The simulator requires a buyer profile. Selecting it fills the matching protocol, Store view and stored identity values, then updates the expanded technical preview automatically. For OCI profiles, the form also shows the profile-owned OCI version and the product identifier required by `DETAIL` or `DETAILADD`. Currency, tax mode and request signing come from the selected Store-view configuration rather than duplicate simulator fields.

The simulator can:

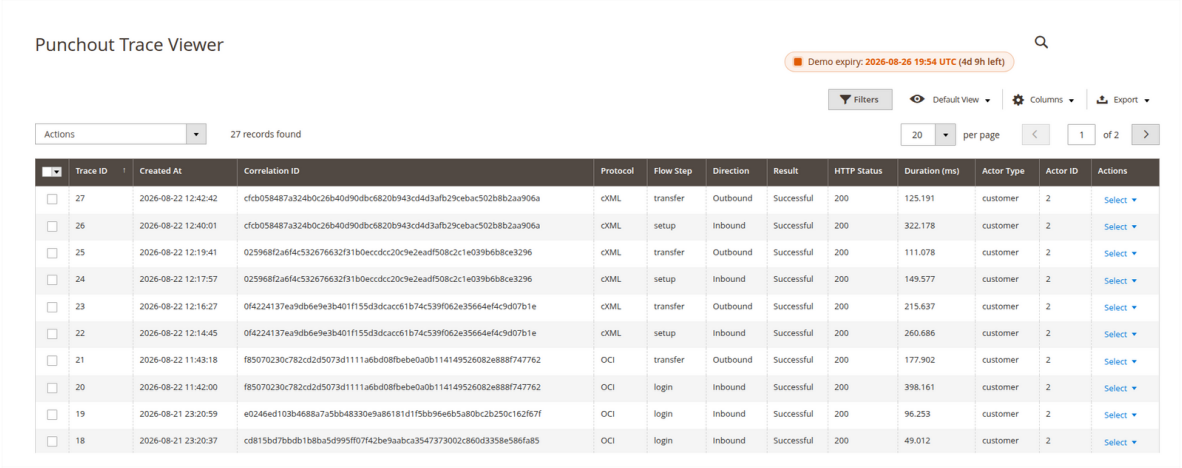
- build the protocol request from entered fields;
- show the target endpoint and technical payload preview;
- generate a cURL example;
- probe the return URL with the same URL-safety policy used by PunchOut;
- open a storefront session after a successful setup.

Open Shop Session keeps the Admin simulator open and launches the mapped storefront in a separate tab. For cXML it sends a standards-compliant `PunchOutSetupRequest`, reads the `StartPage/URL` from Magento's response and opens that URL. A fictional opaque BuyerCookie should still be treated as integration data and kept out of public screenshots when it resembles a real customer's identifier.

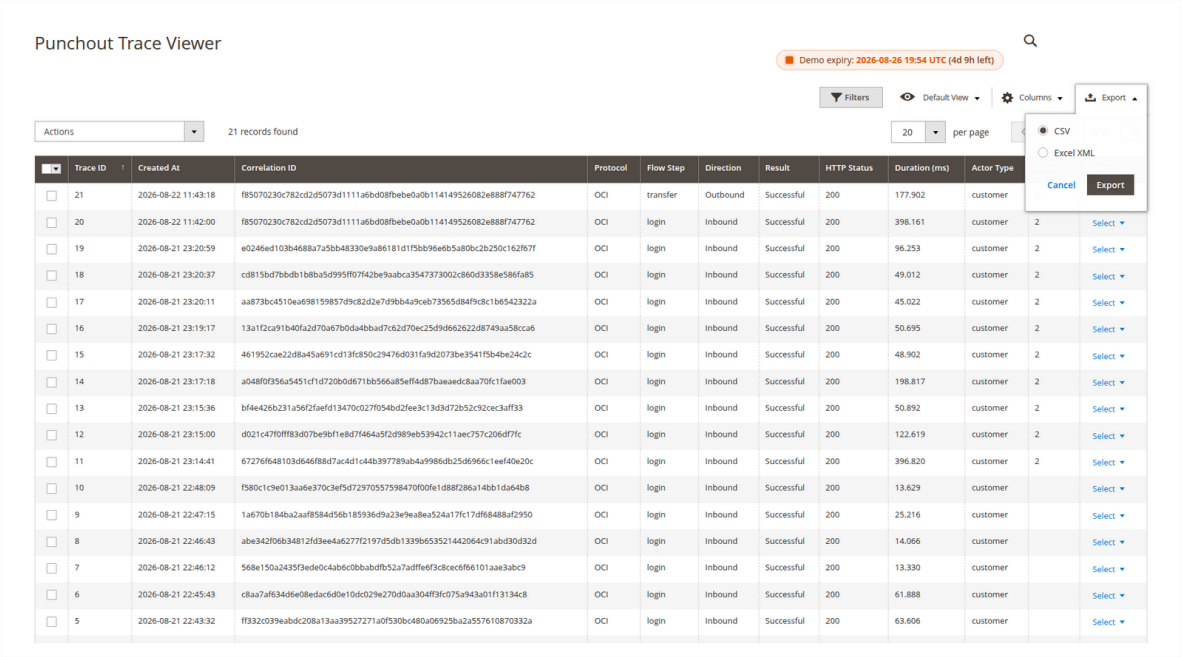
Do not paste a production shared secret into screenshots, tickets or documentation. A masked admin field proves that a value exists; it does not prove the other system uses the same value.

Read a trace

Open **Punchout Trace** and filter by correlation ID. One setup and its later actions use the same correlation value, which is more reliable than searching by time alone.



The grid exports the selected trace metadata as CSV or Excel XML without requiring administrators to open the stored payload:



A trace records direction, flow step, protocol, endpoint, HTTP status, duration, result and available customer context. Protocol, direction and result have typed filters in the grid. The stored payload is sanitized for passwords, shared secrets, tokens, BuyerCookie values and Magento form keys.

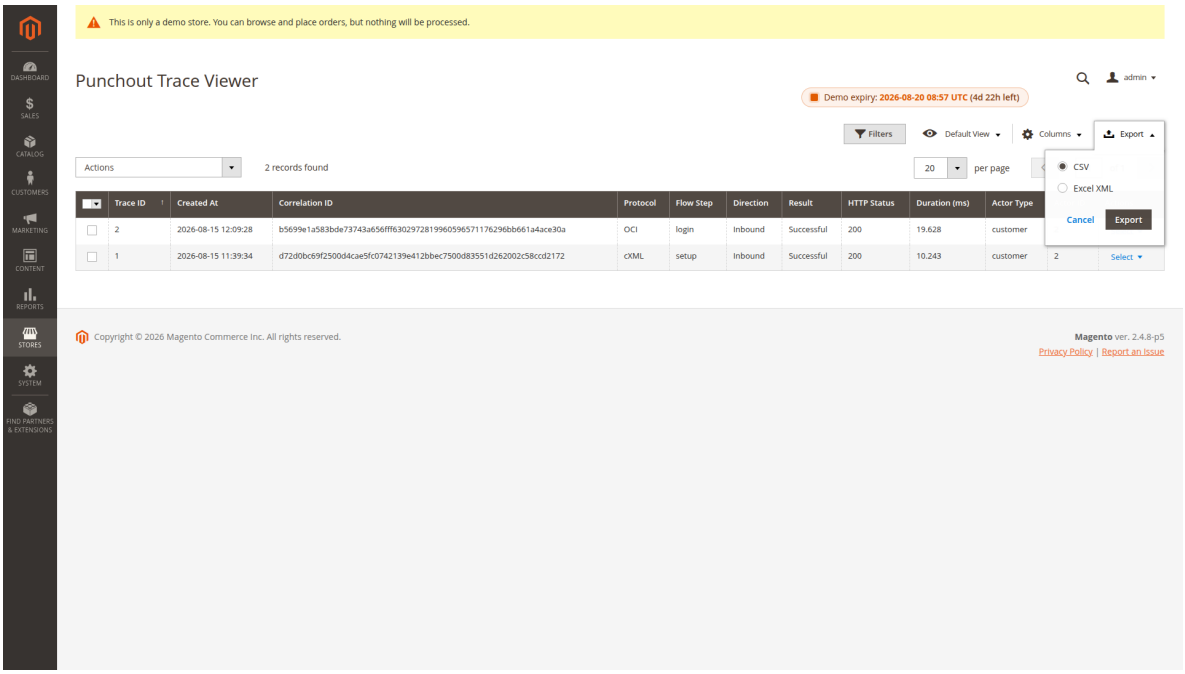
JSON export is useful when working with an integration partner, but review it before sharing. Sanitization reduces accidental disclosure; it is not a reason to make trace exports public.

A practical diagnosis sequence

1. Find the first failed step for the correlation ID.
2. Compare its store, buyer, protocol and endpoint with the intended profile.

3. Read the public `error_code` and any nested security codes.
4. Correct one cause, then generate a new semantic request and idempotency key.
5. Confirm the return payload on the buyer side, not only the HTTP 200 from Magento.

Old traces are removed daily according to **Trace Retention**. Export evidence needed for a support case before that window closes, without including credentials.



Protocol Compatibility

Choose the protocol and version on the Buyer Profile before exchanging test documents. An inbound request cannot switch a profile to another protocol or OCI version.

cXML

Capability	cXML 1.0	cXML 1.1	cXML 1.2
`PunchOutSetupRequest` and `PunchOutSetupResponse`	Yes	Yes	Yes
`PunchOutOrderMessage` cart return	Yes	Yes	Yes
ProfileRequest/ProfileResponse	No	Yes	Yes
Inbound `OrderRequest`	Yes	Yes	Yes
Confirmation, ship-notice and invoice documents	No	No	Yes, from 1.2.001

Magento preserves a supported request version when it builds the corresponding setup, profile or order response. Confirmation, ship-notice and invoice documents require cXML 1.2.001 or newer. Older OrderRequests can still be processed, but they do not start the outbound fulfilment-document loop.

The request extractor recognizes current schema URLs and the older `xml.cxml.org/dtd/<version>/cXML.dtd` form used by some established systems. XML parsing blocks external network access and unsafe document constructs.

OCI

Function	OCI 4.0	OCI 5.0	Behavior
Normal catalog entry	Yes	Yes	Opens a PunchOut shopping session.
`DETAIL`	Yes	Yes	Opens the resolved product page without allowing a cart change.
`DETAILADD`	No	Yes	Opens one product, permits one add and immediately returns it to the procurement system.
`QUANTITYCHECK`	No	Yes	Returns the product's available quantity and lead time as JSON without opening a session.

Function	OCI 4.0	OCI 5.0	Behavior
`VALIDATE`	No	No	Rejected as unsupported.
`BACKGROUND_SEARCH`	No	No	Rejected as unsupported.
`SOURCING`	No	No	Rejected as unsupported.

For product functions, `PRODUCTID` is resolved through the Buyer Profile's external product ID source. Use Magento SKU or a globally unique product attribute maintained for every relevant Store View.

`QUANTITYCHECK` is an availability query on `/punchout/oci/login`, not a shopping session. The request authenticates the buyer as usual but does not require `HOOK_URL`, creates no context token and involves no storefront operation policy. `EXT_PRODUCT_ID` is resolved through the profile's external product ID source; `QUANTITY` is optional and defaults to `1`. The JSON response returns `AVAILABLE_QUANTITY` from the resolved store's MSI salable quantity and `LEADTIME` in whole days from the product attribute selected as **OCI Quantity Check Lead Time Attribute** on the Buyer Profile. See the OCI Flow for the request and response details.

Mapping and buyer contracts

Protocol versions do not make a generic mapping automatically valid for a buyer. Confirm required fields, units, decimal precision, classification domains, custom fields and return behavior against the procurement platform's current integration specification.

Use separate Buyer Profiles when two procurement accounts require different protocols, OCI versions, credentials, Store Views or field contracts.

Endpoint Reference

All paths are relative to the Magento storefront base URL. POST endpoints accept form values or JSON. cXML endpoints also parse a safe XML body with external network access disabled during XML parsing.

Setup and transfer

Path	Method	Purpose
`/punchout/oci/login`	GET or POST	Authenticate an OCI buyer and create a context token.
`/punchout/cxml/profile`	POST	Authenticate a cXML buyer and return its enabled Store-specific transactions.
`/punchout/cxml/setup`	POST	Parse a `PunchOutSetupRequest`, authenticate the sender and create a context token.
`/punchout/oci/transfer`	POST	Build a mapped OCI auto-submit form and return it in JSON.
`/punchout/cxml/transfer`	POST	Build a mapped `PunchOutOrderMessage` and return it in JSON.

OCI login requires a buyer identity (`buyer_id` or `USER`), matching OCI credentials, and `hook_url` or `HOOK_URL`. cXML setup reads sender identity/domain/shared secret and `BrowserFormPost/URL` from the XML body or their normalized request-field equivalents.

Profile Discovery accepts cXML 1.1 and 1.2 `ProfileRequest` documents. The response advertises only the setup and OrderRequest transactions enabled on the resolved Buyer Profile. See cXML Profile Discovery.

A successful setup response includes `token`, `correlation_id`, actor type and status. cXML also returns parsed level-2 metadata and a deep-link hint when a selected item or supported extrinsic is present.

OCI login, cXML setup and `/punchout/document_po/inbound` accept an optional `store_id` request parameter to select the Store View used for module-status checks and buyer resolution. A Buyer Profile pinned to a specific Store View still overrides the requested value.

Transfer requires an active token. `X-Punchout-Request-Key` is recommended for safe retries. The JSON response contains `content_type` and `payload`; the caller is responsible for using that generated return document. The storefront return route performs the browser post automatically.

Browser session

Path	Method	Purpose
`/punchout/session/activate`	GET	Bind a valid token to the browser, sign in the mapped customer and continue to a same-host target.
`/punchout/session/back`	GET	Attach the active cart and post the mapped OCI/cXML result to the return URL.
`/punchout/session/logout`	POST	Expire the context and clear the active browser token.

Cart

Path	Method	Purpose
`/punchout/cart/add`	POST	Add one item or an array of SKU/quantity items.
`/punchout/cart/update`	POST	Update by quote item ID or SKU.
`/punchout/cart/remove`	POST	Remove one or several item IDs/SKUs.
`/punchout/cart/get`	GET or POST	Return the current PunchOut cart summary.

These routes accept the token through `token` or `X-Punchout-Token`. A missing token returns HTTP 400; a missing or expired context returns HTTP 410.

Document loop

Path	Method	Purpose
`/punchout/document_po/inbound`	POST	Authenticate and process a cXML OrderRequest in acknowledge, validate or create-order mode.
`/punchout/document_asn/outbound`	POST	Deliver `ShipNoticeRequest` cXML immediately to the effective ASN URL.

Path	Method	Purpose
`/punchout/document_invoice/outbound`	POST	Deliver `InvoiceDetailRequest` cXML immediately to the effective Invoice URL.

All three support `X-Punchout-Request-Key`. The inbound OrderRequest endpoint also accepts the optional `store_id` parameter described above. Request-body aliases and alternate idempotency headers are not accepted. The OrderRequest authenticates the cXML sender against the buyer profile and can use a retained returned-cart snapshot even when the short-lived browser session has expired. Later shipment and invoice queue processing uses durable routing data from the inbound order rather than the browser cart-return URL.

Diagnostics

Path	Method	Availability
`/punchout/sandbox/callback`	GET or POST	Available only when Sandbox Callback Mode is enabled for the selected scope.

Admin operational map

Punchout is a main Magento Admin menu. Each screen has a separate ACL resource, so an administrator can inspect traces without being allowed to change buyer credentials or retry a document.

Admin screen	Owns	Protected actions
Punchout Buyers	External identities, customer resolution and overrides	Create, edit, deactivate and delete require Buyer Management permission
Punchout Mappings	Outbound runtime profiles and template copies	Save, duplicate and delete require Mapping Management permission
Punchout Trace	Sanitized request and response diagnostics	View and export require Trace Viewer permission
Inbound Orders	Acknowledge, validate, create, update and cancellation records	Process, retry and reject require Inbound Order Management; deletion has a separate permission

Admin screen	Owns	Protected actions
Document Queue	Confirmation, ASN and invoice delivery state	View, force-run and reset require Document Queue permission; retrying unknown delivery has an additional permission
Analytics	Daily sessions, cart returns, orders, revenue, Buyer performance and transferred products	View, filter and export require PunchOut Analytics permission
Punchout Simulator	Probe, request preview and sandbox session launch	All simulator actions require Simulator permission
Configuration	Store-view defaults, security, operation policy and retention	Save requires Punchout Configuration permission

State-changing Admin actions use Magento's normal POST and form-key protection. Do not reproduce an action URL as a GET request in an integration test. For operational evidence, capture the status, correlation ID and sanitized result, not the raw request body or credential fields.

Admin Field Reference

Use this page when you need the literal Magento field or configuration path. For setup decisions and consequences, start with Configuration.

Buyer profile

Field	Required or effect
Store	Required. Selects catalog, customer website and scoped PunchOut configuration. `All Store Views` creates a global fallback profile.
Buyer ID	Required external identity used to resolve the profile.
Protocol	Required: `oci` or `cxml`. Controls visible credentials and compatible mappings.
cXML Sender Domain	Required for cXML identity matching.
cXML Sender Identity	Required for cXML identity matching and normally the resolved buyer ID.
cXML Shared Secret	Required for a new cXML buyer; encrypted at rest. Use the Change switch to replace it later.
Accept PunchOutSetupRequest	Enables cXML setup for this buyer and advertises the transaction through Profile Discovery.
Accept OrderRequest	Enables inbound cXML orders for this buyer and advertises the transaction through Profile Discovery.
OCI Username	Required for OCI identity matching.
OCI Password	Required for a new OCI buyer; encrypted at rest. Use the Change switch to replace it later.
Customer Resolution Mode	Selects guest access, a fixed Magento customer, customer e-mail mapping, or optional/required request customer ID mapping.
Magento Customer	Required only for the fixed-customer mode. The selected account must belong to the buyer profile's website.
Identity Source	Selects the protocol field or cXML Extrinsic that supplies the customer e-mail or request customer ID.

Field	Required or effect
JIT Provisioning	Inherits the Store-view setting or explicitly enables/disables customer creation when e-mail mapping finds no account.
JIT Customer Group	Customer group for accounts created through JIT provisioning.
OCI First/Last Name Parameter	Request parameter names used to populate a JIT-created OCI customer.
cXML First/Last Name Extrinsic	Extrinsic names used to populate a JIT-created cXML customer.
Company ID	Optional context value for compatible bridge/mapping logic.
Price Group ID	Optional context value for compatible pricing/mapping logic.
OCI Version	Selects the OCI contract used by this buyer profile.
OCI External Product ID Source	Uses Magento SKU or a selected globally unique product attribute for OCI `PRODUCTID`.
OCI External Product ID Attribute	Required when the external product ID source is a product attribute.
OCI Quantity Check Lead Time Attribute	Optional for OCI buyers. Selects a global `varchar`/`int` product attribute whose whole-day value is returned as `LEADTIME` in OCI 5.0 `QUANTITYCHECK` responses.
Mapping Profile	Optional selector limited to active outbound runtime profiles with matching protocol and store scope. Required for controlled mapped output.
Active	Allows new requests to resolve this buyer.

Storefront Entry

- **Start Destination** is required and selects Shopping Cart, Store Homepage, CMS Page, Category or Store-relative Path.
- **CMS Page** is required only for the CMS Page destination and is limited to pages available in the selected Store View.
- **Category** is required only for the Category destination and is resolved in the selected Store View catalog.
- **Store-relative Path** is required only for the path destination. External URLs are rejected.

Request-specific and Level-2 product destinations take priority over this general entry

setting.

Inbound Order Overrides

Every override is optional. An empty value uses the configuration of the buyer's Store view.

Field	Effect
Processing Mode	Overrides acknowledge, validate or create-order behavior for this buyer.
Authoritative Price Source	Overrides the Store-view price authority.
Returned Cart Line Matching	Selects auxiliary-then-SKU or SKU-only matching.
Changed Quantity, New Item, Removed Item, Missing Snapshot	Controls each returned-cart difference independently.
Price Mismatch Action and tolerances	Sets buyer-specific unit-price acceptance.
Customer, Product and Address Resolution	Changes how the Magento quote context is built.
Shipping and Payment Method	Selects buyer-specific quote methods.
Update and Cancellation Policy	Controls later `OrderRequest` revisions and deletes.
Send Customer Order Email	Overrides Magento's normal order email setting.
Send ConfirmationRequest and Confirmation URL	Controls buyer-specific cXML confirmation delivery.
Confirmation Mode	Selects automatic whole-order confirmation or manual decisions for every order line.
ASN URL	Overrides the Store-view endpoint for `ShipNoticeRequest`. It is not the cart-return URL.
Invoice URL	Overrides the Store-view endpoint for `InvoiceDetailRequest`. It is not the cart-return URL.
Outbound cXML Response Mode	Allows an empty successful HTTP response or requires a successful cXML response status.
Maximum Processing Attempts	Limits processing attempts for this buyer.

Advanced transport authentication

Mode	Request value
None	No additional transport credential. Protocol credentials are still required.
Basic	`Authorization: Basic ...` using the separately configured username/password.
Bearer	`Authorization: Bearer ...` or `X-Punchout-Auth-Token`.
Shared Secret Header	`X-Punchout-Shared-Secret`.

Mapping profile

Field	Required or effect
Store	Required. `All Store Views` makes the mapping available across stores.
Code	Required and unique in scope; lowercase letters, numbers, hyphens and underscores only.
Name	Required admin-facing description.
Protocol	`oci` or `cxml`.
Direction	`inbound` or `outbound`; buyers can select outbound profiles only.
Reusable Template Profile	Marks the profile as a copy source rather than a runtime buyer mapping. Synced system templates can be updated during `setup:upgrade`.
Start From Template	New-profile helper that copies compatible rules; not persisted as a profile relationship.
Active	Makes a runtime profile available to buyer configuration.

Mapping rows contain **Data Source Area**, **Source Value**, **Punchout Target Key**, optional **Custom Target Key**, **Value Formatting**, **Fallback Value**, **Mandatory** and **Export Position**.

System configuration

General (`punchout/general/*`)

Config path	Admin label	Default
`punchout/general/enabled`	Enable Punchout	Yes

Config path	Admin label	Default
punchout/general/return_action_label	Punchout return action label	Return to System
punchout/general/sandbox_mode	Enable Sandbox Callback Mode	No

Security (punchout/security/*)

Config path	Admin label	Default
punchout/security/strict_signature	Require Strict Signature Validation	No
punchout/security/signature_secret	Signature Secret	Empty
punchout/security/allowed_ips	Allowed IP Ranges	Empty; permits all source IPs
punchout/security/allow_request_customer_id_oci	Allow OCI request customer override	No
punchout/security/allow_request_customer_id_cxml	Allow cXML request customer override	No
punchout/security/allow_insecure_outbound_tls	Allow insecure outbound TLS in sandbox mode	No

Cart (punchout/cart/*)

Config path	Admin label	Default
punchout/cart/allow_inbound_item_prices	Allow inbound cart item prices	No

cXML Profile Discovery (`punchout/profile/*`)

Config path	Admin label	Default
`punchout/profile/enabled`	Enable cXML Profile Discovery	Yes
`punchout/profile/public_base_url`	Public PunchOut Base URL	Empty; uses the Store View base URL

Customer Identity (`punchout/identity/*`)

Config path	Admin label	Default
`punchout/identity/jit_enabled`	Enable JIT Customer Provisioning	No
`punchout/identity/jit_customer_group_id`	JIT Customer Group	Empty

Inbound Orders (`punchout/inbound_order/*`)

Config path	Admin label	Default
`punchout/inbound_order/processing_mode`	Processing Mode	Acknowledge Only
`punchout/inbound_order/price_source`	Authoritative Price Source	Current Magento Pricing
`punchout/inbound_order/line_match_strategy`	Returned Cart Line Matching	Auxiliary ID, then SKU
`punchout/inbound_order/changed_quantity_action`	Changed Quantity Action	Recalculate with Magento
`punchout/inbound_order/new_item_action`	New Item Action	Recalculate with Magento
`punchout/inbound_order/removed_item_action`	Removed Item Action	Allow Removal
`punchout/inbound_order/missing_snapshot_action`	Missing/Expired Snapshot Action	Reject
`punchout/inbound_order/price_mismatch_action`	Price Mismatch Action	Reject Order
`punchout/inbound_order/price_tolerance_absolute`	Absolute Unit Price Tolerance	`0.01`
`punchout/inbound_order/price_tolerance_percent`	Percent Unit Price Tolerance	`0`

Config path	Admin label	Default
`punchout/inbound_order/customer_resolution`	Customer Resolution	Buyer Profile Customer
`punchout/inbound_order/product_resolution`	Product Resolution	Supplier Part ID
`punchout/inbound_order/address_source`	Address Source	Incoming Order Addresses
`punchout/inbound_order/shipping_method`	Shipping Method	Flat Rate
`punchout/inbound_order/payment_method`	Payment Method	PunchOut Purchase Order
`punchout/inbound_order/update_policy`	Order Update Policy	Manual Review
`punchout/inbound_order/cancellation_policy`	Order Cancellation Policy	Manual Review
`punchout/inbound_order/send_customer_email`	Send Customer Order Email	No
`punchout/inbound_order/confirmation_enabled`	Send cXML ConfirmationRequest	No
`punchout/inbound_order/confirmation_mode`	Confirmation Mode	Automatic Whole-Order Confirmation
`punchout/inbound_order/confirmation_url`	Default Confirmation URL	Empty
`punchout/inbound_order/asn_url`	ASN URL	Empty
`punchout/inbound_order/invoice_url`	Invoice URL	Empty
`punchout/inbound_order/response_mode`	Outbound cXML Response Mode	Allow Empty HTTP 2xx Response
`punchout/inbound_order/supplier_domain`	Supplier Credential Domain	Empty
`punchout/inbound_order/supplier_identity`	Supplier Identity	Empty
`punchout/inbound_order/supplier_shared_secret`	Supplier Shared Secret	Empty
`punchout/inbound_order/max_attempts`	Maximum Processing Attempts	`5`
`punchout/inbound_order/batch_size`	Retry Batch Size	`20` at Default Config
`punchout/inbound_order/retry_delay_minutes`	Retry Delay	`15` minutes
`punchout/inbound_order/retention_days`	Inbound Order Payload Retention (days)	`365` days

Config path	Admin label	Default
`punchout/inbound_order/snapshot_retention_days`	Returned Cart Snapshot Retention	`90` days

Trace, analytics, debug and session

Config path	Admin label	Default
`punchout/trace/enabled`	Enable Trace Logging	Yes
`punchout/trace/retention_days`	Trace Retention (days)	30
`punchout/analytics/enabled`	Enable PunchOut Analytics	Yes
`punchout/debug/enabled`	Enable Debug Logging	No
`punchout/session/token_expiration_hours`	Token Expiration (hours)	2

Debug logging writes technical diagnostics to `var/log/punchout.log`. It does not write sensitive protocol payloads.

Document Queue (`punchout/queue/*`)

Config path	Admin label	Default
`punchout/queue/enabled`	Enable Queue Processing	Yes
`punchout/queue/batch_size`	Batch Size per Cron Run	20
`punchout/queue/max_retry`	Maximum Retry Attempts	5
`punchout/queue/retry_delay_minutes`	Retry Delay (minutes)	15
`punchout/queue/retention_days`	Completed Queue Retention (days)	365

Retention

Config path	Admin label	Default
`punchout/request_replay/retention_days`	Request Replay Retention (days)	7
`punchout/retention/batch_size`	Cleanup Batch Size	500

Operation Policy (`punchout/operation/*`)

Config path	Admin label	Default
`punchout/operation/policy_enabled`	Enable Frontend Operation Policy	Yes
`punchout/operation/landing_route`	Restricted Action Landing Route	`checkout/cart`
`punchout/operation/create_allowed_actions`	Allowed Full Actions for CREATE	`*`
`punchout/operation/edit_allowed_actions`	Allowed Full Actions for EDIT	`*`
`punchout/operation/inspect_allowed_actions`	Allowed Full Actions for INSPECT	Cart read and PunchOut return/logout actions
`punchout/operation/detail_allowed_actions`	Allowed Full Actions for DETAIL	Selected product page and session support actions
`punchout/operation/detail_add_allowed_actions`	Allowed Full Actions for DETAILADD	Selected product, Add to Cart and immediate return actions

Embedded PunchOut (`punchout/iframe/*`)

These settings are provided by the `mageb2b/punchout-iframe` add-on. See Iframe Compatibility Add-On for behavior and buyer-level overrides.

Config path	Admin label	Default
`punchout/iframe/enabled`	Enable Iframe Compatibility	No
`punchout/iframe/allowed_origins`	Allowed Procurement Origins	Empty

Extension Points

Keep the PunchOut package unchanged. Add project-specific behavior in a separate Magento module through dependency injection, plugins, service contracts and events. This keeps Composer updates predictable and separates customer-specific integration code from the product.

Public service contracts

Use these interfaces for the main integration workflows:

Interface	Purpose
`MageB2B\Punchout\Api\ContextManagerInterface`	Creates and reads PunchOut session contexts.
`MageB2B\Punchout\Api\PunchoutContextInterface`	Defines the active session context passed between PunchOut services and events.
`MageB2B\Punchout\Api\TransferServiceInterface`	Prepares the cart return for OCI or cXML.
`MageB2B\Punchout\Api\ProtocolAdapterInterface`	Defines protocol-specific cart output.
`MageB2B\Punchout\Api\InboundOrderManagementInterface`	Processes and manages inbound cXML orders.
`MageB2B\Punchout\Api\InboundOrderRepositoryInterface`	Reads durable inbound-order records.
`MageB2B\Punchout\Api\JitCustomerProvisionerInterface`	Creates customers for JIT e-mail mapping.
`MageB2B\Punchout\Api\LineConfirmationManagementInterface`	Validates line decisions and queues one manual cXML confirmation.
`MageB2B\Punchout\Api\OciProductIdentifierInterface`	Resolves the configured OCI external product identifier.

Interface	Purpose
`MageB2B\Punchout\Api\OrderLineMetadataResolverInterface`	Resolves external line numbers and units for outbound documents.
`MageB2B\Punchout\Api\CxmlBuyerCredentialResolverInterface`	Resolves the effective cXML Buyer credentials for durable outbound documents.
`MageB2B\Punchout\Api\OutboundDocumentEndpointResolverInterface`	Resolves the effective Confirmation, ASN or Invoice endpoint.
`MageB2B\Punchout\Api\TraceRepositoryInterface`	Persists sanitized technical trace records.
`MageB2B\Punchout\Api\Bridge\ActorResolverInterface`	Supplies project-specific actor context.
`MageB2B\Punchout\Api\Bridge\PermissionGuardInterface`	Applies project-specific transfer permissions.

Prefer a plugin when the standard service already performs the required workflow and you only need to adjust its input or result. Use a preference only when your implementation must replace the complete contract.

Inject the interface rather than a concrete model. Magento's dependency injection configuration supplies the default implementation, and a separate integration module can add a plugin or replace the contract where a full implementation is required.

Do not use the trace repository for operational payload storage. Its input is intended for sanitized diagnostics. Inbound-order and document services retain their own operational data separately.

Lifecycle events

Event	When to use it
`punchout_actor_resolve_before`	Add actor candidates before actor resolution.

Event	When to use it
`punchout_actor_resolve_after`	Read the resolved actor context and source.
`punchout_permission_check_before`	Add project data before transfer authorization.
`punchout_permission_check_after`	React to a successful permission check.
`punchout_transfer_prepare_before`	Add data before Magento prepares the cart payload.
`punchout_transfer_prepare_after`	Inspect or extend a valid prepared transfer result.
`punchout_transfer_success`	React after Magento has generated the handoff payload. This event does not prove that the receiving browser processed it.
`punchout_customer_authentication`	React after Magento has checked customer status, lock state and confirmation.
`punchout_inbound_identity_resolved`	Adjust normalized identity data before customer mapping.
`punchout_jit_customer_created`	React after PunchOut creates a Magento customer through JIT provisioning.

Event	When to use it
`punchout_inbound_address_resolved`	Adjust a resolved billing or shipping address before quote assignment.
`punchout_inbound_order_processed`	React after inbound-order processing has reached its result.
`punchout_cxml_order_request_build_before_serialize`	Add supported cXML data to the mutable document before serialization.
`punchout_mapping_source_field_options`	Add fields to an Admin mapping source.
`punchout_mapping_context_values`	Supply runtime values for context mapping rules.
`punchout_document_order_after`	React to a Magento order document event with durable PunchOut context.
`punchout_document_shipment_after`	React to a shipment document event.
`punchout_document_invoice_after`	React to an invoice document event.
`punchout_document_event`	Handle order, shipment and invoice events through one common event.

Observers should remain small and must not modify stored credentials or unsanitized trace data. Put substantial integration logic behind an injected service and call that service from the observer.

Mapping extensions

Use `punchout_mapping_source_field_options` to expose a new typed source field in Magento Admin. Supply its runtime value through `punchout_mapping_context_values`. This keeps the mapping editor and runtime evaluation in sync.

Protected protocol fields remain owned by PunchOut. For example, the OCI external product ID comes from the buyer profile and cannot be replaced by an ordinary mapping rule.

Templates Overview

PunchOut ships 23 mapping fixtures. Magento synchronizes them during `setup:upgrade` and marks them as active reusable templates in **All Store Views**.

Select a template only while creating a mapping profile. Its rules are copied into the new profile, where you can review and change them. A future package update may refresh the reusable system template but does not silently rewrite the copied runtime profile.

Runtime boundary

Buyer profiles can use only active **outbound**, non-template profiles. The six inbound fixtures document expected setup fields and support future/custom normalization work, but the current OCI login and cXML setup controllers parse their supported fields directly. Assigning an inbound profile to a buyer is not supported by the core runtime.

Template names describe an intended starting point. They do not certify an integration with SAP, SAP Ariba, Coupa or Oracle Procurement. Confirm protocol versions, mandatory fields, classification schemes and custom extrinsics with the buyer.

Complete template catalog

Template	Direction	Intended starting point	Runtime note
`oci_inbound_core`	Inbound	Generic OCI login fields	Reference/custom use; not assigned to buyers
`oci_outbound_core`	Outbound	Six required OCI cart fields	Copy to a runtime profile
`oci_outbound_extended`	Outbound	Optional OCI enrichments	Combine with required core rules as needed
`sap_oci_v4_inbound_login`	Inbound	SAP OCI 4 login fields	Reference/custom use; not assigned to buyers
`sap_oci_v4_outbound_core`	Outbound	SAP OCI 4 cart return	Copy to a runtime profile
`sap_oci_v5_inbound_login`	Inbound	SAP OCI 5 login fields	Reference/custom use; not assigned to buyers
`sap_oci_v5_outbound_core`	Outbound	SAP OCI 5 cart return	Copy to a runtime profile
`sap_oci_v5_outbound_hierarchy`	Outbound	OCI 5 hierarchy/material-group fields	Copy when the buyer requires those fields
`sap_oci_v5_outbound_secure`	Outbound	Eight mandatory OCI 5 fields	Use only with complete source data
`cxml_inbound_posr_core`	Inbound	Generic `PunchOutSetupRequest` fields	Reference/custom use; not assigned to buyers

Template	Direction	Intended starting point	Runtime note
`cxml_outbound_core`	Outbound	Core `PunchOutOrderMessage` item data	Copy to a runtime profile
`cxml_outbound_level2`	Outbound	Optional level-2 enrichments	Add required core rules as needed
`ariba_cxml_inbound_posr_1_2_x`	Inbound	Ariba-oriented setup fields	Reference/custom use; not assigned to buyers
`ariba_cxml_outbound_core`	Outbound	Ariba-oriented core item data	Copy to a runtime profile
`ariba_cxml_outbound_level2`	Outbound	Manufacturer and buyer extrinsics	Copy when required by the buyer
`ariba_cxml_outbound_level2_strict`	Outbound	Same 12 fields, all mandatory	Use only with complete source data
`coupa_cxml_inbound_posr_1_2_014`	Inbound	Coupa-oriented setup fields	Reference/custom use; not assigned to buyers
`coupa_cxml_outbound_poom_1_2_023`	Outbound	Coupa-oriented order message	Copy to a runtime profile
`coupa_cxml_outbound_poom_1_2_023_strict`	Outbound	Same 10 fields, all mandatory	Use only with complete source data
`jaggaer_cxml_outbound_core`	Outbound	JAGGAER-oriented core cart fields and optional UNSPSC	Copy to a runtime profile; verify BuyerCookie compatibility
`oracle_cxml_inbound_posr_1b`	Inbound	Oracle-oriented setup fields	Reference/custom use; not assigned to buyers
`oracle_cxml_outbound_1b_core`	Outbound	Oracle-oriented order message	Copy to a runtime profile
`oracle_cxml_outbound_1b_core_strict`	Outbound	Same 10 fields, all mandatory	Use only with complete source data

Continue with the Template Selection Guide and the detailed protocol pages:

- OCI and SAP Templates
- cXML Core Templates
- SAP Ariba Templates
- Coupa Templates
- JAGGAER cXML Setup
- Oracle Procurement Templates

OCI and SAP Templates

These fixtures cover generic OCI and SAP-oriented OCI 4/5 field sets. Copy an outbound template into a runtime profile, then compare every target field with the buyer's actual OCI specification.

Inbound reference fixtures

`oci_inbound_core`

Eight rules describe common login values: required `FUNCTION` and `HOOK_URL`, plus optional `USER`, `PASSWORD`, buyer ID, currency, language and session ID.

`sap_oci_v4_inbound_login`

Seven rules describe the OCI 4 login baseline: required function and hook URL plus optional credentials, buyer, currency and language.

`sap_oci_v5_inbound_login`

Nine rules add `~OkCode` and session ID to the OCI 4-oriented set.

These inbound fixtures are synchronized into the catalog, but the core login controller does not execute inbound mapping rules. It parses supported OCI parameters directly. Keep them as implementation references unless custom code explicitly consumes them.

Outbound runtime starters

`oci_outbound_core`

- 6 rules, all mandatory
- Description, quantity, unit, price, currency and vendor material number
- Use for a generic OCI cart return after the buyer confirms those field names

`oci_outbound_extended`

- 3 optional rules
- Additional cost, material group and long text
- An enrichment set rather than a complete standalone profile; keep required core rules in the final runtime mapping

The module always adds `NEW_ITEM-EXT_PRODUCT_ID` from the external product ID configured on the buyer profile. This is protected output and is not an editable mapping rule.

`NEW_ITEM-LEADTIME` and `NEW_ITEM-MANUFACTCODE` are available as custom mapping targets. Map them only when Magento has a maintained product attribute that matches the buyer's expected value. PunchOut does not invent a lead time or treat a Magento

manufacturer label as an SAP manufacturer code.

`sap_oci_v4_outbound_core`

- 7 rules, 6 mandatory
- Core OCI fields plus optional SAP material number

`sap_oci_v5_outbound_core`

- 8 rules, 6 mandatory
- Core fields plus optional material number and external quote ID

`sap_oci_v5_outbound_hierarchy`

- 8 rules, 6 mandatory
- Includes optional parent ID and material group for hierarchy-aware buyer configurations

`sap_oci_v5_outbound_secure`

- 8 rules, all mandatory
- Includes external quote and contract values
- Use only when all eight source values are maintained and the buyer rejects incomplete rows

`secure` means fail-fast required-field behavior in the mapping. It does not enable request signing, encryption or SAP certification.

Selection rule

Start with the smallest outbound set accepted by the buyer. Add optional rules deliberately. Moving from OCI 4 to OCI 5 is a buyer-system decision, not an automatic Magento upgrade.

cXML Core Templates

Use these fixtures when the buyer needs standard cXML-style fields without a platform-specific starter pack.

`cxml_inbound_posr_core`

This inbound reference contains eight `PunchOutSetupRequest` rules. Operation, sender domain, sender identity, buyer cookie and browser form-post URL are mandatory in the fixture; shared secret and selected-item IDs are optional.

The current setup controller parses these fields directly from cXML. It does not execute this inbound profile. Use the fixture as a field reference or as a base for custom inbound normalization code.

`cxml_outbound_core`

- 6 rules, 5 mandatory
- Quantity, supplier part ID, unit price, description and unit of measure are required
- Classification is optional

This is the normal starting point for a generic `PunchOutOrderMessage` runtime profile. Independently of the mapping, every emitted `ItemIn` carries a sequential `lineNumber`; a procurement platform can echo it as `ItemOut@lineNumber` in the later `OrderRequest` for exact returned-cart line matching.

`cxml_outbound_level2`

- 5 optional enrichment rules
- Supplier auxiliary ID, manufacturer name/part ID, classification domain and image URL extrinsic

This fixture is not complete by itself because every rule is optional. Add its relevant rules to a profile that also contains the required core cart fields.

Before go-live, confirm the buyer's DTD expectations, units, classification domain, currency and extrinsic names. The module generates cXML, but acceptance remains buyer-specific.

SAP Ariba Templates

These mappings provide an Ariba-oriented starting point. They are not proof of SAP Ariba certification and do not replace the buyer's catalog-format specification or integration test.

`ariba_cxml_inbound_posr_1_2_x`

The ten inbound reference rules cover setup operation, sender credentials, buyer cookie, browser form-post URL, selected-item IDs, user email and unique-name extrinsics. Five are marked mandatory.

The current core setup controller parses its supported cXML fields directly and does not run this inbound template. Use it as a field catalog for project-specific normalization.

`ariba_cxml_outbound_core`

- 9 rules, 6 mandatory
- Core item data plus an Ariba-oriented classification domain field set
- Recommended first outbound copy when the buyer wants a basic Ariba-style order message

`ariba_cxml_outbound_level2`

- 12 rules, 6 mandatory
- Adds manufacturer name, manufacturer part ID and `AribaBuyerId` extrinsic fields
- Use only when the buyer requests those fields and Magento has reliable source data

`ariba_cxml_outbound_level2_strict`

- The same 12 target fields, all mandatory
- Any missing mapped value stops transfer validation

Begin with core or non-strict level 2. Move to the strict variant only after sample carts prove that every product and customer supplies all twelve values.

Coupa Templates

The Coupa fixtures name the cXML document versions used when they were created. Confirm the versions and mandatory fields with the buyer before treating them as a runtime contract.

`coupa_cxml_inbound_posr_1_2_014`

- 6 rules, 5 mandatory
- Setup operation, sender domain/identity, buyer cookie and browser form-post URL are mandatory; selected supplier part ID is optional

This is an inbound reference fixture. The current cXML setup controller parses supported values directly and does not apply the template as a normalization profile.

`coupa_cxml_outbound_poom_1_2_023`

- 10 rules, 6 mandatory
- Core order-message item fields plus optional supplier PunchOut URL and related enrichments
- Normal starting copy for a buyer using the corresponding Coupa-oriented field contract

`coupa_cxml_outbound_poom_1_2_023_strict`

- Same 10 target fields, all mandatory
- Use only after every mapped value is populated for every transferable cart line

The version in the template code describes the fixture. It does not force a DTD declaration or guarantee that a current Coupa tenant accepts the message unchanged.

Oracle Procurement Templates

Use these fixtures as a starting point for an Oracle Procurement buyer that has supplied its cXML PunchOut requirements.

`oracle_cxml_inbound_posr_1b`

The nine inbound reference rules cover operation, sender credentials, buyer cookie, browser form-post URL, selected-item IDs and a `User` extrinsic. Five are marked mandatory.

The core setup controller does not execute this inbound template. It parses its supported cXML fields directly; custom normalization must be implemented separately if the project needs it.

`oracle_cxml_outbound_1b_core`

- 10 rules, 6 mandatory
- Core cart-line fields plus optional `OracleBuyerId` and related fields
- Recommended first runtime copy for an Oracle-oriented buyer

`oracle_cxml_outbound_1b_core_strict`

- Same 10 fields, all mandatory
- Missing optional Magento data becomes a transfer error

Verify the buyer ID extrinsic, classification, unit and version expectations with the actual Oracle tenant. The template name is an implementation aid, not certification.

Template Selection Guide

Choose the template from the cart document the buyer expects, not from the procurement brand alone.

1. Choose protocol

- Use OCI when the buyer opens the shop with OCI parameters and expects indexed `NEW_ITEM-*` form fields back.
- Use cXML when the buyer sends a `PunchOutSetupRequest` and expects a `PunchOutOrderMessage` back.

2. Choose an outbound starting point

Buyer contract	Start with	Add or change when
Generic OCI	`oci_outbound_core`	Copy selected rules from `oci_outbound_extended` for extra fields.
SAP OCI 4	`sap_oci_v4_outbound_core`	The buyer explicitly accepts a different field/version set.
SAP OCI 5	`sap_oci_v5_outbound_core`	Use hierarchy fields only when requested.
Generic cXML	`cxml_outbound_core`	Add selected level-2 enrichment rules when requested.
SAP Ariba-oriented cXML	`ariba_cxml_outbound_core`	Move to level 2 for buyer-required manufacturer/extrinsic data.
Coupa-oriented cXML	`coupa_cxml_outbound_poom_1_2_023`	Change versions or fields to match the buyer's current specification.
JAGGAER-oriented cXML	`jaggaer_cxml_outbound_core`	Confirm customer-required classification/manufacturer fields and BuyerCookie compatibility.
Oracle-oriented cXML	`oracle_cxml_outbound_1b_core`	Add or remove extrinsics according to the tenant contract.

Do not assign an inbound template to a buyer. Inbound fixtures are catalog/reference records in the current core and are not executed by the setup controllers.

3. Decide whether strict is appropriate

A strict template marks every supplied rule mandatory. It is useful only when the buyer rejects incomplete rows and Magento reliably contains every value.

Strict mapping does not mean stricter authentication. Request signing, IP restrictions and transport authentication are configured separately.

4. Validate the copied profile

1. Copy the template into a new outbound profile.
2. Remove fields the buyer does not accept.
3. Map required fields to maintained Magento data.
4. Test products with missing optional attributes, decimal quantities and buyer-specific prices.
5. Inspect the generated payload and have the procurement system accept it.
6. Keep the runtime copy stable; do not edit the synchronized template for project-specific behavior.

See the Templates Overview for the current template catalog.

JAGGAER cXML Setup

Use `jaggaer_cxml_outbound_core` as an outbound mapping starter for a JAGGAER buyer. It copies a small set of cart fields into an editable runtime profile. It is not JAGGAER certification or a replacement for the customer's integration specification and acceptance tests.

The template must be present in the installed PunchOut package. Magento synchronizes the reusable mapping templates during the normal `bin/magento setup:upgrade` deployment step. Updating a reusable template does not rewrite existing customer runtime mappings.

Confirm the buyer's contract

Ask the procurement team for its JAGGAER product, current integration specification, test account, required fields and document endpoints. Confirm the cXML version separately for cart return, purchase orders, confirmations, shipments and invoices. The starter's metadata references the documented 1.1.010 PunchOut baseline; it does not force every transaction to that version.

JAGGAER publishes a Supplier Integration Specification through its Supplier Reference Materials. The publicly linked specification is dated July 2023, product version 23.2. Confirm whether the customer's current contract differs.

Create the runtime mapping

1. Open **Punchout > Punchout Mappings** and add a profile.
2. Select the buyer's Store View, protocol **cXML**, direction **Outbound**, and a unique profile code.
3. Leave **Reusable Template Profile** disabled.
4. In **Start From Template**, select **JAGGAER cXML Outbound Core**.
5. Review the copied rules, adapt customer-required fields, and save the active profile.

Returned field	Source and behavior
Quantity	Actual cart quantity, formatted to three decimal places
Supplier part ID	Cart SKU; required
Unit price	Actual cart price, formatted to two decimal places; an explicit zero remains zero
Currency	Resolved PunchOut context currency; no EUR or USD substitution
Description	Cart description; required

Returned field	Source and behavior
Unit of measure	Cart unit with the configured unit mapping; no automatic `EA` replacement for missing data
Classification	Optional product attribute `unspsc`
Classification domain	Context classification domain, defaulting to `UNSPSC`

If the buyer requires UNSPSC, select the actual maintained product attribute, verify its values and make the classification rule mandatory. Add manufacturer name, manufacturer part number and other customer-required fields only from maintained sources. Do not substitute the SKU for a manufacturer's part number without an explicit business agreement.

The protocol adapter supplies line numbers and line-specific `SupplierPartAuxiliaryID` values. Keep those identifiers intact in the procurement return and subsequent purchase order. They distinguish configurations that share the same SKU.

Configure the buyer

Create an active cXML Buyer Profile for the intended Store View. Configure its sender domain, sender identity and shared secret, choose the customer-resolution mode, and assign the copied outbound runtime mapping. Review product visibility and negotiated prices under that exact Magento customer.

The shared setup endpoint is `/punchout/cxml/setup`. cXML sender credentials identify the buyer; the buyer-specific browser return URL comes from the authenticated setup request. Confirmation, shipment and invoice endpoints are separate settings. See Buyer and Mapping Profiles and Endpoint Reference.

BuyerCookie compatibility checkpoint

The JAGGAER specification describes a BuyerCookie that can be constant across transactions. Punchout 1.2.3 stores returned carts by their internal PunchOut session, so independent sessions can retain the same BuyerCookie. Earlier versions require an upgrade for this workflow.

Do not alter the received BuyerCookie to manufacture uniqueness. Echo the original value. For later purchase orders, preserve each line's SupplierPartAuxiliaryID exactly as supplied by Magento. An exact identifier selects the retained original cart within the authenticated buyer, store, customer and currency context, including when the PO contains no BuyerCookie. If no exact identifier matches, the missing-snapshot policy applies; neither a unique remaining BuyerCookie nor a matching SKU can select another cart. Unknown explicit line identifiers cannot inherit a different line's original price or options. Conflicting references to several retained carts are rejected; a single PO combining lines from several returned carts is not supported by the current one-cart-per-

order model.

Validate the complete workflow

Test setup, customer activation, shopping and browser cart return with the copied mapping. Inspect the actual received cXML, not only Magento's successful HTTP response. Compare quantities, units, currency, configured product options, negotiated prices and computed totals.

Also test two sessions with the same BuyerCookie, two configurations of the same SKU, Edit/Inspect, changed quantities, and a purchase order submitted after the browser session expires. Verify the saved Magento order against the approved PO and retained cart data. Configure and test the intended inbound-order price policy.

Finally create a Magento shipment and invoice for the test order and inspect their actual cXML delivery and acknowledgements. These are separate document integrations; the cart mapping alone does not enable them. See Document Loop and Queue.

Record the accepted request/response samples and customer-specific changes using sanitized test data. Keep credentials, tokens and production identifiers out of screenshots and support exports.

OCI Flow

OCI uses request parameters to open the Magento shopping session and an HTML form to post the finished cart back to the procurement system.

1. Receive the login request

The buyer calls `/punchout/oci/login` with GET or POST. The minimum integration values are:

- buyer identity through `buyer_id` or `USER`;
- the OCI username and password expected by the buyer profile;
- return URL through `hook_url` or `HOOK_URL`.

Common context values such as `FUNCTION`, currency, language and SAP session identifiers can also be retained for mapping. The login controller parses supported request fields directly; an inbound mapping-template fixture is not executed during login.

The buyer profile defines whether it uses OCI 4 or OCI 5. The request version must match the profile. For product-specific functions, `PRODUCTID` is resolved through the profile's external product ID source: Magento SKU or a configured globally unique product attribute.

2. Authenticate and create context

Magento resolves the buyer and store, applies IP/signature checks, verifies protocol and optional transport credentials, validates the HTTPS return URL and creates an expiring token. The JSON response includes the token and correlation ID.

The integration then opens `/punchout/session/activate?token=<token>`. This binds the browser session and applies the profile's customer-resolution mode. The session can remain a guest, use a fixed Magento customer, resolve a customer from the request, or create a customer through JIT e-mail mapping.

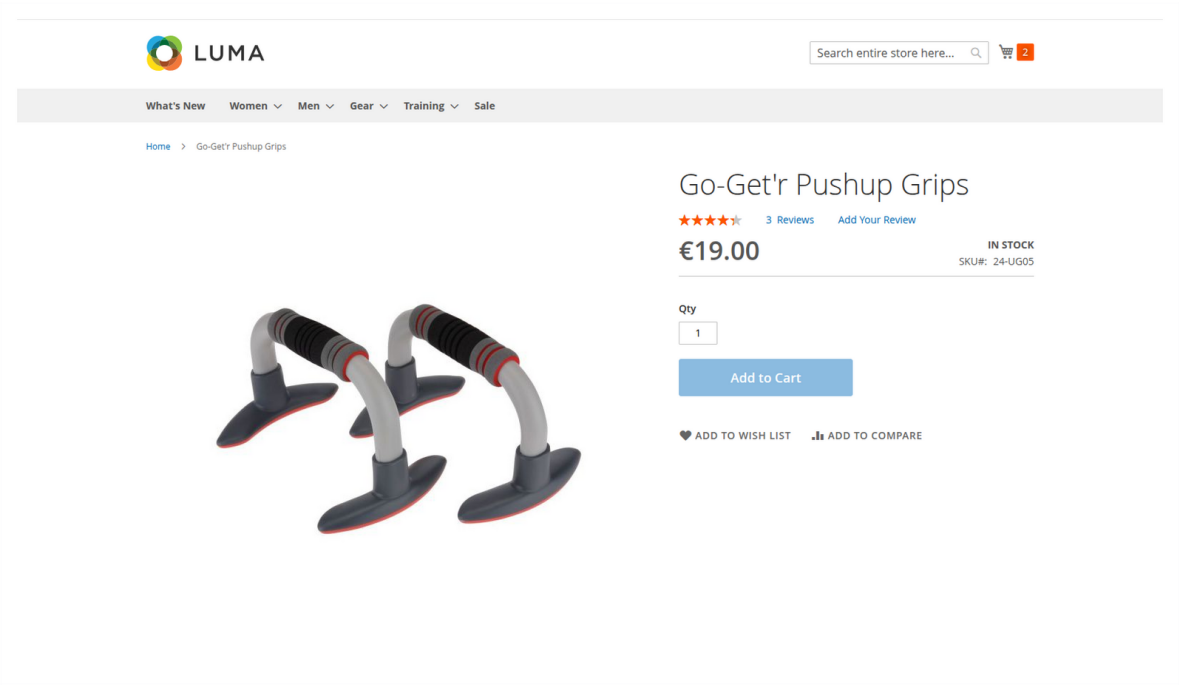
3. Select the OCI operation

The profile's OCI version and product-ID source are authoritative. `FUNCTION` does not switch the profile between OCI 4.0 and 5.0; it selects the supported storefront operation within that profile:

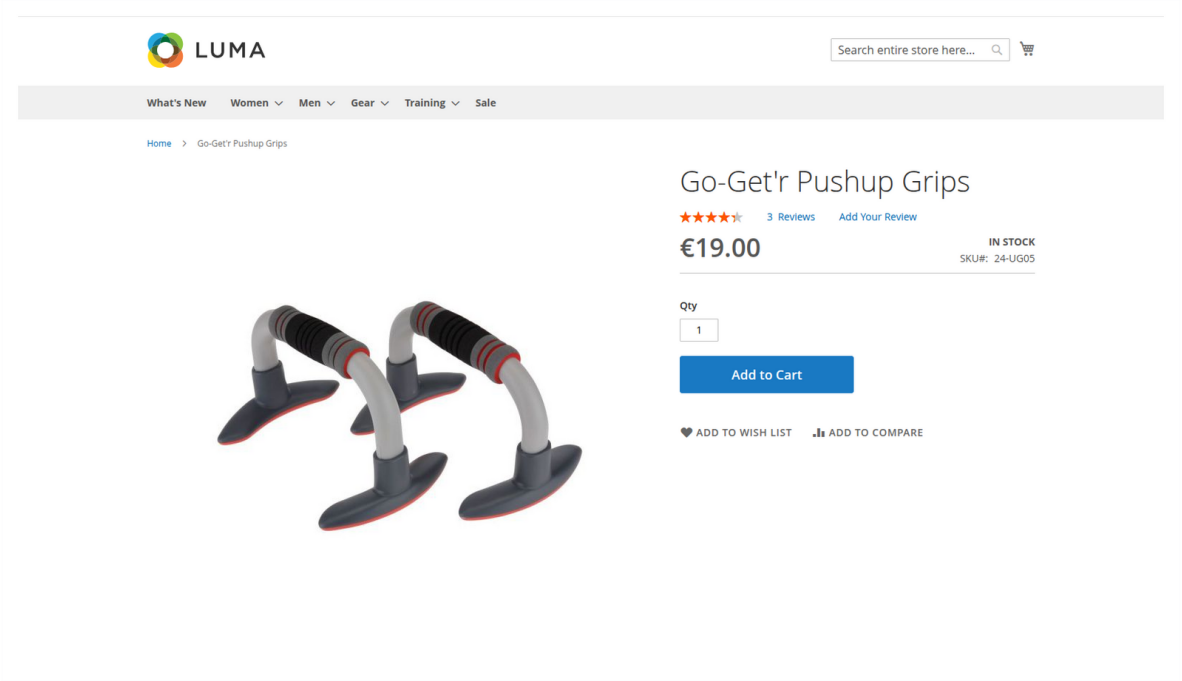
Request	Result	Cart/return behavior
omitted or `CREATE`	Normal catalog session	Isolated PunchOut quote; cart can be returned normally

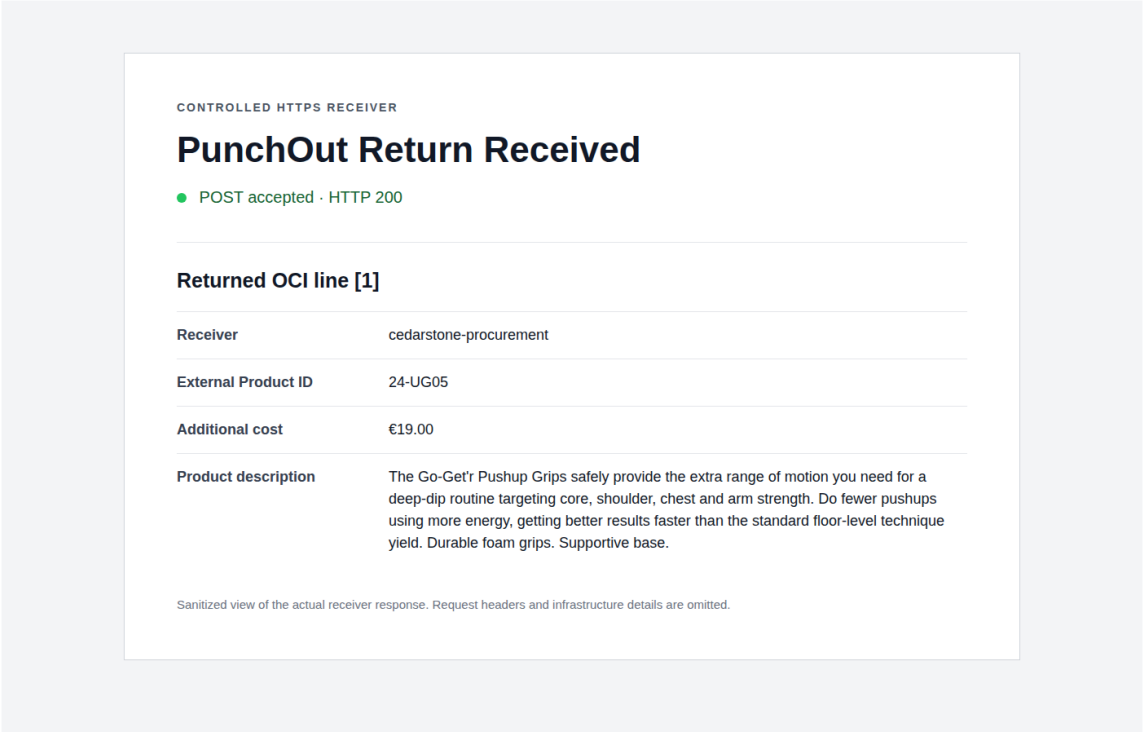
Request	Result	Cart/return behavior
`DETAIL` + `PRODUCTID`	Exact product page	Product-specific session; no normal cart-building return action
`DETAILADD` + `PRODUCTID` on OCI 5.0	Exact product page, then one-item return after a successful add	The selected line is returned immediately; grouped products are rejected
`QUANTITYCHECK` + `EXT_PRODUCT_ID` on OCI 5.0	JSON availability response from the login endpoint	No context, storefront session or cart is created
`VALIDATE`, `BACKGROUND_SEARCH` or `SOURCING`	Explicit unsupported-function response	Never treated as a normal shopping session

`PRODUCTID` is resolved as the configured Magento SKU or the selected globally unique product attribute. `NEW_ITEM-EXT_PRODUCT_ID` is then emitted from that same profile-owned identifier and remains protected from ordinary mapping overwrite.



DETAILADD uses a separate OCI 5.0 session. The resolved product exposes one enabled Add to Cart action. After that single add, Magento prepares the one-line OCI return and navigates the browser to the validated receiver.





`QUANTITYCHECK` works differently from every other function: the login endpoint answers it directly and never creates a context or storefront session. Send `FUNCTION=QUANTITYCHECK` with `EXT_PRODUCT_ID` and an optional `QUANTITY` on an OCI 5.0 Buyer Profile; `HOOK_URL` is not required. `EXT_PRODUCT_ID` is resolved through the profile's external product ID source. A successful JSON response contains:

Field	Value
`AVAILABLE_QUANTITY`	Salable quantity of the resolved product in the store's MSI stock; `0` for a disabled or unavailable product
`LEADTIME`	Whole days read from the product attribute selected as **OCI Quantity Check Lead Time Attribute** on the Buyer Profile; `0` when unset or empty

A supplied `QUANTITY` must be a positive number with up to three decimals. A missing or unresolvable `EXT_PRODUCT_ID`, an invalid `QUANTITY` or an invalid lead-time attribute returns `PUNCHOUT_OCI_QUANTITYCHECK_FAILED`; a Buyer Profile that is not OCI 5.0 returns `PUNCHOUT_OCI_QUANTITYCHECK_VERSION_NOT_SUPPORTED`.

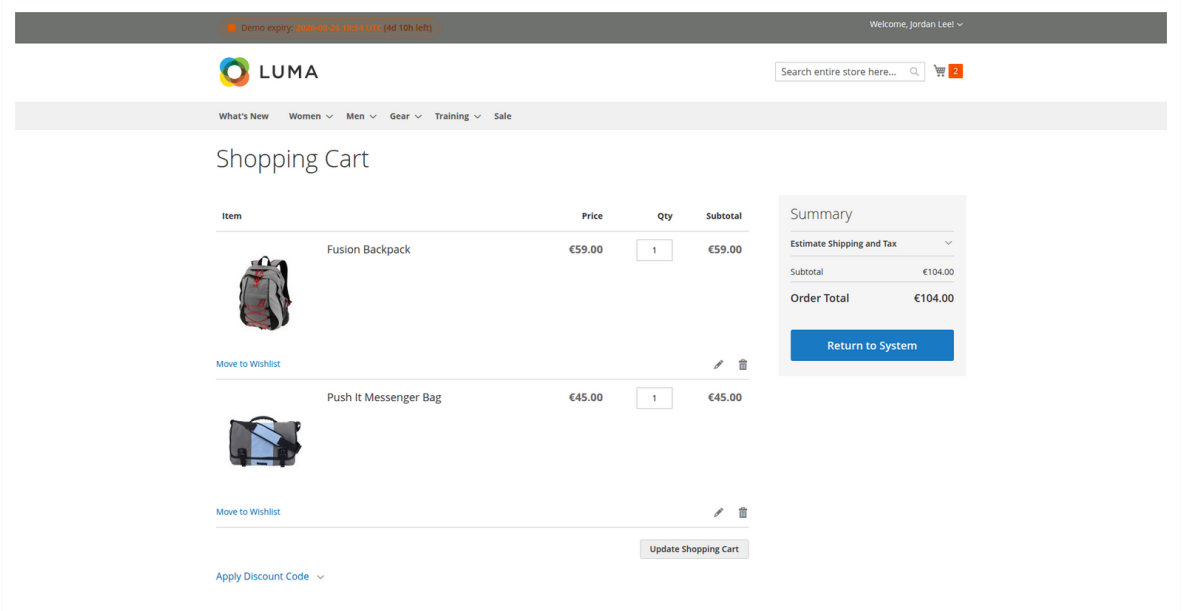
4. Build the cart

The buyer shops in Magento or uses the tokenized cart endpoints. Prices come from Magento unless inbound cart prices are explicitly enabled. The frontend operation policy applies the selected OCI function:

- a normal request opens the configured shopping flow;
- `DETAIL` opens the product resolved from `PRODUCTID`;
- OCI 5 `DETAILADD` permits one successful Add to Cart action and immediately returns that single line to the procurement system.

Unsupported OCI functions and version mismatches are rejected instead of being treated as a normal shopping request.

5. Return the cart



The storefront **Return to System** action calls `/punchout/session/back`. Magento attaches the active quote, applies the assigned outbound mapping and renders an auto-submit form to the validated hook URL.

Without a custom mapping, the core OCI payload includes indexed description, quantity, unit, price, currency and material-number fields. With a mapping profile, its target keys and required rules control the output. `NEW_ITEM-EXT_PRODUCT_ID` is protected system output and always uses the identifier configured on the buyer profile.

Direct API clients can call `POST /punchout/oci/transfer`; the HTML form is then returned as the `payload` value of a JSON response and is not posted automatically by that API call.

6. Handle retries

Send `X-Punchout-Request-Key` on direct transfers. Reuse the key only for the same semantic request. A changed request with the same key returns HTTP 409.

Start with an outbound template from OCI and SAP Templates, then validate the fields with the buyer's OCI specification.

cXML Flow

cXML PunchOut starts with a `PunchOutSetupRequest` and returns a `PunchOutOrderMessage`. SoftwareSilo parses the supported setup fields, opens a Magento session and maps the cart into the outbound message.

Before setup, a cXML 1.1 or 1.2 procurement platform can use the authenticated `/punchout/cxml/profile` endpoint to discover the Store-specific setup and OrderRequest URLs. See cXML Profile Discovery.

1. Receive `PunchOutSetupRequest`

Post cXML to `/punchout/cxml/setup`. The module reads:

- sender credential domain, identity and shared secret;
- `BuyerCookie`;
- setup operation;
- `BrowserFormPost/URL`;
- selected supplier part ID or auxiliary ID;
- setup extrinsics used by the level-2 resolver.

The XML security scanner rejects unsafe XML constructs and parsing uses no network access. The synchronized inbound template fixtures are reference/catalog records in the current core; they are not run as a transformation pipeline by this endpoint.

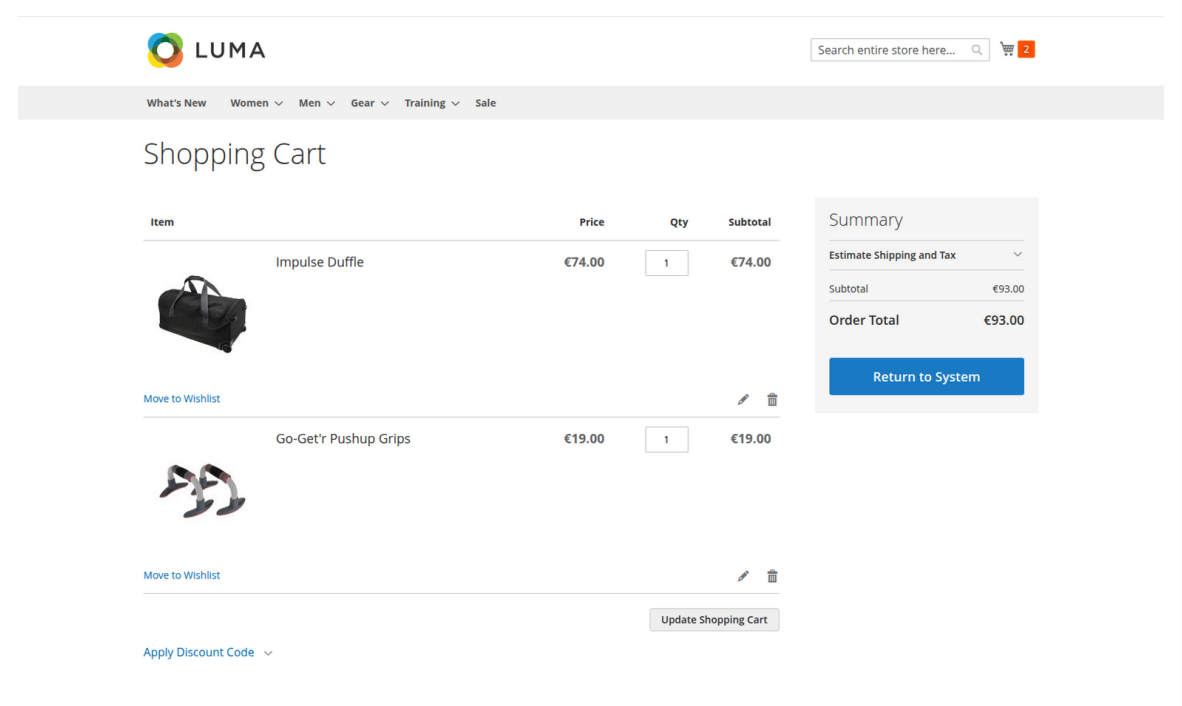
2. Resolve the buyer, operation and level-2 target

The sender identity selects the buyer profile. After security, authentication and return-URL checks, Magento creates a context token. When the request contains a selected item or a recognized deep-link extrinsic, the response includes level-2 metadata and a target URL hint. Otherwise the integration starts at the normal storefront target.

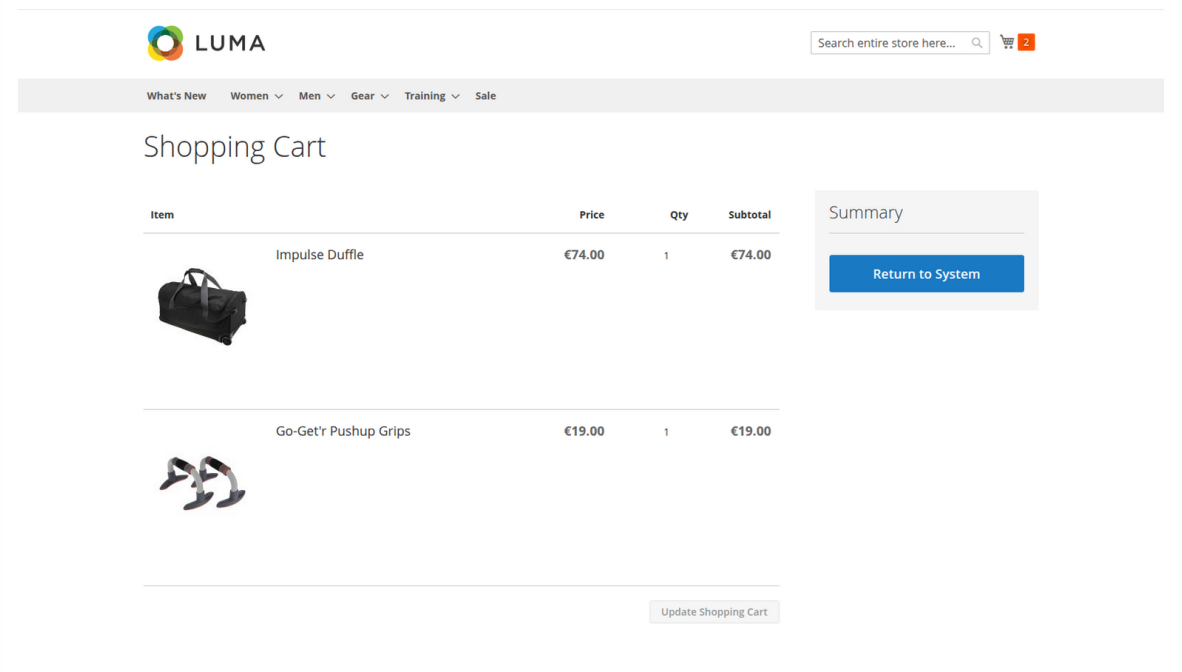
Open `/punchout/session/activate?token=<token>&target_url=<same-host-path>` to bind the browser and mapped customer.

The setup operation is preserved in the context and return message:

- `create` starts a new isolated quote;
- `edit` reconstructs the submitted `ItemOut` lines into a new isolated quote and recalculates them in Magento's resolved customer context;
- `inspect` reconstructs the lines for read-only review. The operation policy removes quantity, edit, remove, wishlist, coupon, shipping-estimate and checkout actions while keeping cart visibility and return support.

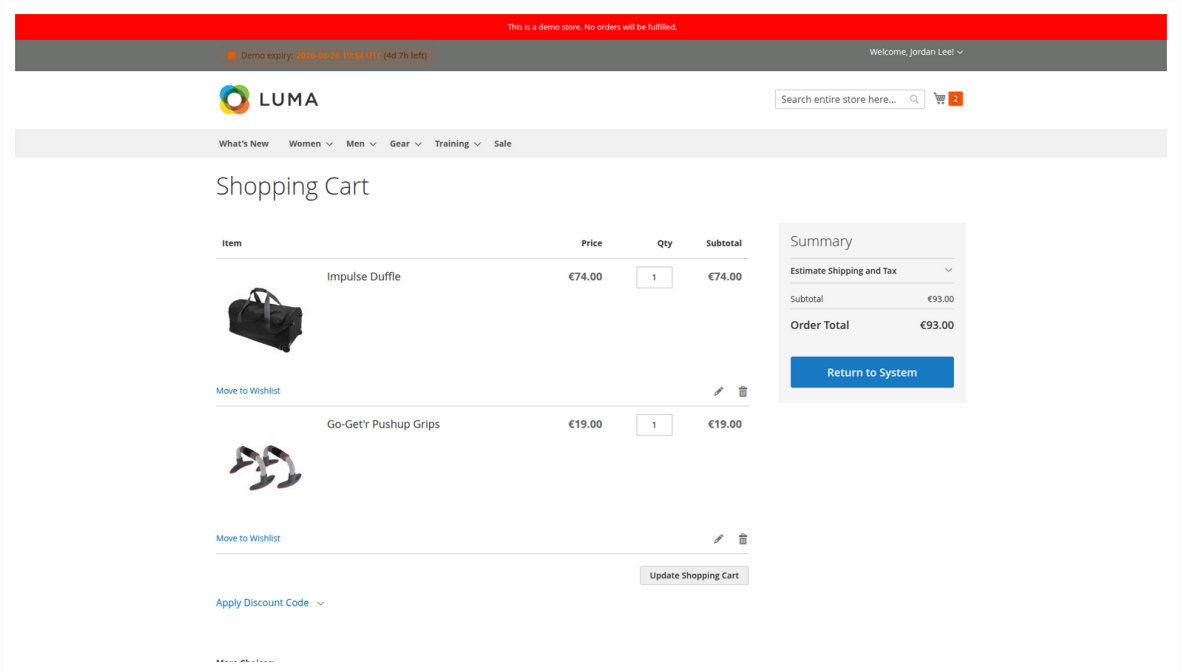


The corresponding inspect session keeps the reconstructed cart visible while removing mutation and checkout controls. Quantity values are read-only, and the only active cart action is the return to the procurement system.



The `SelectedItem`/`Extrinsic` Level-2 resolver first honors a valid explicit HTTP(S) target. Otherwise it resolves an exact `SupplierPartID` in the Store-view catalog; unknown identifiers fall back to catalog search. A `SupplierPartAuxiliaryID` is a returned-line identifier and is not treated as a Magento SKU.

3. Build and return the cart



The buyer shops in Magento or uses the cart endpoints. **Return to System** applies the active outbound mapping and builds a `PunchOutOrderMessage`. The browser posts it to `BrowserFormPost/URL` in a form field named `cXML-urlencoded`.

The generated message returns the procurement system's original opaque `BuyerCookie`. The internal Magento token appears only in the `StartPage/URL`. The message preserves `create`, `edit` or `inspect` as `operationAllowed`, rejects mixed item currencies and calculates the total from item quantity and price.

Direct integrations may call `POST /punchout/cxml/transfer`; that route returns the XML inside JSON and does not perform the browser form post.

When Magento returns the cart, it also stores an immutable cart snapshot. The snapshot links the BuyerCookie to the buyer profile, customer, Store view, currency, original quote and returned lines. An approved `OrderRequest` can use it later, even after the short-lived browser context has expired.

4. Process the approved OrderRequest

The procurement system posts the approved cXML order to:

```
POST /punchout/document_po/inbound
Content-Type: application/xml
```

`Header/Sender/Credential` authenticates the buyer. `OrderRequestHeader` supplies the purchase-order identity and operation, while `ItemOut` lines supply products, quantities and submitted prices. For snapshot-based pricing, include the original

BuyerCookie as an `Extrinsic` in `OrderRequestHeader`.

The effective processing mode decides the result:

- **Acknowledge Only** stores and acknowledges the request without building a quote.
- **Validate Only** resolves the customer, products, addresses, shipping, payment and totals, but does not submit the quote.
- **Create Magento Order** runs the same validation and places a Magento sales order.

Use Inbound Orders and Pricing for returned-cart matching, price decisions and manual review.

5. Optional outbound document loop

Confirmation, shipment and invoice delivery are separate from `/punchout/cxml/transfer`. Read Document Loop and Queue for their prerequisites and retry behavior.

Use the Template Selection Guide to choose an outbound starter mapping for generic cXML, SAP Ariba, Coupa or Oracle Procurement.

For version-specific behavior, read Protocol Compatibility.

Add-Ons

The base PunchOut module handles OCI and cXML sessions, cart transfer, inbound orders, mappings, traces and outbound business documents. The add-ons cover two separate needs: embedding the Magento storefront inside a procurement application and installing a complete set of demonstration data.

Add-on	Install it when you need	Composer package
Iframe Compatibility	The procurement system opens the Magento supplier shop inside an iframe	`mageb2b/punchout-iframe`
Sample Data	A local, demo or staging system needs a connected OCI and cXML example workflow	`mageb2b/punchout-sample-data`

Neither package replaces the base module. Install only the add-on required by the project. Sample Data must never be installed in a production shop.

Iframe Compatibility Add-On

Some procurement platforms open the supplier shop inside an iframe instead of a separate browser tab. Browsers and Magento normally restrict this because an embedded shop needs cross-site framing and cookies. The Iframe Compatibility add-on applies the required response and cookie settings only to an authenticated PunchOut context from an explicitly allowed procurement origin.

Normal PunchOut sessions that open Magento as a top-level page do not need this package.

- Composer package: `mageb2b/punchout-iframe`
- Magento module: `MageB2B_PunchoutIframe`
- Required base module: `MageB2B_Punchout`

Install the package

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

php bin/magento module:enable MageB2B_PunchoutIframe
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento cache:flush
```

Use the deployment commands required by the current Magento mode and storefront locales.

Uninstalling the package removes the `punchout_iframe_buyer_config` table with all buyer-level origin overrides and the `punchout/iframe/` configuration values. See Installation for the base module's destructive uninstall scope.

Configure the Store View

Open **Punchout > Configuration > Embedded PunchOut** in Magento Admin.

1. Select the Store View used by the PunchOut buyer.
2. Enable **Iframe Compatibility**.
3. Enter one exact HTTPS origin per line under **Allowed Procurement Origins**.
4. Save the configuration and clear the configuration cache.

An origin contains the scheme, host and optional non-standard port, but no path:

```
https://procurement.example.com
```

```
https://purchasing.example.net:8443
```

HTTP origins, wildcard hosts, credentials, query strings and fragments are rejected. `https://procurement.example.com/app` is not an origin and must not be entered. The add-on normalizes host names, removes duplicate entries and omits the standard HTTPS port.

Override settings for one buyer

Open **Punchout > Punchout Buyers**, edit the buyer profile and expand **Embedded PunchOut**.

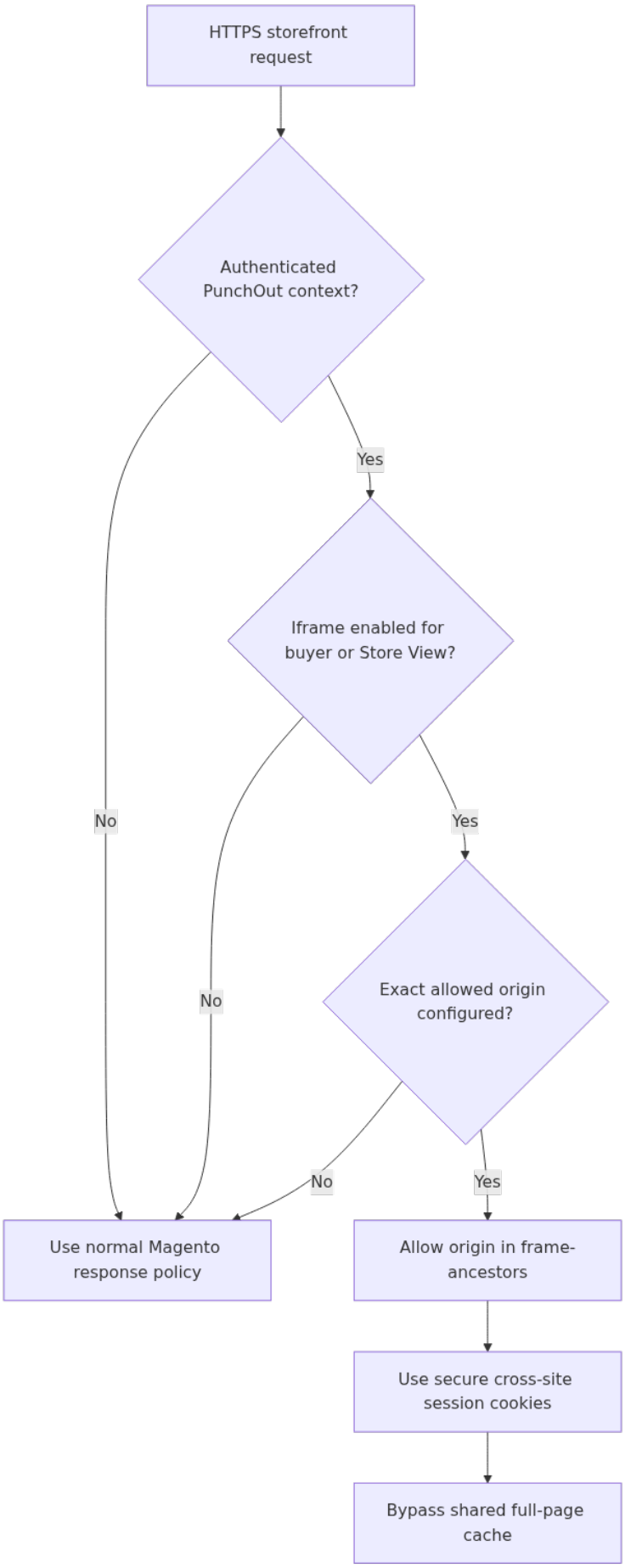
The **Iframe Compatibility** field can inherit the Store View value or explicitly enable or disable embedding for this buyer. **Allowed Procurement Origins** can likewise replace the Store View list for this buyer. Leave the buyer-specific origin field empty to inherit the Store View origins.

Use an override when buyers on the same Store View use different procurement hosts. The buyer profile is the more specific setting and therefore takes precedence over Store View configuration.

Buyer setting	Effective result
Inherit	Use the Store View enablement and origin list
Enabled with origins	Allow only the buyer-specific HTTPS origins
Disabled	Keep this buyer in the normal top-level PunchOut flow

What happens during an embedded session

The add-on does not make every storefront page embeddable. It first confirms that the request uses HTTPS and belongs to a current, authenticated PunchOut context. That context must identify a valid buyer profile and an allowed procurement origin.


HTTPS storefront request
Authenticated PunchOut context?
No
Yes
Iframe enabled for buyer or Store View?
No
Yes
Exact allowed origin configured?
No
Yes
Use normal Magento response policy
Allow origin in frame-ancestors
Use secure cross-site session cookies
Bypass shared full-page cache

For an eligible request, the response allows the configured origin through the Content Security Policy `frame-ancestors` directive. Magento's normal `X-Frame-Options: SAMEORIGIN` header is suppressed for that request. Session and storefront cookies receive the secure cross-site cookie attributes required by modern browsers.

The request also bypasses shared full-page cache. This prevents a response containing one buyer's framing policy from being served to another visitor. Requests outside an eligible PunchOut context retain Magento's normal headers, cookies and cache behavior.

The package does not disable Magento CSP, form keys or customer-session checks.

Configure the procurement platform too

Both applications must permit the embedded relationship:

- Magento must list the procurement application's exact origin.
- The procurement application must allow the Magento storefront in its own iframe or `frame-src` policy.
- The storefront and procurement pages must use HTTPS.
- The browser must accept the cross-site session cookies used by the embedded shop.

Some managed browsers block third-party cookies regardless of the server configuration. If the customer cannot change that policy, configure the procurement system to open the supplier shop in a top-level tab instead.

Session lifecycle

After successful PunchOut activation, the add-on stores a short-lived secure marker for the active context. Its lifetime follows the PunchOut context. A successful cart return removes the marker. Logout and expiry cleanup remove it once the context is no longer usable.

This marker does not authenticate a buyer on its own. The server still validates the underlying PunchOut context for each eligible request.

Troubleshooting

The browser refuses to display Magento in the iframe

Check the response headers in the browser developer tools. Confirm that the request belongs to the active PunchOut session, Iframe Compatibility is enabled at the effective scope, and the parent page's exact HTTPS origin is configured. Also inspect the procurement application's framing policy.

The login or cart is lost inside the iframe

Confirm that every page uses HTTPS and inspect whether the browser accepted the Magento session cookie with secure cross-site attributes. Browser or corporate third-

party-cookie restrictions can still block the session. Test the same PunchOut link in a top-level tab to separate cookie policy from buyer-profile or mapping problems.

The setting works for one buyer but not another

Compare both buyer profiles. An explicit buyer setting overrides the Store View value, including an explicit disabled state or a buyer-specific origin list. Then confirm that each context resolves to the expected buyer profile and Store View.

The regular storefront headers changed

The special response policy must apply only to an eligible embedded PunchOut context. Check whether the request still carries a valid PunchOut token or marker. Clear stale browser data after ending the session, then verify an unrelated storefront request again.

Related guides

- Authentication and Security
- Cart API and Session Lifecycle
- Storefront Operation Policy
- Common Issues

Sample Data Add-On

The Sample Data add-on builds a connected PunchOut example in Magento. It provides OCI and cXML buyer profiles, customers, products, returned-cart data, inbound purchase orders and completed outbound documents. This is useful for demonstrations, administrator training and integration checks where an empty installation would hide how the records relate to each other.

The package contains fixed public sample credentials. Install it only in a local, demo or staging environment. Never install it in production.

- Composer package: `mageb2b/punchout-sample-data`
- Magento module: `MageB2B_PunchoutSampleData`
- Required base module: `MageB2B_Punchout`

Install the package

```
composer config bearer.repo.softwaresilo.io <token>
composer config repositories.softwaresilo composer
https://repo.softwaresilo.io/
composer require mageb2b/punchout-sample-data:*

php bin/magento module:enable MageB2B_PunchoutSampleData
php bin/magento setup:upgrade
php bin/magento setup:di:compile
php bin/magento cache:clean config
```

The data patch runs during `setup:upgrade`. A default Store View must exist before it runs.

What the fixture set contains

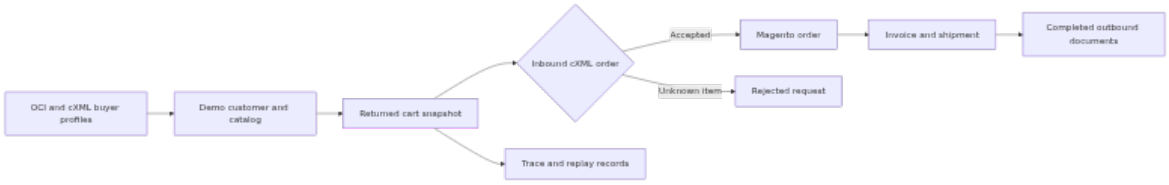
The add-on creates a small demo catalog with an office chair, a docking station, a laptop stand and safety gloves. It also creates a procurement customer with billing and shipping addresses.

Three buyer profiles exercise different identity and order paths:

Profile	Purpose
OCI	Authenticated OCI shopping session and cart return
cXML with fixed customer	cXML setup, cart return and inbound order for the customer assigned to the buyer profile

Profile	Purpose
cXML with customer mapping	Resolves the customer from the end-user email and can create the customer when the buyer profile permits it

The fixture graph includes both a successful and a rejected cXML `OrderRequest`. The successful request is connected to a Magento order, invoice and shipment. The rejected request contains an unknown supplier part ID and therefore has no Magento order. Completed replay protection, trace entries and outbound invoice and shipment queue records are also included.



The add-on enables PunchOut sandbox mode at Default Config only when no stored default value exists. It also supplies sample supplier credentials and a flat-rate shipping setup. Existing Default Config values are not overwritten, and more specific website or Store View configuration continues to take precedence.

Where to inspect the data

The add-on has no separate Admin page. Use the standard Magento and PunchOut areas:

- **Punchout > Punchout Buyers** for the OCI and cXML profiles
- **Punchout > Punchout Trace** for setup, transfer, inbound-order and rejection stages
- **Punchout > Inbound Orders** for the accepted and rejected purchase orders
- **Punchout > Document Queue** for completed invoice and shipment documents
- **Customers > All Customers** for the procurement customer and addresses
- **Catalog > Products** for the demo products
- **Sales > Orders** for the connected order, invoice and shipment

This separation is intentional. The sample records exercise the same screens and relations used by a real implementation.

Verify the installation

Run the supplied verifier after installation:

```
php bin/magento sampledata:verify:punchout
```

It checks the customer, addresses, category, products, buyer profiles, mappings and settings. It also follows the lifecycle from the transferred context and returned cart through accepted and rejected inbound orders, Magento sales records, replay protection,

trace stages and the two completed document-queue entries.

A non-zero exit code means the fixture set is incomplete. Read the reported item before resetting anything. A missing base product or buyer profile can make several later lifecycle checks fail at once.

Repeated installation

The installer uses stable sample identifiers and can run more than once without duplicating the fixture graph. Existing sample buyer profiles are updated. Existing customers, products and categories with the sample identifiers are reused or skipped as appropriate.

The standard mapping templates are shared module data. They are synchronized before the PunchOut fixtures are installed and are not treated as disposable sample records.

Reset the fixtures

Create a database backup first if anyone has edited the sample records or attached other work to them. Then run:

```
php bin/magento sampledata:reset:punchout
```

The command asks for confirmation and defaults to **No**. It targets the known sample customer, catalog records, buyer profiles and lifecycle identifiers. Unrelated PunchOut data and shared mapping templates remain in place. Sample configuration is removed only while its current value still matches the value installed by the package.

To install the fixtures again:

```
php bin/magento setup:upgrade
php bin/magento sampledata:verify:punchout
```

The reset can be repeated safely when part of the sample set is already missing.

Troubleshooting

Verification reports several missing lifecycle records

Start with the first missing customer, product, buyer profile or returned-cart record. Later order, invoice, shipment and queue records depend on those earlier fixtures. Review the preceding `setup:upgrade` output and Magento logs before running reset.

The sample configuration differs from this guide

The installer does not replace existing Default Config values, and website or Store View values can override them. Check the effective configuration scope before assuming the

fixture is incomplete.

Reset leaves a configuration value in place

This protects later administrator changes. The reset removes a sample setting only when the current stored value is still exactly the installed sample value. Review and remove a deliberately changed setting manually if the environment no longer needs it.

The rejected order has no Magento order

That is expected. It intentionally references an unknown supplier part ID so the error path, trace and inbound-order record can be inspected without creating a sales order.

Related guides

- Trace and Simulator
- Inbound Orders and Pricing
- Document Loop and Queue
- Common Issues

Configuration Fields

This reference lists the Magento Admin settings exposed by the base package and its documented add-ons: `mageb2b/punchout`, `mageb2b/punchout-iframe`.

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.

`mageb2b/punchout`

General

Field	Configuration path	Input	Scope
Enable Punchout	`punchout/general/enabled`	Select	Default, Website, Store view
Punchout return action label	`punchout/general/return_action_label`	Text	Default, Website, Store view
Enable Sandbox Callback Mode	`punchout/general/sandbox_mode`	Select	Default, Website, Store view

Security

Field	Configuration path	Input	Scope
Require Strict Signature Validation	`punchout/security/strict_signature`	Select	Default, Website, Store view
Signature Secret	`punchout/security/signature_secret`	Protected text	Default, Website, Store view
Allowed IP Ranges	`punchout/security/allowed_ips`	Text area	Default, Website, Store view
Allow OCI request customer override	`punchout/security/allow_request_customer_id_oci`	Select	Default, Website, Store view

Field	Configuration path	Input	Scope
Allow cXML request customer override	`punchout/security/allow_request_customer_id_cxml`	Select	Default, Website, Store view
Allow insecure outbound TLS in sandbox mode	`punchout/security/allow_insecure_outbound_tls`	Select	Default, Website, Store view

Visibility dependencies:

- `punchout/security/allow_insecure_outbound_tls` is available when `punchout/general/sandbox_mode` is `1`.

Customer Identity

Field	Configuration path	Input	Scope
Enable JIT Customer Provisioning	`punchout/identity/jit_enabled`	Select	Default, Website, Store view
JIT Customer Group	`punchout/identity/jit_customer_group_id`	Select	Default, Website, Store view

Visibility dependencies:

- `punchout/identity/jit_customer_group_id` is available when `punchout/identity/jit_enabled` is `1`.

Cart

Field	Configuration path	Input	Scope
Allow inbound cart item prices	`punchout/cart/allow_inbound_item_prices`	Select	Default, Website, Store view

cXML Profile Discovery

Field	Configuration path	Input	Scope
Enable Profile Discovery	`punchout/profile/enabled`	Select	Default, Website, Store view
Public PunchOut Base URL	`punchout/profile/public_base_url`	Text	Default, Website, Store view

Visibility dependencies:

- `punchout/profile/public_base_url` is available when `punchout/profile/enabled` is `1`.

Inbound Orders

Field	Configuration path	Input	Scope
Processing Mode	`punchout/inbound_order/processing_mode`	Select	Default, Website, Store view
Authoritative Price Source	`punchout/inbound_order/price_source`	Select	Default, Website, Store view
Returned Cart Line Matching	`punchout/inbound_order/line_match_strategy`	Select	Default, Website, Store view
Changed Quantity Action	`punchout/inbound_order/changed_quantity_action`	Select	Default, Website, Store view
New Item Action	`punchout/inbound_order/new_item_action`	Select	Default, Website, Store view
Removed Item Action	`punchout/inbound_order/removed_item_action`	Select	Default, Website, Store view
Missing/Expired Snapshot Action	`punchout/inbound_order/missing_snapshot_action`	Select	Default, Website, Store view
Price Mismatch Action	`punchout/inbound_order/price_mismatch_action`	Select	Default, Website, Store view
Absolute Unit Price Tolerance	`punchout/inbound_order/price_tolerance_absolute`	Text	Default, Website, Store view

Field	Configuration path	Input	Scope
Percent Unit Price Tolerance	`punchout/inbound_order/price_tolerance_percent`	Text	Default, Website, Store view
Customer Resolution	`punchout/inbound_order/customer_resolution`	Select	Default, Website, Store view
Product Resolution	`punchout/inbound_order/product_resolution`	Select	Default, Website, Store view
Address Source	`punchout/inbound_order/address_source`	Select	Default, Website, Store view
Shipping Method	`punchout/inbound_order/shipping_method`	Select	Default, Website, Store view
Payment Method	`punchout/inbound_order/payment_method`	Select	Default, Website, Store view
Order Update Policy	`punchout/inbound_order/update_policy`	Select	Default, Website, Store view
Order Cancellation Policy	`punchout/inbound_order/cancellation_policy`	Select	Default, Website, Store view
Send Customer Order Email	`punchout/inbound_order/send_customer_email`	Select	Default, Website, Store view
Send cXML ConfirmationRequest	`punchout/inbound_order/confirmation_enabled`	Select	Default, Website, Store view
Confirmation Mode	`punchout/inbound_order/confirmation_mode`	Select	Default, Website, Store view
Default Confirmation URL	`punchout/inbound_order/confirmation_url`	Text	Default, Website, Store view
ASN URL	`punchout/inbound_order/asn_url`	Text	Default, Website, Store view

Field	Configuration path	Input	Scope
Invoice URL	`punchout/inbound_order/invoice_url`	Text	Default, Website, Store view
Supplier Credential Domain	`punchout/inbound_order/supplier_domain`	Text	Default, Website, Store view
Supplier Identity	`punchout/inbound_order/supplier_identity`	Text	Default, Website, Store view
Supplier Shared Secret	`punchout/inbound_order/supplier_shared_secret`	Protected text	Default, Website, Store view
Maximum Processing Attempts	`punchout/inbound_order/max_attempts`	Text	Default, Website, Store view
Retry Batch Size	`punchout/inbound_order/batch_size`	Text	Default
Retry Delay (minutes)	`punchout/inbound_order/retry_delay_minutes`	Text	Default, Website, Store view
Outbound cXML Response Mode	`punchout/inbound_order/response_mode`	Select	Default, Website, Store view
Inbound Order Payload Retention (days)	`punchout/inbound_order/retention_days`	Text	Default, Website, Store view
Returned Cart Snapshot Retention (days)	`punchout/inbound_order/snapshot_retention_days`	Text	Default, Website, Store view

Visibility dependencies:

- `punchout/inbound_order/confirmation_mode` is available when `punchout/inbound_order/confirmation_enabled` is `1`.
- `punchout/inbound_order/confirmation_url` is available when `punchout/inbound_order/confirmation_enabled` is `1`.

Trace

Field	Configuration path	Input	Scope
Enable Trace Logging	`punchout/trace/enabled`	Select	Default, Website, Store view

Field	Configuration path	Input	Scope
Trace Retention (days)	`punchout/trace/retention_days`	Text	Default, Website, Store view

Analytics

Field	Configuration path	Input	Scope
Enable PunchOut Analytics	`punchout/analytics/enabled`	Select	Default, Website, Store view

PunchOut Debugging

Field	Configuration path	Input	Scope
Enable Debug Logging	`punchout/debug/enabled`	Select	Default, Website, Store view

Session

Field	Configuration path	Input	Scope
Token Expiration (hours)	`punchout/session/token_expiration_hours`	Text	Default, Website, Store view

Document Queue

Field	Configuration path	Input	Scope
Enable Queue Processing	`punchout/queue/enabled`	Select	Default, Website, Store view
Batch Size per Cron Run	`punchout/queue/batch_size`	Text	Default, Website, Store view
Maximum Retry Attempts	`punchout/queue/max_retry`	Text	Default, Website, Store view
Retry Delay (minutes)	`punchout/queue/retry_delay_minutes`	Text	Default, Website, Store view

Field	Configuration path	Input	Scope
Completed Queue Retention (days)	`punchout/queue/retention_days`	Text	Default, Website, Store view

Retention

Field	Configuration path	Input	Scope
Request Replay Retention (days)	`punchout/request_replay/retention_days`	Text	Default, Website, Store view
Cleanup Batch Size	`punchout/retention/batch_size`	Text	Default

Operation Policy

Field	Configuration path	Input	Scope
Enable Frontend Operation Policy	`punchout/operation/policy_enabled`	Select	Default, Website, Store view
Restricted Action Landing Route	`punchout/operation/landing_route`	Text	Default, Website, Store view
Allowed Full Actions for CREATE	`punchout/operation/create_allowed_actions`	Text area	Default, Website, Store view
Allowed Full Actions for EDIT	`punchout/operation/edit_allowed_actions`	Text area	Default, Website, Store view
Allowed Full Actions for INSPECT	`punchout/operation/inspect_allowed_actions`	Text area	Default, Website, Store view
Allowed Full Actions for DETAIL	`punchout/operation/detail_allowed_actions`	Text area	Default, Website, Store view

Field	Configuration path	Input	Scope
Allowed Full Actions for DETAILADD	`punchout/operation/detail_add_allowed_actions`	Text area	Default, Website, Store view

`mageb2b/punchout-iframe`

Embedded PunchOut

Field	Configuration path	Input	Scope
Enable Iframe Compatibility	`punchout/iframe/enabled`	Select	Default, Website, Store view
Allowed Procurement Origins	`punchout/iframe/allowed_origins`	Text area	Default, Website, Store view

Visibility dependencies:

- `punchout/iframe/allowed_origins` is available when `punchout/iframe/enabled` is `1`.

Database Reference

PunchOut connects procurement identities and shopping sessions with standard Magento customers, quotes and orders. The relationships below focus on the records an implementation or ERP team is likely to encounter.

Buyers and mappings

Relationship	Why it matters
`punchout_buyer_config` belongs to one `store`	Selects the Store View used for the buyer's PunchOut sessions.
`punchout_buyer_config` can use one `customer_entity`	Supports a fixed Magento customer for a procurement identity.
`punchout_buyer_config` can use one `customer_group`	Provides the customer group used when just-in-time provisioning is enabled.
`punchout_buyer_config` can use one `punchout_mapping_profile`	Selects the outbound field mapping for the buyer and protocol.
`punchout_mapping_rule` belongs to one `punchout_mapping_profile`	Defines how Magento values are written into OCI or cXML fields.
`punchout_mapping_value_map` belongs to one `punchout_mapping_rule`	Translates known source values for a specific mapping rule.

Session and returned cart

Relationship	Why it matters
`punchout_session_context` belongs to one `store`	Keeps protocol handling within the Store View selected for the session.
`punchout_session_context` can identify one `customer_entity`	Connects the procurement user to the resolved Magento customer.
`punchout_session_context` can use one `quote`	Links the active PunchOut context to its Magento shopping cart.

Relationship	Why it matters
`punchout_returned_cart_snapshot` can belong to one `punchout_buyer_config`	Records which buyer received the returned cart.
`punchout_returned_cart_snapshot` can refer to one `quote`	Preserves the original cart context used for later order comparison.

Inbound orders and document loop

Relationship	Why it matters
`punchout_inbound_order` can belong to one `punchout_buyer_config`	Applies the correct buyer policies to an incoming order document.
`punchout_inbound_order` can identify one `customer_entity`	Records the Magento customer resolved for the order.
`punchout_inbound_order` can refer to Magento `quote` and `sales_order`	Connects the procurement document to the cart and order it created or updated.
`punchout_inbound_order` can refer to original and replacement `sales_order`	Preserves order history when the configured update policy creates a replacement.
`punchout_confirmation_line` belongs to one `punchout_inbound_order`	Stores line-level confirmation results for the procurement response.
`punchout_document_queue` can belong to one `punchout_session_context` or `punchout_inbound_order`	Keeps confirmations, ship notices and invoices tied to their business transaction.

Operations and add-on data

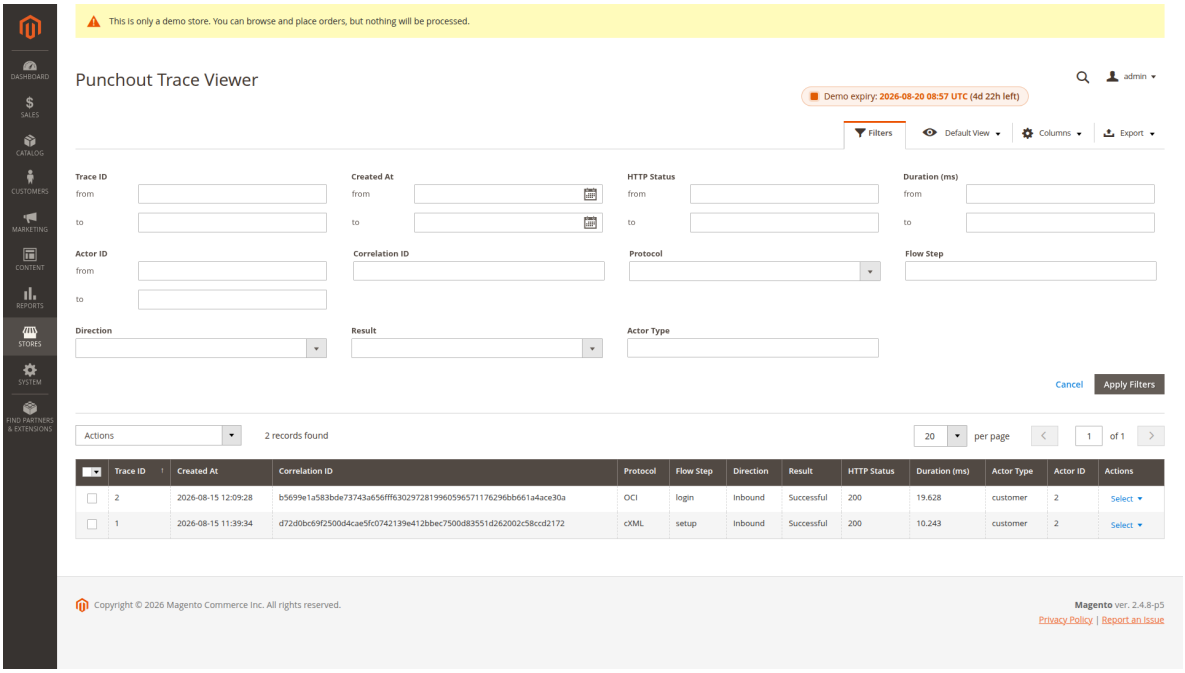
Relationship	Why it matters
`punchout_trace` can belong to one `punchout_session_context`	Groups diagnostic stages for the same PunchOut exchange.
PunchOut analytics records belong to one `store`	Keeps usage and item analytics Store View specific.

Relationship	Why it matters
`punchout_iframe_buyer_config` belongs to one `punchout_buyer_config`	Adds embedded-session policy to an existing buyer without duplicating its credentials or protocol settings.

Payload contents, credentials, replay data and internal queue fields are intentionally omitted. Use the PunchOut services and Admin screens so authentication, idempotency and order policies remain enforced.

Common Issues

Start with the correlation ID and the first failed trace step. Later errors are often consequences of an earlier buyer, security or context failure.



Buyer is not allowed

Check the inbound protocol and buyer identity character for character. Then confirm that an active profile exists in the requested store or as an **All Store Views** fallback. A store-specific match takes precedence over a global profile.

For OCI, compare `USER`/username and password separately from optional HTTP transport credentials. For cXML, compare sender domain, sender identity and shared secret.

Security validation fails

The top-level response is `PUNCHOUT_SECURITY_VALIDATION_FAILED`; the `errors` array gives the cause.

- For `PUNCHOUT_IP_NOT_ALLOWED`, confirm the real client IP seen by Magento and the configured IPv4/IPv6 range.
- For signature errors, confirm strict mode, secret and the exact body or normalized parameters used to calculate HMAC-SHA256.
- Do not enable strict mode simply because HTTPS is active; the caller must send the custom signature.

Return URL is rejected

Production URLs must use HTTPS, resolve successfully and resolve only to public IP addresses. Sandbox mode can permit localhost/private addresses, but it still requires HTTPS. Check both A and AAAA resolution when a hostname behaves differently between environments.

Storefront opens as the wrong customer

Check **Customer Resolution Mode** and its corresponding Magento Customer or identity source. Verify that the resolved customer belongs to the Website of the selected Store View unless global customer sharing is enabled. For a newly provisioned account, check whether Magento requires account confirmation.

Also check whether the request selected the intended store. Protocol-specific request customer overrides should normally remain disabled.

Cart price differs from the request

This is expected when **Allow inbound cart item prices** is No. Magento uses its catalog and customer pricing. If buyer-supplied prices are part of the approved integration contract, enable the setting only for the required scope and retest tax and currency totals.

Mapping fails or fields are empty

Confirm that the buyer has an active outbound, non-template profile with the same protocol and compatible store scope. In the trace, look for `PUNCHOUT_MAPPING_REQUIRED_*` errors and map that target to a source value maintained for every relevant product/customer.

An optional-enrichment template such as `cxml_outbound_level2` is not a complete replacement for the core required rules.

Return to System redirects back to the cart

The browser may have no active token, the context may be expired, the cart may be empty, the hook URL may now fail validation or mapping validation may have failed. Inspect the browser session and the corresponding trace rather than repeatedly clicking the action.

Request replay returns HTTP 409

- `PUNCHOUT_REQUEST_REPLAY_IN_PROGRESS`: wait for the original call to finish before retrying the identical request.
- `PUNCHOUT_REQUEST_REPLAY_CONFLICT`: the same `X-Punchout-Request-Key`

was reused with changed input. Generate a new key for the new semantic action.

Request-body key aliases and `X-Idempotency-Key` are not accepted.

Inbound OrderRequest does not create an order

Open the inbound-order detail before retrying. Check its effective processing mode, status, failure code and normalized payload.

- **Acknowledge Only** never builds a quote.
- **Validate Only** can finish successfully without a Magento order ID.
- **Create Magento Order** requires a resolvable customer and products, valid addresses, and available shipping and payment methods.
- A buyer-profile override can replace the Store-view processing or pricing mode.
- Snapshot-based pricing needs the original BuyerCookie and an unexpired returned-cart snapshot.

If the status is **Manual Review**, inspect the reason and process or reject the record from **Punchout > Inbound Orders**. Do not repeatedly submit the same PO with changed content under the same identity.

Document remains pending or failed

Verify that Magento cron runs the `default` group. The queue processor is scheduled every five minutes:

```
php bin/magento cron:run --group=default
```

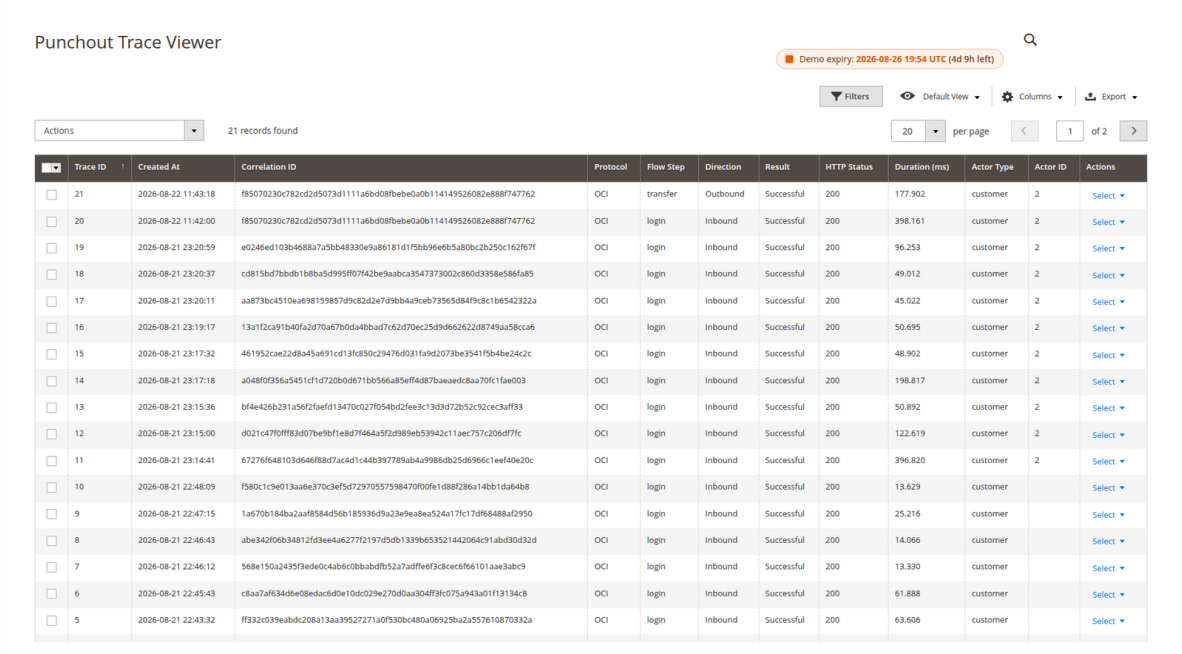
Open the queue row and check its next-attempt time, Store View, destination, status and last error. Delivery uses durable Buyer Profile/order routing and does not require an unexpired browser context. Confirm the dedicated ASN or Invoice URL and supplier credentials before using Reset or Force Run.

If the row is **Delivery Unknown**, Magento cannot know whether the receiver accepted the request before the connection failed. Check the procurement platform before using the separately authorized Retry Unknown Delivery action; an unverified retry can create a duplicate business document.

Trace appears empty

Trace logging is store-scoped. Confirm it is enabled in the buyer's resolved store, not only at Default Config. Old records are removed by the daily retention cleanup.

A working trace for one OCI session looks like this — login and cart-transfer steps share one correlation ID:



Trace ID	Created At	Correlation ID	Protocol	Flow Step	Direction	Result	HTTP Status	Duration (ms)	Actor Type	Actor ID	Actions
21	2026-08-22 11:43:18	f85070230c782cd2d5073d1111a6bd08fbebe0a0b114149526082e888f747762	OCI	transfer	Outbound	Successful	200	177.902	customer	2	Select
20	2026-08-22 11:42:00	f85070230c782cd2d5073d1111a6bd08fbebe0a0b114149526082e888f747762	OCI	login	Inbound	Successful	200	398.161	customer	2	Select
19	2026-08-21 23:20:59	e02246ed103b4688a7a5bb48330e9a86181d1f5bb96e6b5a80bc2b250c162f67f	OCI	login	Inbound	Successful	200	96.253	customer	2	Select
18	2026-08-21 23:20:37	cd815bd7bbdb1b8ba5d995ff07f42be9aabca3547373002c860d3358e586fa85	OCI	login	Inbound	Successful	200	49.012	customer	2	Select
17	2026-08-21 23:20:11	aa873bc4510ea698159857d9c82d2e7d9bb4a9ceb73565d84f9c8c1b6542322a	OCI	login	Inbound	Successful	200	45.022	customer	2	Select
16	2026-08-21 23:19:17	13a1f2ca91b40fa2d70a67b0da4bbad7c62d70ec25d9d662622d8749aa58cca6	OCI	login	Inbound	Successful	200	50.695	customer	2	Select
15	2026-08-21 23:17:32	461952cae22d8a45a691cd13fc850c29476d031fa9d2073be3541f5b4be24c2c	OCI	login	Inbound	Successful	200	48.902	customer	2	Select
14	2026-08-21 23:17:18	a048f0f356a5451cf1d720b0d671bb566a85eff4d87baeaedc8aa70fc1fae003	OCI	login	Inbound	Successful	200	198.817	customer	2	Select
13	2026-08-21 23:15:36	bf4e426b231a56f2faefd13470c027f054bd2fee3c13d3d72b52c92cec3aff33	OCI	login	Inbound	Successful	200	50.892	customer	2	Select
12	2026-08-21 23:15:00	d021c47f0fff83d07be9bf1e8d7f464a5f2d989eb53942c11aec757c206df7fc	OCI	login	Inbound	Successful	200	122.619	customer	2	Select
11	2026-08-21 23:14:41	67276f648103d646f88d7ac4d1c44b397789ab4a9986db25d6966c1eef40e20c	OCI	login	Inbound	Successful	200	396.820	customer	2	Select
10	2026-08-21 22:48:09	f580c1c9e013aa6e370c3ef5d72970557598470f00fe1d88f286a14bb1da64b8	OCI	login	Inbound	Successful	200	13.629	customer		Select
9	2026-08-21 22:47:15	1a670b184ba2aaf8584d56b185936d9a23e9ea8ea524a17fc17df68488af2950	OCI	login	Inbound	Successful	200	25.216	customer		Select
8	2026-08-21 22:46:43	abe342f06b34812fd3ee4a6277f2197d5db1339b653521442064c91abd30d32d	OCI	login	Inbound	Successful	200	14.066	customer		Select
7	2026-08-21 22:46:12	568e150a2435f3ede0c4ab6c0bbabdfb52a7adffe6f3c8cec6f66101aae3abc9	OCI	login	Inbound	Successful	200	13.330	customer		Select
6	2026-08-21 22:45:43	c8aa7af634d6e08edac6d0e10dc029e270d0aa304ff3fc075a943a01f13134c8	OCI	login	Inbound	Successful	200	61.888	customer		Select
5	2026-08-21 22:43:32	ff332c039eabdc208a13aa39527271a0f530bc480a06925ba2a557610870332a	OCI	login	Inbound	Successful	200	63.606	customer		Select

Use the Error Catalog for the exact response code and Trace and Simulator for a repeatable diagnosis flow.

Error Catalog

These are the public `PUNCHOUT_*` response codes returned by the PunchOut module. A security response may also include more specific codes in its `errors` array.

Availability and setup

Code	Meaning and next check
`PUNCHOUT_DISABLED`	PunchOut is disabled in the resolved store scope.
`PUNCHOUT_MISSING_REQUIRED_FIELDS`	OCI buyer/hook values or required cXML setup values were not parsed.
`PUNCHOUT_BUYER_MISSING`	The request did not contain the buyer identity required for context creation.
`PUNCHOUT_BUYER_NOT_ALLOWED`	No active buyer matched protocol, identity and store scope.
`PUNCHOUT_AUTHENTICATION_FAILED`	Protocol or optional transport credentials did not match; read the returned messages.
`PUNCHOUT_INVALID_CUSTOMER_REFERENCE`	A supplied/assigned customer reference is invalid for the resolved context.
`PUNCHOUT_INTERNAL_ERROR`	An unexpected server error occurred; use the correlation time and Magento logs without exposing the request secret.

Request security

`PUNCHOUT_SECURITY_VALIDATION_FAILED` is the top-level error. Its nested codes can include:

Nested code	Meaning and next check
`PUNCHOUT_IP_NOT_ALLOWED`	Magento's resolved client IP is outside the configured list.
`PUNCHOUT_SIGNATURE_SECRET_MISSING`	Strict mode is on but the scoped signature secret is empty.
`PUNCHOUT_SIGNATURE_MISSING`	No signature header/parameter was supplied.

Nested code	Meaning and next check
`PUNCHOUT_SIGNATURE_INVALID`	The supplied digest is not a 64-character hexadecimal SHA-256 value.
`PUNCHOUT_SIGNATURE_MISMATCH`	HMAC secret or signed payload representation differs.

Return URL

The setup controller returns `PUNCHOUT_INVALID_HOOK_URL` with one or more details:

Nested code	Meaning and next check
`PUNCHOUT_MISSING_HOOK_URL`	No OCI hook or cXML form-post URL was found.
`PUNCHOUT_INSECURE_HOOK_URL`	The URL is not HTTPS.
`PUNCHOUT_HOOK_URL_CREDENTIALS_NOT_ALLOWED`	The URL contains user or password credentials. Remove them and authenticate through the supported transport modes instead.
`PUNCHOUT_INVALID_HOOK_URL_PORT`	The URL port is outside the valid TCP port range.
`PUNCHOUT_HOOK_URL_BLOCKED_HOST`	`localhost` was used outside sandbox mode.
`PUNCHOUT_HOOK_URL_BLOCKED_IP`	A literal or resolved address is private/reserved outside sandbox mode.
`PUNCHOUT_HOOK_URL_DNS_RESOLVE_FAILED`	The hostname produced no usable A/AAAA address.

Context and cart

Code	Meaning and next check
`PUNCHOUT_MISSING_TOKEN`	No token was supplied in the supported parameter/header.
`PUNCHOUT_CONTEXT_NOT_FOUND`	Logout could not resolve the supplied token.
`PUNCHOUT_CONTEXT_EXPIRED`	The token is absent, expired or already logged out for the requested action.

Code	Meaning and next check
`PUNCHOUT_CONTEXT_NOT_ACTIVE`	HTTP 409. The resolved context is no longer authenticated, for example because its cart was already transferred or the session was logged out. Start a new PunchOut session instead of reusing the token.
`PUNCHOUT_CART_ADD_FAILED`	SKU, quantity, availability or Magento add-to-cart validation failed.
`PUNCHOUT_CART_UPDATE_FAILED`	No item was resolved or an update value was invalid.
`PUNCHOUT_CART_REMOVE_FAILED`	No matching item identifier was supplied/found.
`PUNCHOUT_CART_GET_FAILED`	Magento could not resolve or read the bound quote.
`PUNCHOUT_OPERATION_RESTRICTED`	The active storefront operation policy does not permit this action.

OCI product functions

Code	Meaning and next check
`PUNCHOUT_OCI_VERSION_INVALID`	The request contains an unsupported OCI version.
`PUNCHOUT_OCI_VERSION_MISMATCH`	The request version differs from the OCI version configured on the buyer profile.
`PUNCHOUT_OCI_PROFILE_VERSION_INVALID`	The buyer profile has no supported OCI version. Open the profile and save a valid value.
`PUNCHOUT_OCI_FUNCTION_NOT_SUPPORTED`	The requested OCI function is not implemented for this flow.

Code	Meaning and next check
`PUNCHOUT_OCI_DETAIL_PRODUCT_REQUIRED`	`DETAIL` or `DETAILADD` was sent without `PRODUCTID`.
`PUNCHOUT_OCI_DETAIL_PRODUCT_NOT_FOUND`	The configured SKU or external product identifier did not resolve to an eligible product.
`PUNCHOUT_OCI_DETAILADD_VERSION_NOT_SUPPORTED`	`DETAILADD` was requested for a buyer profile that does not use OCI 5.
`PUNCHOUT_OCI_QUANTITYCHECK_VERSION_NOT_SUPPORTED`	`QUANTITYCHECK` was requested for a buyer profile that does not use OCI 5.0.
`PUNCHOUT_OCI_QUANTITYCHECK_FAILED`	The availability check failed: `EXT_PRODUCT_ID` is missing or unresolvable, `QUANTITY` is invalid, or the configured lead-time attribute is unusable. Read the returned message.

Transfer and mapping

Code	Meaning and next check
`PUNCHOUT_VALIDATION_ERROR`	Read the returned `errors` list for item, URL, mapping or currency details.
`PUNCHOUT_EMPTY_ITEMS`	The transfer has no cart lines.
`PUNCHOUT_ITEM_<n>_MISSING_DESCRIPTION`	The numbered line has no description.

Code	Meaning and next check
`PUNCHOUT_ITEM_<n>_INVALID_QTY`	The numbered line has a non-positive quantity.
`PUNCHOUT_ITEM_<n>_MISSING_UNIT`	The numbered line has no unit.
`PUNCHOUT_ITEM_<n>_MISSING_PRICE`	The numbered line has a missing, non-numeric or negative price.
`PUNCHOUT_ITEM_<n>_INVALID_FORMAT`	The numbered cart line is not a readable item structure.
`PUNCHOUT_ITEM_<n>_INVALID_OCI_UNIT`	The numbered OCI line has a unit longer than three characters.
`PUNCHOUT_ITEM_<n>_INVALID_OCI_QTY`	The numbered OCI line has a quantity that cannot be formatted for OCI output.
`PUNCHOUT_ITEM_<n>_INVALID_OCI_PRICE`	The numbered OCI line has a price that cannot be formatted for OCI output.
`PUNCHOUT_MIXED_ITEM_CURRENCIES`	cXML lines do not share one currency.
`PUNCHOUT_MAPPING_INVALID_PROFILE`	The context mapping is absent or not a usable runtime profile.
`PUNCHOUT_MAPPING_INVALID_RULE_CONFIGURATION`	A saved rule cannot be executed safely.
`PUNCHOUT_MAPPING_REQUIRED_<TARGET>`	A mandatory mapped target is empty after formatting/fallback.
`PUNCHOUT_MAPPING_INVALID_NEW_ITEM_UNIT`	An OCI unit contains more than three characters.
`PUNCHOUT_MAPPING_INVALID_NEW_ITEM_LEADTIME`	OCI lead time is not an integer with at most five digits.

Code	Meaning and next check
`PUNCHOUT_MAPPING_INVALID_NEW_ITEM_MANUFACTCODE`	The OCI manufacturer code exceeds ten characters.
`PUNCHOUT_UNSUPPORTED_PROTOCOL`	The transfer resolved neither OCI nor cXML.

Request replay

Code	Meaning and next check
`PUNCHOUT_REQUEST_REPLAY_IN_PROGRESS`	Another call with this operation and `X-Punchout-Request-Key` is still processing.
`PUNCHOUT_REQUEST_REPLAY_CONFLICT`	The operation and key were reused with a different request hash in the same Store view.
`PUNCHOUT_REQUEST_KEY_REQUIRED`	A manual retry of an unknown delivery was requested without `X-Punchout-Request-Key`.

A successful exact retry returns the previously stored response and is not an error.

Inbound OrderRequest processing

Code	Meaning and next check
`PUNCHOUT_INVALID_ORDER_REQUEST`	The cXML document is malformed or does not contain a supported OrderRequest.
`PUNCHOUT_ORDER_REQUEST_DISABLED`	OrderRequest intake is disabled on the resolved buyer profile. Enable **Accept OrderRequest** if this buyer should submit orders.
`PUNCHOUT_PO_NUMBER_MISSING`	No PO/order number was parsed.
`PUNCHOUT_PO_EMPTY_ITEMS`	The PO contains no parsed line items.
`PUNCHOUT_PO_CANCEL_FAILED`	A legacy context-cart cancellation could not clear the linked PunchOut quote.

Code	Meaning and next check
`PUNCHOUT_INVALID_PRICE`	A submitted or authoritative unit price is invalid for processing.
`PUNCHOUT_ITEM_MISSING`	Magento cannot resolve a submitted product using the configured product reference.
`PUNCHOUT_ORDER_REJECTED`	An inbound policy or Magento quote validation rejected the order.
`PUNCHOUT_MANUAL_REVIEW`	The request requires an administrator to review and process or reject it.
`PUNCHOUT_PRICE_REVIEW`	A price difference reached the configured manual-review action.
`PUNCHOUT_QUANTITY_REVIEW`	A changed returned-cart quantity requires review.
`PUNCHOUT_NEW_ITEM_REVIEW`	A product not present in the returned cart requires review.
`PUNCHOUT_REMOVED_ITEM_REVIEW`	The approved order removed a returned-cart line that requires review.
`PUNCHOUT_SNAPSHOT_REVIEW`	The returned-cart snapshot is missing or expired and the configured action is manual review.
`PUNCHOUT_UPDATE_REVIEW`	An OrderRequest update requires manual review.
`PUNCHOUT_UPDATE_REPLACEMENT_CANCELLED`	A cancel-and-recreate update produced a replacement order that is cancelled; the inbound order is held for manual review.
`PUNCHOUT_UPDATE_CANCELLATION_REVIEW`	The original Magento order could not be cancelled during a cancel-and-recreate update; the inbound order is held for manual review with the failure detail.
`PUNCHOUT_CANCELLATION_REVIEW`	An OrderRequest delete requires manual review.
`PUNCHOUT_MANUALLY_REJECTED`	An administrator rejected the inbound order.

Code	Meaning and next check
`PUNCHOUT_PROCESSING_ERROR`	Quote or order processing raised an unexpected error. Inspect the detail record and Magento log for the same time.
`PUNCHOUT_ORDER_PROCESSING`	The same OrderRequest is still being processed. Retry the unchanged request later.
`PUNCHOUT_RETRY_EXHAUSTED`	The configured inbound-order processing attempts have been used. Review the stored failure before retrying manually.

Confirmation documents

These codes appear on document-queue rows for `ConfirmationRequest` delivery.

Code	Meaning and next check
`PUNCHOUT_CONFIRMATION_ENDPOINT_MISSING`	The stored confirmation routing has no endpoint URL. Check the buyer's **Confirmation URL** and the Store-view default.
`PUNCHOUT_CONFIRMATION_DATA_INVALID`	The queued confirmation record lacks a valid order reference or confirmation type.
`PUNCHOUT_CONFIRMATION_PREREQUISITE_MISSING`	The inbound order lacks the store, Magento order, buyer or endpoint data required to queue a confirmation.
`PUNCHOUT_CONFIRMATION_PAYLOAD_FAILED`	The confirmation routing could not be decrypted or the cXML payload could not be generated; the queue row was closed with this diagnostic.

ASN and invoice documents

Code	Meaning and next check
`PUNCHOUT_ASN_ENDPOINT_MISSING`	No effective ASN URL is configured for shipment delivery.
`PUNCHOUT_INVOICE_ENDPOINT_MISSING`	No effective Invoice URL is configured for invoice delivery.
`PUNCHOUT_ASN_PAYLOAD_MISSING`	Magento could not build or load the retained shipment payload.
`PUNCHOUT_INVOICE_PAYLOAD_MISSING`	Magento could not build or load the retained invoice payload.
`PUNCHOUT_ASN_SHIPMENT_ID_MISSING`	No shipment/ASN/notice number was supplied.
`PUNCHOUT_ASN_EMPTY_ITEMS`	No shippable line items were supplied or parsed.
`PUNCHOUT_ASN_DATA_INCOMPLETE`	Magento could not hydrate the required shipment, order or supplier cXML data.
`PUNCHOUT_ASN_DELIVERY_FAILED`	The HTTPS POST raised an error or returned non-2xx.
`PUNCHOUT_INVOICE_NUMBER_MISSING`	No invoice number was supplied or parsed.
`PUNCHOUT_INVOICE_TOTAL_INVALID`	Invoice total is missing or not greater than zero.
`PUNCHOUT_INVOICE_CURRENCY_MISSING`	No currency was supplied or parsed.
`PUNCHOUT_INVOICE_DATA_INCOMPLETE`	Magento could not hydrate the required invoice, order or supplier cXML data.
`PUNCHOUT_INVOICE_DELIVERY_FAILED`	The HTTPS POST raised an error or returned non-2xx.
`PUNCHOUT_DOCUMENT_TYPE_UNSUPPORTED`	A queue record refers to an unsupported outbound document type.
`PUNCHOUT_DOCUMENT_CONFIGURATION_ERROR`	Required routing or supplier configuration is incomplete.

Code	Meaning and next check
`PUNCHOUT_DOCUMENT_PAYLOAD_MISSING`	Document generation completed without an outbound payload.
`PUNCHOUT_DOCUMENT_PAYLOAD_DECRYPTION_FAILED`	Magento cannot decrypt the retained outbound payload. Check the installation's encryption key and deployment history.
`PUNCHOUT_INBOUND_ORDER_REFERENCE_MISSING`	A queue row has no durable inbound-order reference.
`PUNCHOUT_DOCUMENT_IDENTITY_INCOMPLETE`	A shipment or invoice queued for reconciliation is missing its Store View, inbound-order or buyer reference.
`PUNCHOUT_BUYER_PROFILE_MISSING`	The inbound order linked to a shipment or invoice no longer references a buyer profile.
`PUNCHOUT_DOCUMENT_STORE_MISSING`	The Store View referenced by a queued shipment or invoice document no longer exists.
`PUNCHOUT_DOCUMENT_STORE_MISMATCH`	The queue row and its inbound order belong to different Store Views.
`PUNCHOUT_QUEUE_OWNERSHIP_LOST`	The worker lost its queue lease before it could save the outbound payload. The row was not completed by that worker.
`PUNCHOUT_DELIVERY_OUTCOME_UNKNOWN`	Magento cannot tell whether the remote system accepted the document. Check the receiver before using the dedicated manual retry action.
`PUNCHOUT_QUEUE_PROCESSING_ERROR`	A queue handler raised an unexpected processing exception.

When the API message and trace do not explain a failure, reproduce it in the simulator with privacy-safe values and then inspect Magento logs for the same time window.

FAQ

Does PunchOut support both OCI and cXML?

Yes. The core package has OCI login/cart-return routes and cXML setup/`PunchOutOrderMessage` routes. A buyer profile uses one protocol at a time.

Does the extension require Adobe Commerce B2B?

No. The Composer package depends on the Magento framework, not Adobe Commerce B2B. The customer, catalog and pricing capabilities available in a particular installation still depend on that Magento project.

Are SAP Ariba, Coupa and Oracle Procurement integrations included?

The module includes editable outbound starter mappings oriented toward those platforms. That is not certification or a promise that every tenant accepts the defaults unchanged. The buyer must supply its current URLs, credentials, protocol dialect, required fields and test cases.

Why can I not assign an inbound template to a buyer?

Buyer profiles intentionally list active outbound runtime profiles. The shipped inbound fixtures are synchronized reference/catalog definitions; the current OCI and cXML setup controllers parse supported fields directly rather than executing inbound mapping rules.

Does cart return place an order in Magento?

No. Return to System sends the cart to the procurement platform. The buyer normally completes approvals and creates its PO outside Magento.

Does the inbound PO endpoint create a Magento order?

It can. **Acknowledge Only** records the request, **Validate Only** builds and validates the Magento quote, and **Create Magento Order** places the order. The effective value comes from the buyer profile first and the selected Store view second.

Which price is returned?

The cart uses Magento pricing by default, including the customer context resolved for the buyer. Prices sent to the cart API are ignored unless **Allow inbound cart item prices** is explicitly enabled.

How does PunchOut resolve a Magento customer?

The Buyer Profile explicitly selects guest, fixed Magento customer, customer e-mail, optional request customer ID or required request customer ID. Fixed and resolved customers must belong to the selected Website unless Magento global customer sharing is enabled. Customer-by-e-mail can create a missing account when JIT provisioning is enabled. Magento's normal account-confirmation policy still applies.

Should strict signatures always be enabled in production?

Only when the calling integration sends the SoftwareSilo HMAC-SHA256 signature calculated with the same secret and payload rules. OCI or cXML support alone does not guarantee that signature. HTTPS, buyer credentials and an IP allowlist remain important regardless.

Why does an exact retry return the old response?

That is the request-replay contract. The same endpoint, `X-Punchout-Request-Key` and request hash replay the saved result with its original HTTP status. Use a new key when the cart or document action changes.

How are shipments and invoices sent?

Magento shipment and invoice events create queue rows for PunchOut-created orders. The default cron group processes them every five minutes and posts cXML to the dedicated ASN or Invoice URL from the Buyer Profile or Store View. Routing and the encrypted payload are durable, so the browser session may already be expired.

Can another module customize actor or permission behavior?

Yes. A separate Magento module can implement or decorate the public actor and permission service contracts and observe the PunchOut lifecycle events. The core package has no dependency on Sublogin or B2B Quote and does not advertise built-in bridges for either module.