cXML Flow

Punchout Documentation

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

LAST UPDATED
2026-09-26

SOURCE
Open this article online

EXTENSION
Magento 2 Extension Punchout

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.

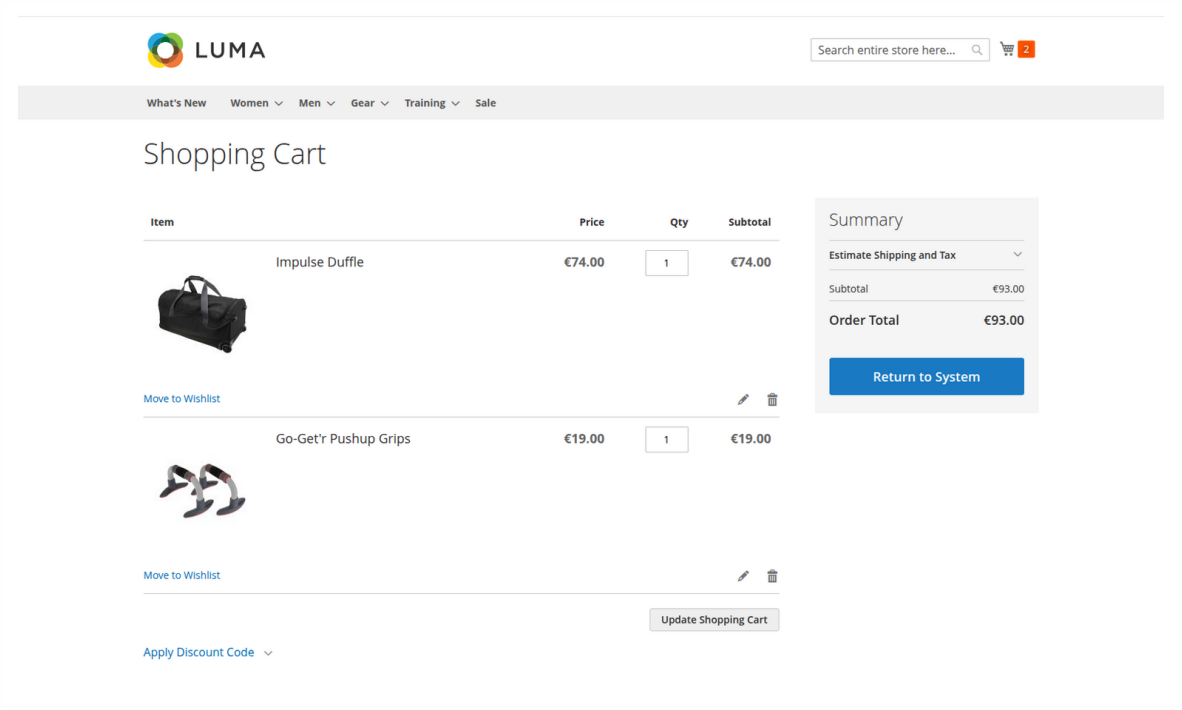
2. Resolve the buyer, operation and level-2 target

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.

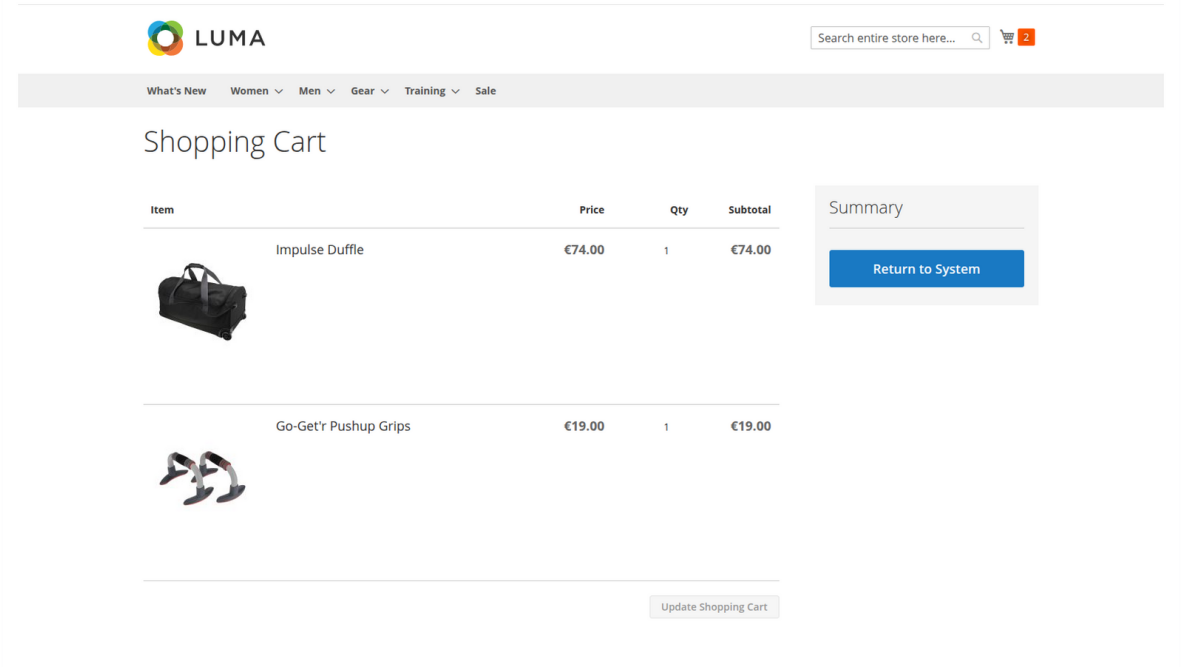
Open `/punchout/session/activate?token=<token>&target_url=<same-host-path>` to bind the browser and mapped customer.

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.

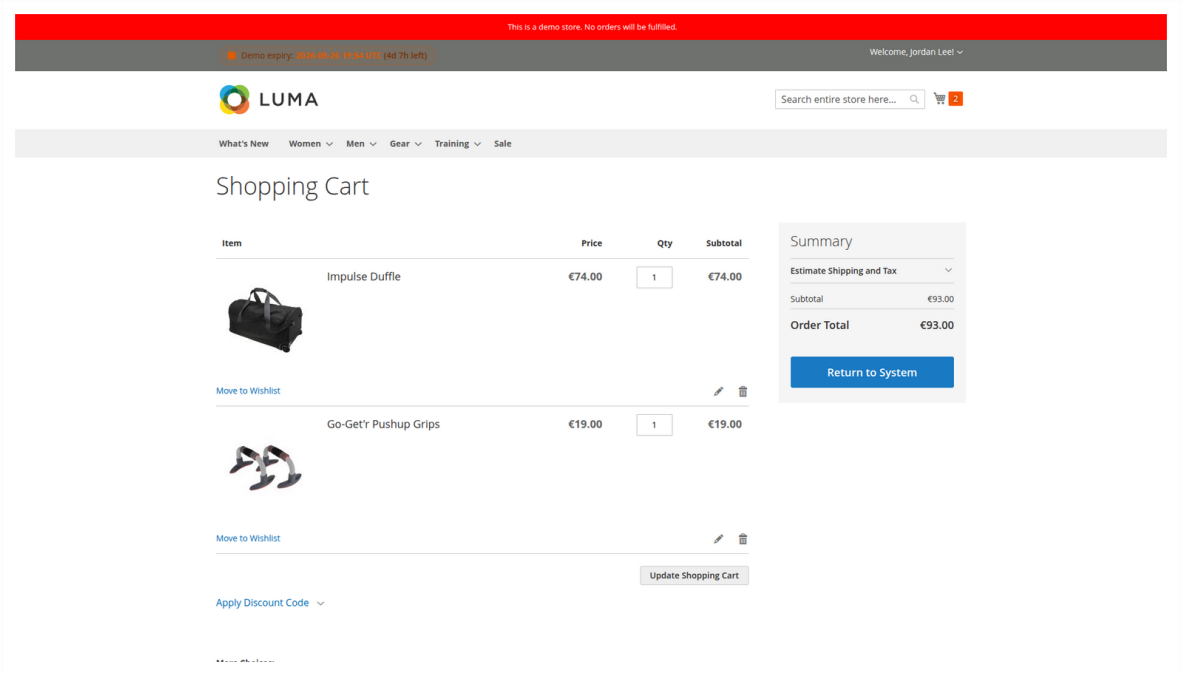


The corresponding inspect session keeps the reconstructed cart visible while removing mutation and checkout controls. Quantity values are read-only, and the only active cart action is the return to the procurement system.



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.