



Endpoint Reference

Punchout Documentation

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

LAST UPDATED
2026-10-13

SOURCE
[Open this article online](#)

EXTENSION
[Magento 2 Extension Punchout](#)

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.