Trace and Simulator

Punchout Documentation

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

LAST UPDATED
2026-09-26

SOURCE
Open this article online

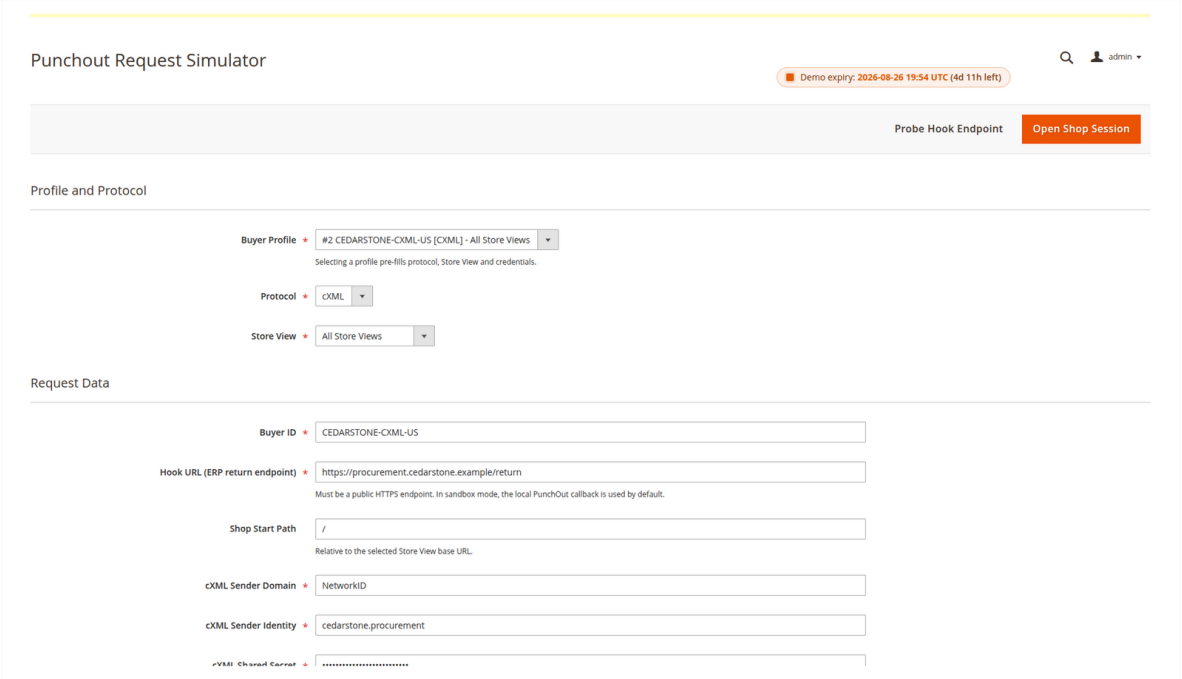
EXTENSION
Magento 2 Extension Punchout

Trace and Simulator

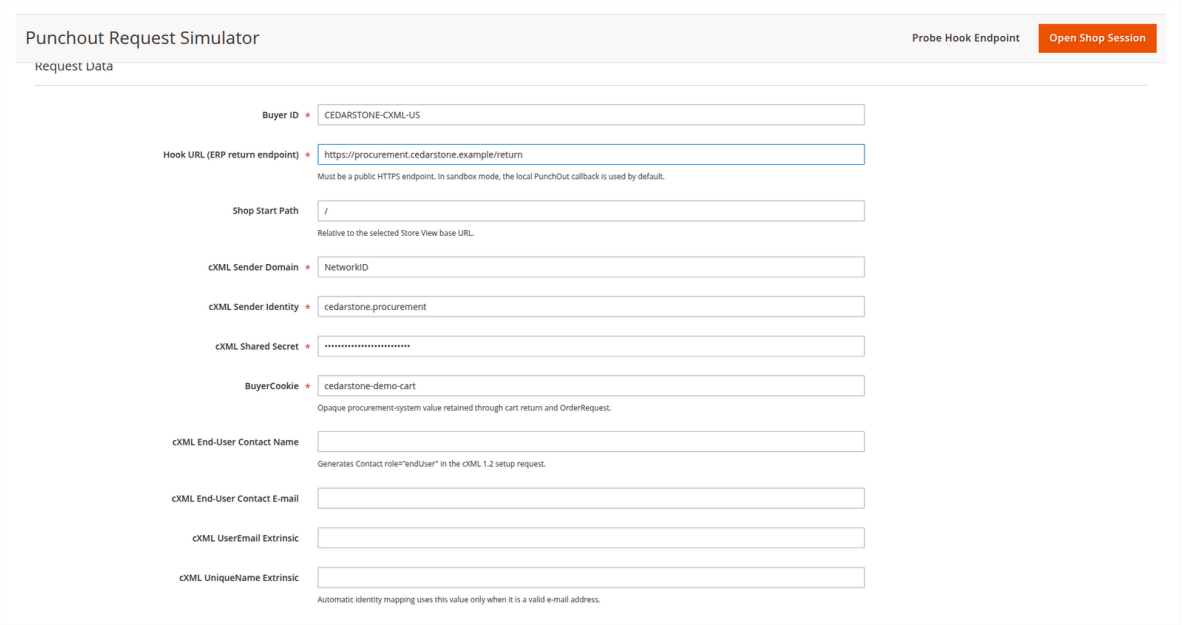
Use the simulator to build a controlled setup request. Use the trace viewer to follow a real request by correlation ID. Together they show whether a failure happened during authentication, setup, cart return, mapping or document delivery.

Build a simulator request

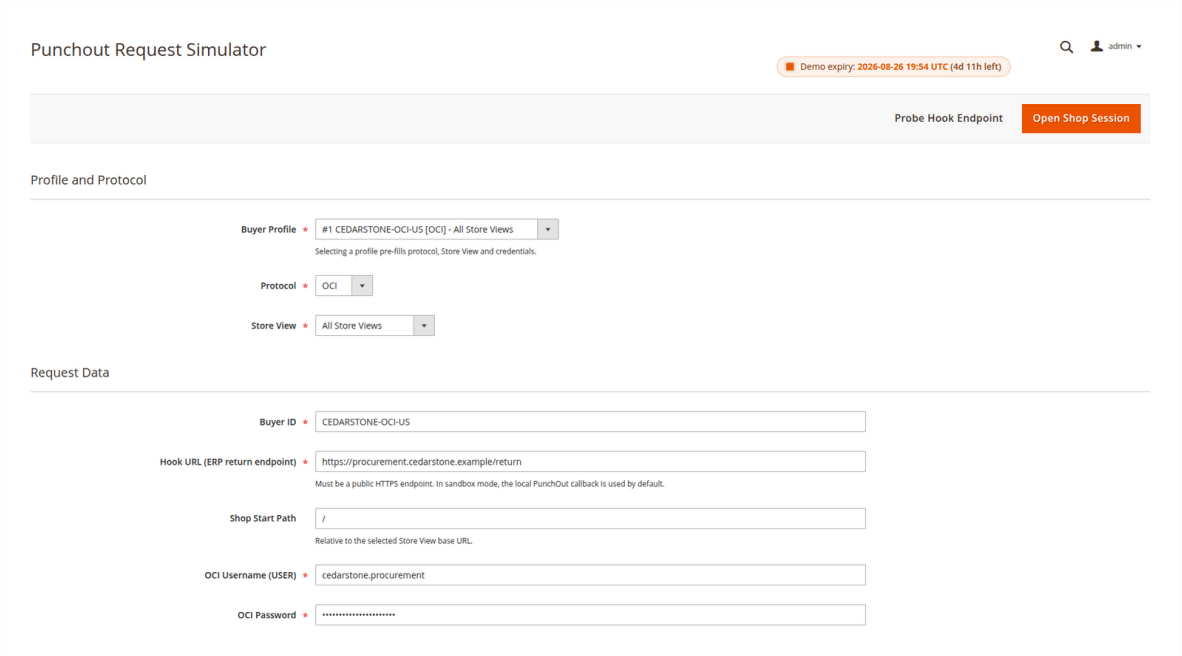
Open **Punchout > Punchout Simulator** and choose OCI or cXML. Use a configured buyer profile and a privacy-safe test customer in the local or buyer-provided sandbox.



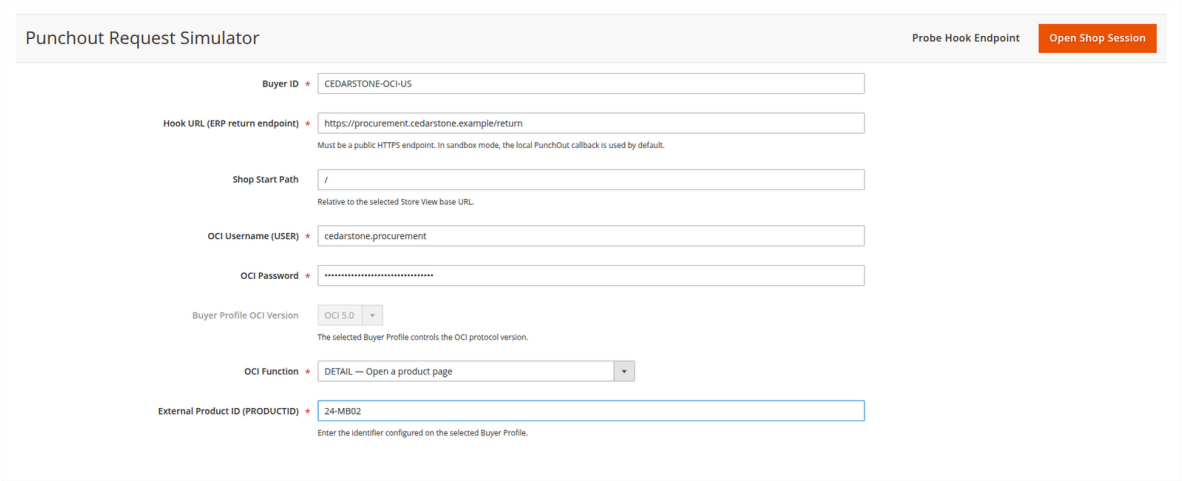
The request section keeps the stored shared secret masked while exposing the sender identity and the fictional BuyerCookie used to correlate the later cart return:



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:

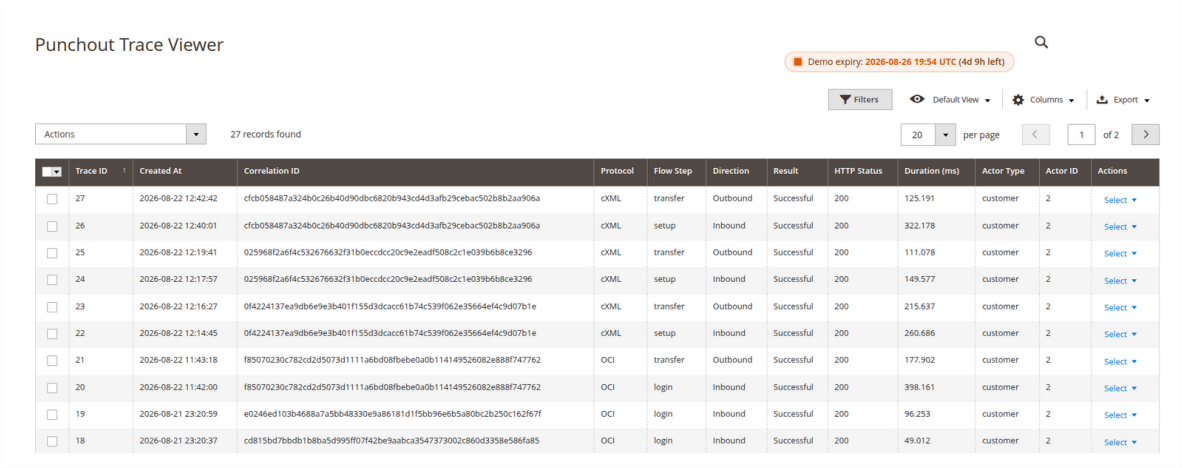
- build the protocol request from entered fields;
- show the target endpoint and technical payload preview;
- generate a cURL example;
- probe the return URL with the same URL-safety policy used by PunchOut;
- open a storefront session after a successful setup.

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.

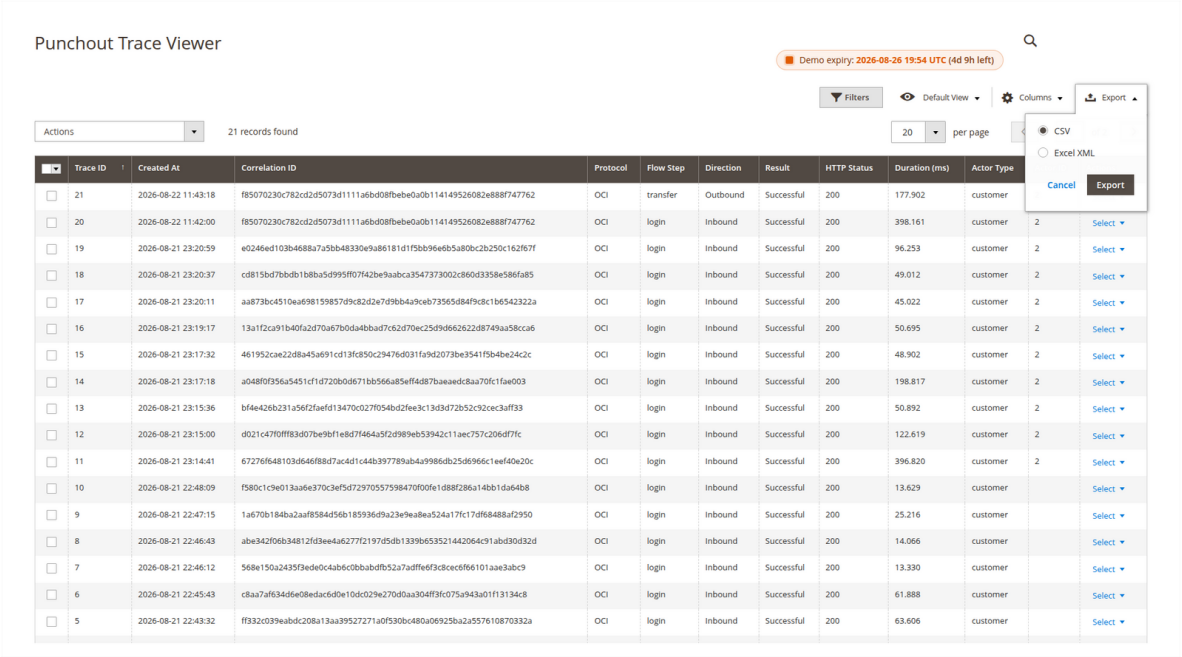
Do not paste a production shared secret into screenshots, tickets or documentation. A masked admin field proves that a value exists; it does not prove the other system uses the same value.

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.

