Contents

Agent Checkout Handoff **1**
1. Overview . . . 1
Key Features . . . 2
Compatibility . . . 2
2. Installation . . . 2
Via Composer . . . 2
Verify . . . 2
3. Configuration . . . 3
4. How It Works . . . 4
4.1 Issuing a link . . . 4
4.2 Opening a link . . . 4
4.3 Refusals . . . 6
4.4 Security . . . 7
4.5 Data . . . 7
5. Use Cases . . . 7
5.1 A chat assistant that sells running shoes . . . 7
5.2 A returning customer with a cart already open . . . 8
5.3 A store that requires an account to order . . . 8
6. Troubleshooting . . . 8
7. Support & Changelog . . . 8

Agent Checkout Handoff

User Guide
Version: 1.0.0
Compatibility: Adobe Commerce and Magento Open Source 2.4.7 – 2.4.9, PHP 8.2 – 8.5
Vendor: Webmaster Ramos

---

1. Overview

Agent Checkout Handoff hands a cart that an AI shopping agent built over GraphQL to the shopper's own browser, and opens the checkout of Adobe Commerce and Magento Open Source with it. It is an extension for Magento that closes a gap the platform leaves open:

- **The checkout shows only the browser's own cart.** A cart built over GraphQL belongs to no browser session, so a shopper sent to the checkout finds it empty.
- **The cart ID cannot travel in a link.** The masked cart ID is a bearer credential: whoever holds it can keep changing the cart – items, addresses – while the shopper is checking out.

The agent's backend asks one GraphQL mutation for a single-use link that expires in minutes. The shopper opens it and lands in checkout with the agent's cart; at that moment a guest cart gets a new cart ID and the agent's one stops working.

Key Features

- GraphQL mutation `createAgentCheckoutHandoff`: cart ID in, checkout link and expiry time out, with Magento's own cart access check
- Single-use links that expire after 1 to 60 minutes; a new link for a cart revokes the earlier unused one
- The agent's guest cart ID retired when the link is opened
- Every browser state handled: guest or signed in, an existing cart replaced or merged, guest checkout off, a customer's own cart
- One generic message for every refusal; only a hash of each link is stored
- The minicart shows the cart on the first page, on Luma and Hyvä Theme
- Enabled, link lifetime and cart policy per store view

Compatibility

Platform	Versions
Adobe Commerce	2.4.7 – 2.4.9
Magento Open Source	2.4.7 – 2.4.9
PHP	8.2 – 8.5
Themes	Luma, Hyvä Theme (with native Magento checkout)

The storefront must be served over HTTPS. Magento cron must run for the hourly purge of expired links; the links themselves work without it.

---

2. Installation

Via Composer

```
composer require webmaster-ramos/module-agent-checkout-handoff
bin/magento module:enable WebRamos_AgentCheckoutHandoff
bin/magento setup:upgrade
bin/magento setup:di:compile
bin/magento cache:flush
```

`setup:upgrade` creates the table `webramos_agent_checkout_handoff`. `setup:di:compile` is required in production mode and harmless elsewhere.

Verify

```
bin/magento module:status WebRamos_AgentCheckoutHandoff
```

The module should be enabled, and **Stores → Configuration → Webmaster Ramos → Agent Checkout Handoff** should open. The extension is off until you enable it.

---

3. Configuration

Stores → Configuration → Webmaster Ramos → Agent Checkout Handoff. Every field can be set per store view, so the extension can be on for one storefront and off for another.

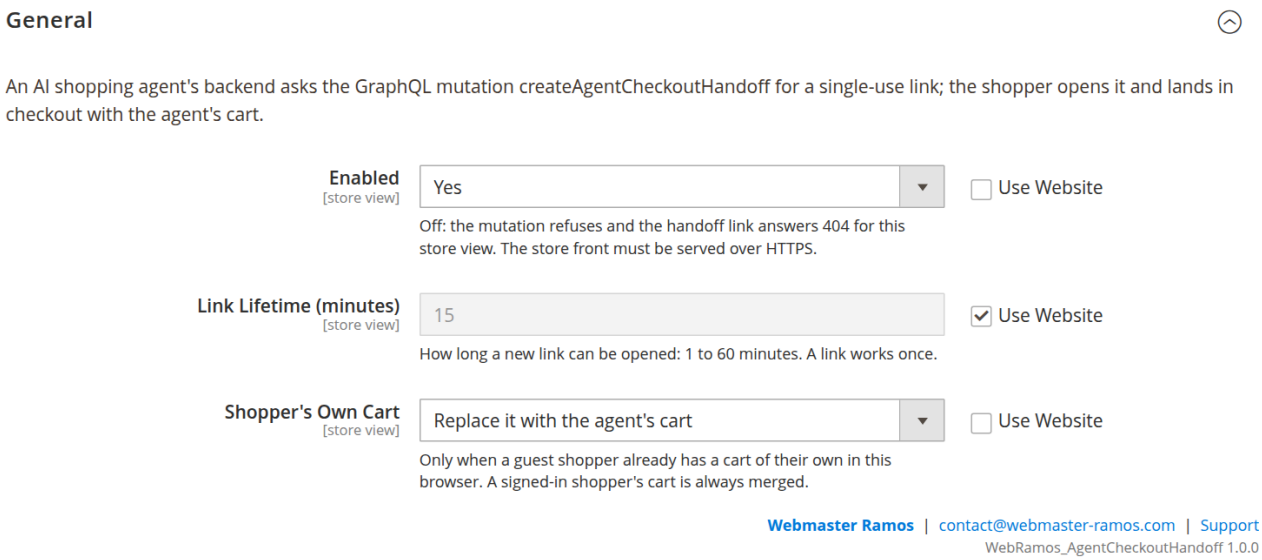


Figure 1: The General group of the configuration section

Field	Default	What it does
Enabled	No	Off: the mutation refuses and every link answers 404 for this store view.
Link Lifetime (minutes)	15	How long a new link can be opened: a whole number from 1 to 60. A link works once.
Shopper's Own Cart	Replace it with the agent's cart	Applies only when a guest shopper's browser already holds a non-empty cart of its own. "Replace it with the agent's cart" sets that cart aside; "Merge the agent's cart into it" adds the agent's items to it. A signed-in shopper's cart is always merged.

The line at the foot of the section names the module and its version – quote it when you contact support.

4. How It Works

4.1 Issuing a link

The agent builds the cart over Magento's GraphQL API as usual, then calls the mutation **from its server**:

```
mutation {
  createAgentCheckoutHandoff(input: { cart_id: "<masked cart id>" }) {
    url
    expires_at
  }
}
```

The answer carries the link – `https://<store>/agentcheckout/handoff/open/t/<token>/` – and its expiry time in ISO 8601, UTC. Send the `Store` header for any store view other than the default one; the link is built for that store view's own host. For a customer cart, send `Authorization: Bearer <customer token>`, as for any operation on that cart.

The mutation refuses when the extension is off for the store view, the cart is empty, or the storefront is not served over HTTPS; Magento's own cart errors apply to an unknown cart, a cart already ordered, and a cart the request may not use – a customer cart without its customer's token or with another customer's, or a guest cart called with a customer token. A link that cannot be stored (a database fault) is not a refusal: the agent gets Magento's "Internal server error" and the cause is logged in `var/log/exception.log`.

4.2 Opening a link

The agent sends the link to the shopper. When the shopper opens it, the extension attaches the cart to the browser and goes to checkout:

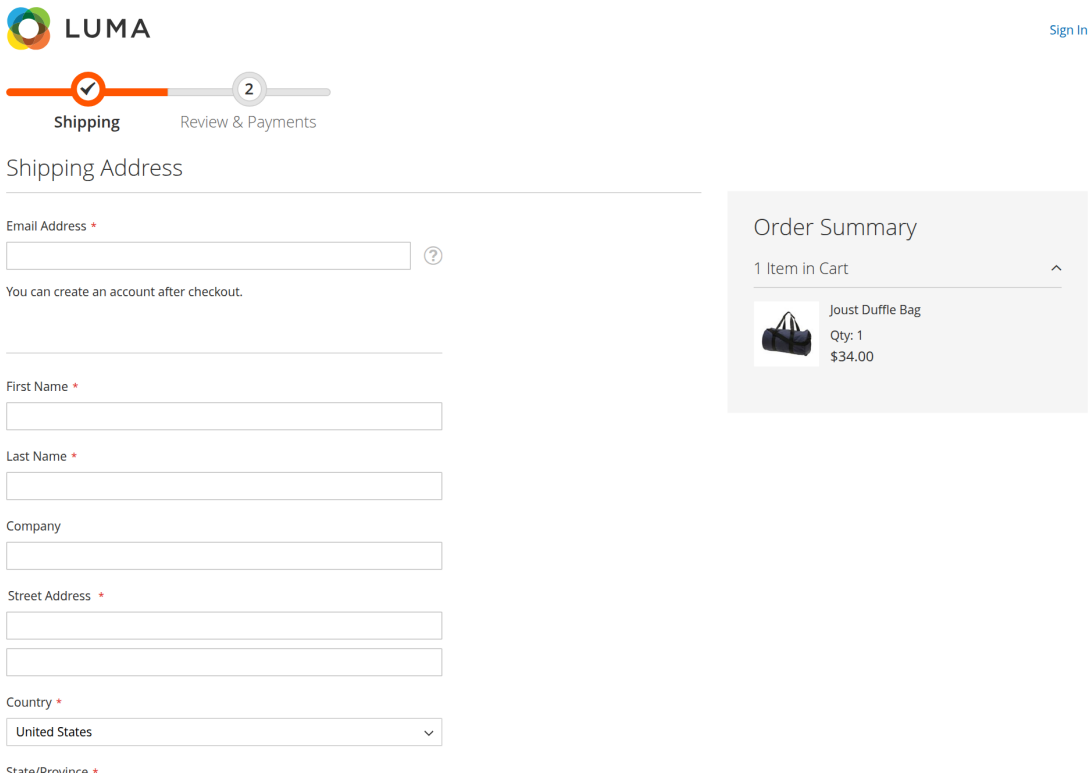


Figure 2: The Luma checkout right after the link

The link holds	The browser	Result
a guest cart	a guest with no cart of their own	Checkout with the agent's cart
a guest cart	a guest with a cart of their own	Checkout; that cart is replaced or merged, per **Shopper's Own Cart**
a guest cart	a signed-in shopper	Checkout; the items are merged into the account cart
a guest cart	guest checkout is switched off	Sign-in page; after sign-in, checkout with the cart
a guest cart	remembered by Persistent Shopping Cart, not signed in	Sign-in page; the link is kept and continues after sign-in
a customer's cart	that customer, signed in	Checkout with the cart
a customer's cart	not signed in	Sign-in page; after sign-in, checkout
a customer's cart	another customer, signed in	Refused; the link is kept

The same happens on Hyvä Theme with the native Magento checkout. The minicart shows the cart on the first page without a reload – here the Hyvä Theme cart page right after the link, its header counter already at one:

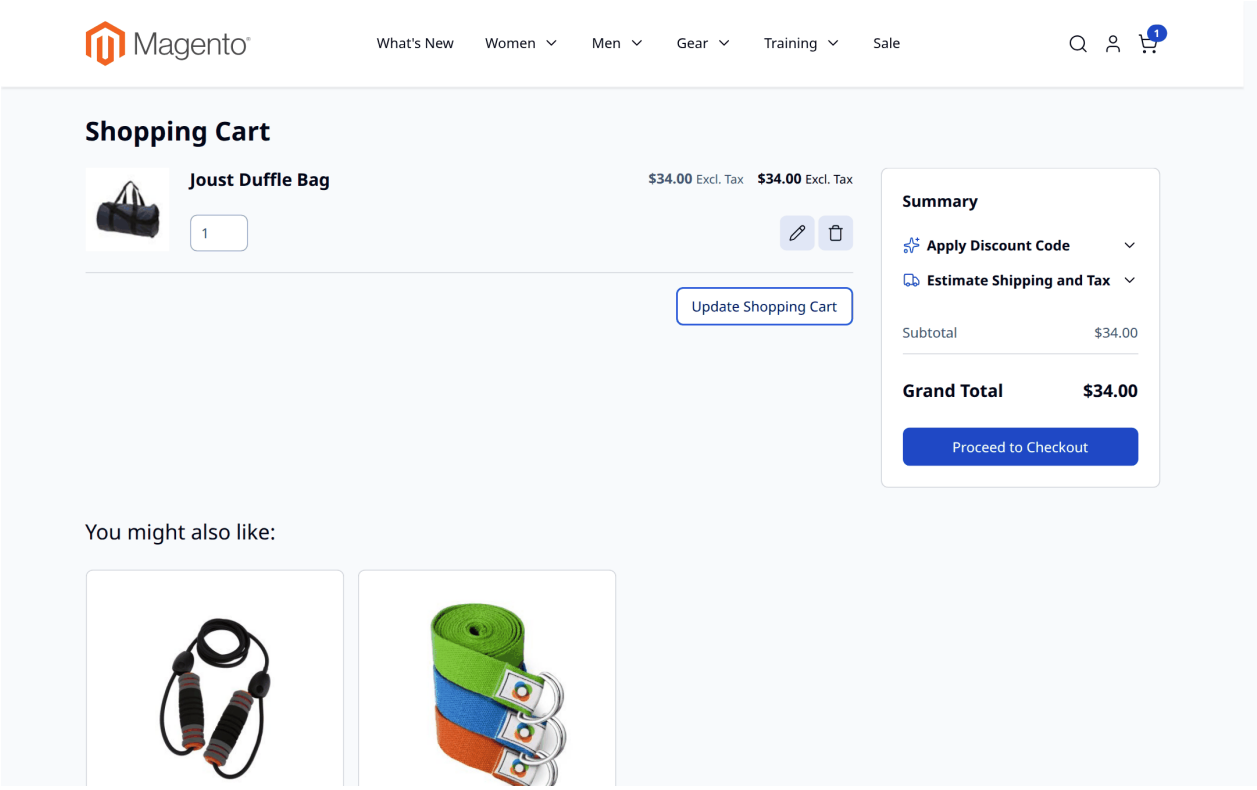


Figure 3: The Hyvä Theme cart right after the link

When a guest cart is opened, it gets a new cart ID: the ID the agent held stops working, so the agent – or anyone the ID was leaked to – cannot change the cart any more. A customer cart keeps its ID; access to it is the customer's token.

4.3 Refusals

An expired, used, unknown or malformed link, a link for another store view, and a cart that is empty or already ordered all end the same way: the cart page with "This checkout link is invalid or has expired." Nothing is attached, and the message never tells which check failed. The agent can issue a new link.

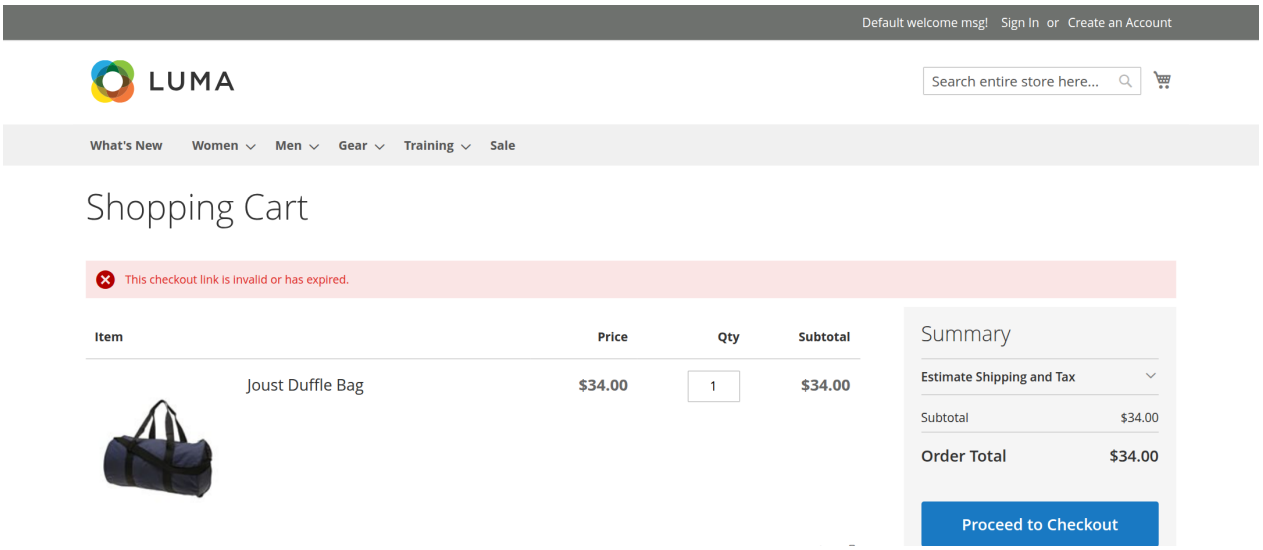


Figure 4: A link opened a second time

4.4 Security

- Links are 256-bit random tokens; only their SHA-256 hash is stored.
- A link works once and for at most 60 minutes, only over HTTPS and only on the store view it was issued for.
- The redirect goes to fixed storefront pages; nothing in the link can steer it elsewhere.
- Every response of the link carries `Cache-Control: no-store`, `Referrer-Policy: no-referrer` and `X-Robots-Tag: noindex, nofollow`.
- Treat a link like a password reset link and send it to the shopper only: whoever opens it first gets the cart's contents. No payment details are in a cart.

4.5 Data

The table `webramos_agent_checkout_handoff` holds the link hash, the cart, store view and customer IDs, and when the link was created, expires and was used. There are no names, addresses or payment data. The cron job `webramos_agent_checkout_handoff_purge` deletes rows 24 hours after they expire, and a row goes with its cart, customer or store view.

---

5. Use Cases

5.1 A chat assistant that sells running shoes

A shopper asks the store's chat assistant for running shoes in size 42. The assistant searches the catalog, adds the pair to a guest cart over GraphQL and asks for a link. The shopper taps it in the chat, lands in checkout with the shoes and pays. The assistant's cart ID is dead from that moment.

5.2 A returning customer with a cart already open

A signed-in customer has two items in their cart from yesterday. The agent builds a guest cart with a third item and sends the link; the customer opens it in the same browser and checks out with all three, as with any sign-in merge.

5.3 A store that requires an account to order

Guest checkout is switched off. The shopper opens the link, is asked to sign in with a notice that the cart is ready, and after signing in lands in checkout with the agent's items.

---

6. Troubleshooting

Symptom	Cause and fix
The mutation says the extension is not enabled	**Enabled** is No for the store view of the request. Check the `Store` header and the store view scope.
The mutation says the storefront needs HTTPS	The store view's secure base URL is not `https://`.
The link answers 404	The extension is off for the store view the link was opened on.
Every link ends on the cart page with the generic message	The link was used, expired, or opened on another store view than the one it was issued for. Issue a new link for the right store view.
The agent's cart ID fails after the handoff	Intended: a guest cart gets a new ID when it is opened. Build a new cart for the next order.
Old links pile up in the table	Magento cron is not running; the purge runs hourly.

---

7. Support & Changelog

- **Author:** Webmaster Ramos
- **Email:** contact@webmaster-ramos.com
- **Website:** https://webmaster-ramos.com
- **Support:** https://webmaster-ramos.com/support
- **Changelog:** see `CHANGELOG.md` inside the package