magmodules



Ads Conversions API Guide

OpenAI Commerce for Magento 2

The OpenAI Commerce extension can send purchase conversions from Magento straight to the OpenAI Ads Conversions API. Conversions are reported server-side, so they are not affected by ad blockers, browser privacy features or customers who close the browser before the success page loads. This gives your OpenAI Ads campaigns complete, accurate attribution in Ads Manager.

How It Works

1. **Click capture** — a visitor arriving from an OpenAI ad lands on a URL containing an `oppref` parameter. A small storefront script stores this identifier in a first-party cookie (`mm_openai_oppref`). It runs in the browser, so it works behind full page cache.
2. **Event capture** — when an order is placed, and for every other enabled event, the extension records it in a queue together with the `oppref` cookie, the Measurement Pixel's `__oppref` / `__obref` cookies (when the pixel is installed), the customer's IP address and user agent.
3. **Delivery** — a cron job runs every 5 minutes and posts the queue to `https://bzr.openai.com/v1/events` . Events are sent in batches; a rejected batch is retried event by event so a single problematic event never blocks the others. Temporary failures (network errors, HTTP 5xx, 429) are retried up to 5 times.
4. **Monitoring** — every event and its result is visible under **Marketing → OpenAI Commerce → Ads Conversions**, including the exact payload that was sent and the API response.

Purchase payload

For every order one `order_created` event is sent:

Field	Value
`id`	Order increment ID (e.g. `000000123`). OpenAI deduplicates on this value, so resending is safe
`timestamp_ms`	Order creation time
`action_source`	`web` for storefront, REST and GraphQL orders, `other` for admin-created orders
`source_url`	The store's checkout success URL
`oppref`	Click identifier from the landing page, when captured
`user`	Matching data — see below
`data.type`	`contents`
`data.amount` / `data.currency`	Conversion value in the order currency, as integer minor units (e.g. `4200` for €42.00)

Field	Value
`data.contents[]`	One entry per ordered line: SKU, name, quantity, row total. Configurable products send the child SKU as `id`, the parent SKU as `group_id` and the selected options as `variant_dict`

Customer data & privacy

When **Include Customer Data** is enabled the following is added to `user`, following the API's normalization rules:

Sent as	Source	Treatment
`emails_sha256`	Order email	Lowercased, trimmed, SHA-256
`phone_numbers_sha256`	Billing telephone	Needs a country calling code: only a leading `+` or `00` is accepted, SHA-256 over the 8–15 digits that follow. A national number such as `06 12345678` is left out rather than guessed at, because the `0` is a trunk prefix and not `+31`
`first_names_sha256`, `last_names_sha256`	Billing name	Lowercased, whitespace/punctuation removed, SHA-256
`external_ids_sha256`	Customer ID (registered customers)	SHA-256
`cities`, `regions`, `postal_codes`, `countries`	Billing address	Sent as-is (required by the API)
`ip_address`, `user_agent`	Checkout request	Sent as-is
`obref`	Pixel `__obref` cookie	Sent unchanged

Hashes are irreversible, but address, IP and user agent are personal data. Make sure your privacy policy covers sharing conversion data with OpenAI, or disable **Include Customer Data** (matching then relies on `oppref` / `obref` only).

Events

Select which events are sent under **Events**. Only *Purchase* is enabled by default. Every event carries the click attribution and, when enabled, the hashed customer data described below.

Event	OpenAI type	Magento trigger	Value reported	Deduplicated by
Purchase	`order_created`	Order placed (or paid, see below)	Order total in the order currency	Order increment ID
Checkout started	`checkout_started`	Checkout page opened	Cart total in the cart currency	Quote ID — reloading checkout does not resend
Add to cart	`items_added`	Product added to the cart	Line total of the added item	One event per addition
Product view	`contents_viewed`	Product page viewed	Product price in the base currency	One event per view
Page view	`page_viewed`	Any storefront page view	None	One event per view
Account registration	`registration_completed`	Storefront account created	None	Customer ID
Newsletter signup	`lead_created`	Subscriber becomes subscribed	None	Subscriber ID

`page_viewed` writes a queue row for **every** storefront page view, which on a busy shop means a very large table and a lot of API traffic for little attribution gain. The Measurement Pixel is the better tool for it; enable it server-side only if you have a specific reason. AJAX requests, POSTs and technical routes (customer section loads, cart sidebar, CAPTCHA, and similar) are never counted.

Only the purchase event honours **Send Purchase Conversion When** and **Purchase Conversion Value**; funnel events are queued as they happen and report the cart or product value.

The API itself also accepts `subscription_created`, `trial_started`, `appointment_scheduled`, `app_installed`, `app_opened` and `custom`. Magento has no equivalent for those, so this extension does not send them.

Events other than purchases store a snapshot of their body when queued, and that snapshot only ever contains **hashed** identity fields — no plain email addresses or names are written to the queue table.

Configuration

Location: Magmodules → OpenAI - Ads Conversions → Conversions API

Connection

Setting	Description
Enable	Enable conversion tracking for this store view
Pixel ID	Pixel ID from the Conversions tab in Ads Manager
API Key	Conversions API key belonging to the Pixel ID (stored encrypted)
Test Mode	Sends every event with `validate_only`: OpenAI validates the payload but records nothing. Disable after testing
Send Test Event	Posts a sample `validate_only` event to verify credentials and connectivity. Save the configuration first

Event Data

Setting	Description
Events	Which events are sent — see Events. Defaults to purchase only
Send Purchase Conversion When	*Order is placed* (default) or *Order is paid*. With the latter the order waits in the queue until an invoice is paid. The event timestamp is always the order date; OpenAI rejects events older than 7 days, so late payments are skipped
Purchase Conversion Value	Grand total (default), subtotal excl. tax or subtotal incl. tax
Only Send Attributed Events	*No* (recommended): send everything so OpenAI can match it via customer data, including view-through conversions. *Yes*: only events carrying an `oppref` or `obref`. Unattributed orders are still recorded as skipped for auditing; other unattributed events are not queued
Include Customer Data	Add hashed and raw matching data as described above

Click Attribution

Setting	Description
Capture oppref On Storefront	Adds the capture script to all storefront pages
Cookie Lifetime (days)	Lifetime of the `mm_openai_oppref` cookie. Match your campaign's click window (default 30)

Maintenance

Setting	Description
Keep History (days)	Processed queue entries older than this are removed by the daily cleanup cron (default 90, 0 = keep forever)

Using the Measurement Pixel too

You can run the OpenAI Measurement Pixel alongside the Conversions API (a "hybrid" setup). To avoid double counting, pass the order increment ID as the pixel's `event_id` for the `order_created` event:

```
oaiq("measure", "order_created", { ... }, { event_id: "000000123" });
```

OpenAI keeps the first event it receives for a given Pixel ID, event name and ID and ignores later duplicates. The extension also reads the pixel's `__oppref` and `__obref` cookies automatically.

Monitoring & Troubleshooting

Location: Marketing → OpenAI Commerce → Ads Conversions

Status	Meaning
Pending	Waiting for the next cron run (or retrying after a temporary failure)
Awaiting Payment	*Send Conversion When* is set to *Order is paid* and no invoice has been paid yet
Sent	Accepted by OpenAI (or validated, in test mode)
Failed	Rejected by the API or retries exhausted — see the Result column and the *Payload* action for the API response

Status	Meaning
Skipped	Not sent on purpose: canceled order, event type switched off after queueing, no attribution while *Only Send Attributed Events* is on, or event older than 7 days

Actions:

- **Send Pending Now** — process the queue immediately instead of waiting for cron.
- **Payload** — shows the exact JSON sent and the API response.
- **Send Now / Resend** — requeue and send again. Safe thanks to deduplication.

Common issues:

Problem	Solution
Everything stays *Pending*	Magento cron is not running. Check with the self-test under OpenAI - Feed Configuration → Debug
Failed with HTTP 401/403	Wrong Pixel ID or API key. Use **Send Test Event** after saving
Attributed is always *No*	The capture script is disabled, the cookie expired, or the ad landing URL does not carry `oppref`. Check the storefront source for `Magmodules_OpenAiCommerce/js/attribution`
Orders appear twice in Ads Manager	Make sure the pixel `event_id` equals the order increment ID

Enable **Debug Mode** (OpenAI - Feed Configuration → Debug) to log every request and response to `var/log/mm-openai-debug.log`.

CLI

```
# Send all pending conversions (same as the cron job)
bin/magento mm-openai:conversions:process
```

Need More Help?

Documentation:

- OpenAI Ads Conversions API reference
- All Help Articles

Support:

- Contact Support

For a complete overview of features and configuration options, see the OpenAI Commerce extension on magmodules.eu

All articles for OpenAI Commerce

Installation

1	Installation using Composer (recommended)
2	Installation using the Adobe Marketplace

Configuration

1	Quick Start Guide
2	llms.txt Guide
3	Configuration Guide

Troubleshooting

1	Troubleshooting

Other

1	**Ads Conversions API Guide (current)**

Background

1	Best Practices
2	CLI Commands

