magmodules



Configuration Guide

OpenAI Commerce for

Complete reference for all OpenAI Commerce settings. The extension generates JSON feeds following the OpenAI Commerce specification for AI-powered shopping platforms.

Location: Magmodules → OpenAI - Feed Configuration

General

Setting	Description
Enable	Master switch for the entire module

Generation

Controls feed generation per store view.

Setting	Description
Enable	Enable feed generation for this store view
Filename	Custom filename for the feed (default: openai.json)
Feed Format	`JSON (single document)` or `JSON Lines (one product per line)`. The filename extension follows this setting
Compress With Gzip	Writes the feed as `.gz`

The feed table shows all configured store views with their feed URLs and generation status.

Which format to pick

The single JSON document is what the OpenAI Commerce specification shows: one file with a `metadata` block and a `products` list, readable by eye. Pick it when you hand the feed to someone or read it yourself.

JSON Lines writes one product object per line and no `metadata` block, because a reader takes every line as a product. The seller fields from the metadata are written on each product as well, so nothing is lost. This is the shape OpenAI's merchant onboarding reads from a hosted URL, next to CSV, TSV and Parquet; a single JSON document cannot be submitted there. Combine it with **Compress With Gzip** for a large catalogue, which gives you a `.jsonl.gz` URL to register.

Seller Data

Configure your business information that appears in the feed. This helps AI platforms present your store correctly to users.

Setting	Description
Seller Name	Required. Shown to shoppers in ChatGPT, and written on every product in the feed
Privacy Policy	Select CMS page containing your privacy policy
Terms of Service	Select CMS page containing your terms
Return Policy	Select CMS page containing your return policy
Return Window	Number of days within which a product can be returned
Enable Checkout	Marks every product as eligible for checkout in ChatGPT (`enable_checkout`). Requires a checkout integration enabled by OpenAI, the policy pages above and search eligibility on the product. Default: No

Tip: Create dedicated CMS pages for each policy if you don't have them. AI platforms may reference these when answering customer questions.

Attributes

Map your Magento attributes to OpenAI Commerce fields.

Product Name & Description

Setting	Description
ID	Unique product identifier. Use SKU for consistency
Title	Product name shown in AI responses
Description	Product description. HTML is stripped automatically

Images & Videos

Setting	Description

Source	Which images to include: Base Image or All Images
Main Image	When using All Images, which image is the primary one
Include Hidden Images	Include images marked as hidden in gallery

Identifiers

Setting	Description
GTIN	EAN/UPC barcode attribute
Brand	Manufacturer or brand name attribute
MPN	Manufacturer Part Number attribute

Condition

Setting	Description
Condition	Static value or attribute-based: New, Refurbished, or Used

Category

Setting	Description
Include Category	Add product category path to the feed

Non-Required Product Fields

Optional attributes that improve product matching:

Setting	Description
Color	Product color attribute
Material	Product material attribute
Size	Product size attribute
Gender	Target gender attribute

Product Types

Configure how different Magento product types are exported.

Configurable Products

Setting	Recommended	Description
Use Configurable/Simple Products	Only Linked Simple Products	Export variants, not the parent
Use Parent URL for Simples	Yes, with Auto-Link	Links to correct variant on product page
Use Parent Image for Simples	Only if Empty	Use parent image as fallback
Use Parent Data for Simples	Name, Description	Inherit these from parent
Use Fallback only on Non Visible Simples	Yes	Only apply parent logic for hidden variants

Bundle Products

Setting	Recommended	Description
Use Bundle Products	Only Bundle Product	Export the bundle, not components
Use Parent URL for Simples	Yes	Link to bundle product page
Use Parent Image for Simples	Only if Empty	Use bundle image as fallback

Grouped Products

Setting	Recommended	Description
Use Grouped Products	Only Grouped Product	Export the grouped product
Parent Price	Configure pricing for grouped products	

Filter Options

Control which products appear in your feed.

Filter on Visibility

Setting	Description
Enable	Filter products by visibility setting
Visibility	Select which visibility levels to include

Filter on Status

Setting	Description
Add disabled products	Include disabled products in feed

Filter by Category

Setting	Description
Enable	Enable category filtering
Type of filter	Include or Exclude selected categories
Category	Select categories to include/exclude

Exclude Out of Stock

Setting	Description
Enable	Exclude out-of-stock products from feed

Advanced Filters

Setting	Description
Enabled	Enable advanced filtering
Filter Conditions	Add attribute-based filter rules

Automation

Cron

Setting	Description

Enable	Enable automatic feed generation
Frequency	How often to regenerate: hourly, daily, custom
Custom Frequency	Crontab expression for custom schedules

llms.txt

Generate an llms.txt Markdown file that guides AI crawlers and LLMs to your store's key content. Each store view gets its own file (e.g. `llms-default.txt`, `llms-nl.txt`), served automatically at the configured URL.

Location: Magmodules → OpenAI - LLM Configuration → llms.txt

Setting	Description
Enable	Enable llms.txt generation for this store view. The module's master switch (Feed → General → Enable) has to be on as well
Files	Overview table showing per-store file status, with Generate and View actions
Filename	The public URL path (default: `llms.txt`). The physical file uses a store-code suffix
Title	H1 heading in the file. Leave empty to use the store name
Description	Optional blockquote summary below the title
Include Categories	Add category links from the navigation menu
Include CMS Pages	Add links to CMS pages
Link Product Feed	Include a link to the generated OpenAI Commerce product feed

llms.txt Cron

Setting	Description
Enable	Enable automatic llms.txt regeneration
Frequency	How often to regenerate: daily, custom
Custom Frequency	Crontab expression for custom schedules

Ads Conversions API

Send purchase conversions server-side to the OpenAI Ads Conversions API. See the Ads Conversions API Guide for details on attribution, payload and privacy.

Location: Magmodules → OpenAI - Ads Conversions → Conversions API

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 (stored encrypted)
Test Mode	Send events with `validate_only`; nothing is recorded by OpenAI
Send Test Event	Verify credentials and connectivity with a sample event
Events	Which events to send: purchase, checkout started, add to cart, product view, page view, account registration, newsletter signup. Defaults to purchase
Send Purchase Conversion When	Order is placed (default) or order is paid
Purchase Conversion Value	Grand total, subtotal excl. tax or subtotal incl. tax
Only Send Attributed Events	Restrict to events with an OpenAI click identifier
Include Customer Data	Add SHA-256 hashed email/phone/name/customer ID plus address, IP and user agent for matching
Capture oppref On Storefront	Store the `oppref` landing parameter in a first-party cookie
Cookie Lifetime (days)	Lifetime of the attribution cookie (default 30)
Keep History (days)	Retention of processed queue entries (default 90)

Delivery results are shown under Marketing → OpenAI Commerce → Ads Conversions.

Debug

Setting	Description
Debug Mode	Enable detailed logging for troubleshooting

Logs are written to `var/log/`.

Need More Help?

Documentation:

- All Help Articles - Complete documentation overview

Support:

- Contact Support - Get help from our team

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 (current)**

Troubleshooting

1	Troubleshooting

Background

1	Best Practices
2	CLI Commands

