

# Product Feed

## **Magento 2 Module — Complete User Guide**

*Complete Guide — Every Screen, Every Setting, Step by Step*

Generate a Google Shopping / Merchant Center product feed for your Magento store — field-mapped, price- and stock-aware, on demand or on a schedule.

Prepared by: eTechFlow Pvt. Ltd.    Contact: [etechflow0@gmail.com](mailto:etechflow0@gmail.com)    Date: July 2026

## Table of Contents

1. Introduction
2. Open the Product Feed Configuration
  - 2.1 Open Stores → Configuration → eTechFlow → Product Feed
3. Configuration
  - 3.1 General
  - 3.2 Schedule & Field Mapping
  - 3.3 Pricing & Product Filter
  - 3.4 Local Inventory, Health & Advanced
4. Generate & View the Feed
  - 4.1 Generate Now
  - 4.2 The generated feed (GoogleShopping.xml)
  - 4.3 The Local Inventory feed
  - 4.4 Cron & CLI generation
5. Support & Contact

## 1. Introduction

Product Feed generates search-engine and marketplace product feeds for your Magento store — primarily a Google Shopping / Google Merchant Center feed listing every sellable product with the fields Google expects (id, title, link, image, price, availability, brand, GTIN, MPN, condition and Google product category). It can also produce a second Local Inventory feed tagged with your Google Business Profile store code for Local Inventory Ads.

You control exactly which products are included (in-stock only, visible only, priced only), how prices are output (with or without tax, with sale prices), and which product attributes map to each feed field. The feed can be rebuilt on demand from the admin, automatically every night by cron, or from the command line, and a Drop Protection guard can abort a run that suddenly shrinks so a broken feed never reaches Google.

The module is licensed through the eTechFlow portal. Until a valid key is entered for your domain, the feed is not generated. This guide covers configuring every setting and generating and viewing the feed.

## 2. Open the Product Feed Configuration

Everything for Product Feed lives under Stores → Configuration → eTechFlow → Product Feed. Open it to configure the feed.

### 2.1 Open Stores → Configuration → eTechFlow → Product Feed

In the Magento admin, click Stores in the left sidebar.



Figure 1 — Click Stores in the left admin sidebar.

Under the Settings column, click Configuration.

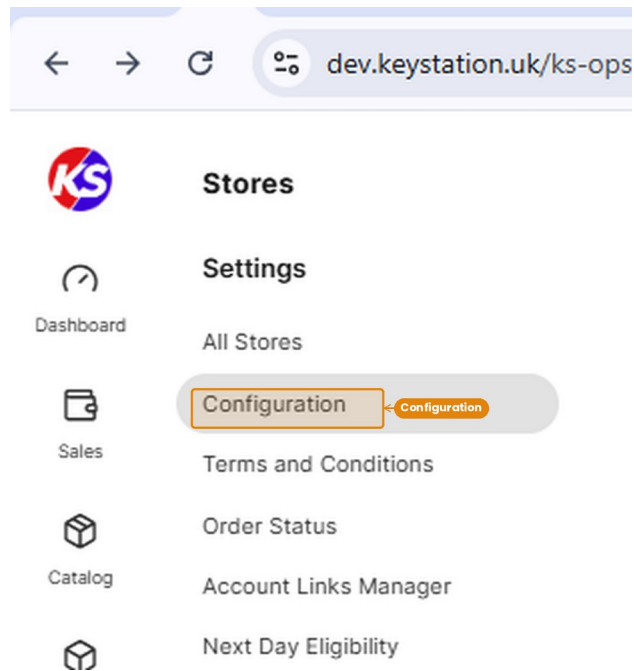


Figure 2 — Stores menu → Settings → Configuration.

In the configuration list on the left, find and expand the eTechFlow group.

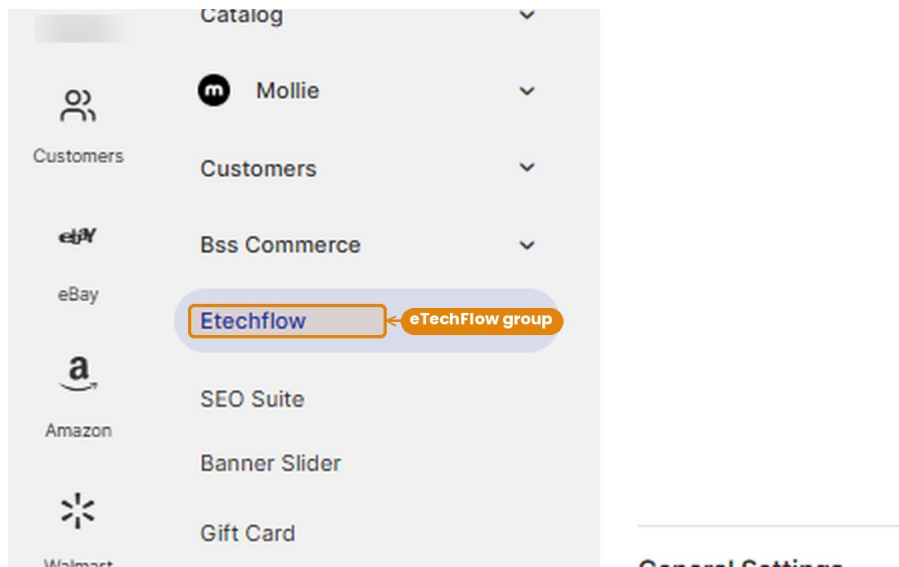


Figure 3 — Open the eTechFlow group in the left-hand section list.

Inside the eTechFlow group, click Product Feed to open the module's settings.

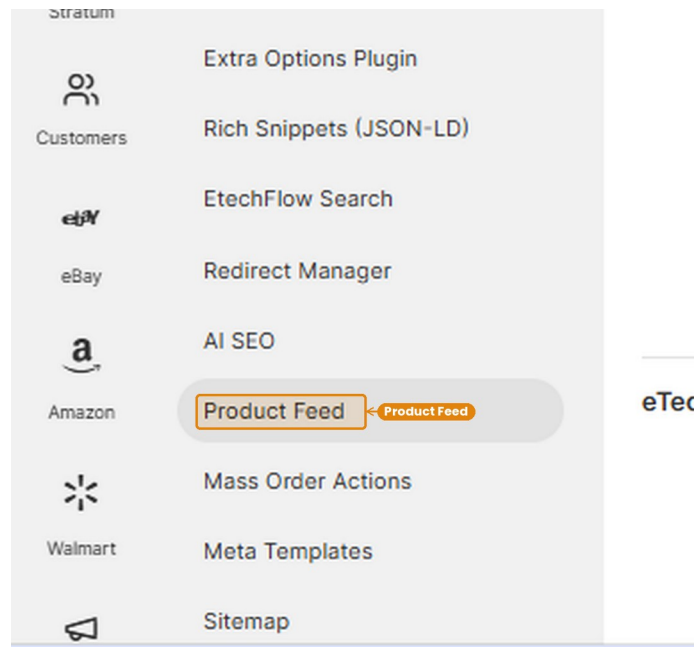


Figure 4 — Click Product Feed inside the eTechFlow group.

### 3. Configuration

With the licence active, configure how the feed is built. The settings are grouped into collapsible sections down the page — General, Schedule, Field Mapping, Pricing, Product Filter, Local Inventory Feed, Health & Drop Protection and Advanced.

#### 3.1 General

General

Enable Feed

store view

Yes

▼

Enable Feed

Master switch. When off, no feed is generated and the cron is idle.

Feed File Name

store view

etech-test-feed.xml

Feed File Name

Written to pub/media/feed/. Served at {base\_url}media/feed/{filename}.

Feed Title

store view

Product Feed

Feed Description

store view

Product feed

Batch Size

store view

500

Products processed per chunk. Keeps memory flat and avoids saturating php-fpm on big catalogues.

Generate Now

Generate Now

Generate Now

Generate immediately. For large catalogues prefer the cron/CLI to avoid a long admin request.

Figure 6 — General: the master switch, feed file name, title/description and batch size, plus the Generate Now button.

[ **Enable Feed** ] The master switch. When set to No, no feed is generated and the scheduled cron stays idle. Set to Yes to turn the module on.

[ **Feed File Name** ] The output file name. The feed is written to pub/media/feed/ and served at {base\_url}media/feed/{filename} (e.g. GoogleShopping.xml).

[ **Feed Title** ] The <title> shown at the top of the feed (your store / feed name).

[ **Feed Description** ] The <description> for the feed channel.

[ **Batch Size** ] How many products are processed per chunk. Keeps memory flat and avoids saturating php-fpm on large catalogues (default 500). This affects performance only, not the output.

[ **Generate Now** ] Builds the feed immediately and reports how many items were written. For very large catalogues, prefer the cron or CLI to avoid a long admin request.

### 3.2 Schedule & Field Mapping

The screenshot shows the Magento 2 admin interface for the feed configuration. The 'Schedule' section is at the top, with 'Generate on a Schedule' set to 'Yes' and 'Cron Expression' set to '0 4 \* \* \*'. Below this is the 'Field Mapping' section, which shows 'Product id Source' as 'SKU', 'Brand Attribute' as 'manufacturer', and 'GTIN Attribute' and 'MPN Attribute' as blank. An orange box highlights the 'Schedule + cron' button and another orange box highlights the 'Field mapping' button.

Figure 7 — Schedule (auto-generate on a cron) and the top of Field Mapping (id source, brand, GTIN, MPN).

#### Schedule

[ **Generate on a Schedule** ] When Yes, the feed is rebuilt automatically by Magento cron on the schedule below.

[ **Cron Expression** ] Standard crontab format, e.g. 0 4 \* \* \* for 04:00 daily.

Field Mapping — which product data fills each feed field:

[ **Product id Source** ] What becomes <g:id> — the product SKU or entity ID. Must match the id you send to GTM / GA4 as item\_id so analytics and the feed line up.

- [ **Brand Attribute** ] The product attribute code used for <g:brand> (e.g. manufacturer).
- [ **GTIN Attribute** ] The attribute code for <g:gtin> (EAN / UPC). Leave blank if you have none.
- [ **MPN Attribute** ] The attribute code for <g:mpn> (manufacturer part number). Leave blank if none.
- [ **Condition** ] The value emitted as <g:condition> — new, used or refurbished.
- [ **Google Category Attribute** ] The attribute holding the Google product category. Falls back to the default below when a product has no value.
- [ **Default Google Category** ] Used for <g:google\_product\_category> when a product has no category attribute value (e.g. Apparel & Accessories > Clothing).

### 3.3 Pricing & Product Filter

Default Google Category [store view]

Used when a product has no category attribute value.

---

**Pricing** ^

Prices Include Tax [store view] Yes

Output Sale Price [store view] Yes

Only Emit Valid Sale Prices [store view] Yes

Drop g:sale\_price when it is not lower than the regular price (Google rejects those).

**Product Filter** ^

In-Stock Products Only [store view] Yes

Visible Products Only [store view] Yes

Require Price > 0 [store view] Yes

exclude products with visibility "Not Visible Individually".

Figure 8 — Pricing (tax and sale-price handling) and Product Filter (which products are included).

#### Pricing

- [ **Prices Include Tax** ] When Yes, <g:price> is output tax-inclusive; when No, it is the pre-tax price.
- [ **Output Sale Price** ] When Yes, products on special also emit a <g:sale\_price>.
- [ **Only Emit Valid Sale Prices** ] Drops <g:sale\_price> when it is not actually lower than the regular price — Google rejects "sales" that are not discounts.

Product Filter — which products make it into the feed:

- [ **In-Stock Products Only** ] Excludes out-of-stock products from the feed.
- [ **Visible Products Only** ] Excludes products whose visibility is "Not Visible Individually".

[ **Require Price > 0** ] Excludes products with no price (or a zero price).

### 3.4 Local Inventory, Health & Advanced

#### Local Inventory Feed

A second feed of in-store stock for Local Inventory Ads, tagged with your Google Business Profile store code.

The screenshot shows three sections of a settings interface. The first section, 'Local Inventory Feed', has a title 'Enable Local Feed' with a sub-label '[store view]'. It contains a dropdown menu set to 'No' and an orange button labeled 'Local Feed'. The second section, 'Health & Drop Protection', has a title 'Enable Drop Protection' with a sub-label '[global]'. It contains a dropdown menu set to 'No' and an orange button labeled 'Drop Protection'. The third section, 'Advanced', has a title 'Debug Logging' with a sub-label '[global]'. It contains a dropdown menu set to 'No' and an orange button labeled 'Debug Logging'. Below the 'Debug Logging' dropdown, there is a small text note: 'Verbose generation logging to var/log/etechflow\_productfeed.log.'

Figure 9 — The Local Inventory Feed, Health & Drop Protection, and Advanced sections.

Local Inventory Feed — an optional second feed of in-store stock for Local Inventory Ads, tagged with your Google Business Profile store code:

[ **Enable Local Feed** ] Turns on the second feed. When enabled you also set its file name and Store Code (the fixed Google Business Profile store code, emitted as `<g:store_code>` on every item).

Health & Drop Protection — catch a feed that suddenly shrinks before Google does:

[ **Enable Drop Protection** ] When on, a run is warned/aborted if the new item count falls more than the Drop Threshold percent below the previous run, and a Notification Email is sent — so a broken, gutted feed never ships.

Advanced:

[ **Debug Logging** ] Verbose generation logging to `var/log/etechflow_productfeed.log` — useful when diagnosing missing products or field values.

## 4. Generate & View the Feed

### 4.1 Generate Now

After saving your settings, click the Generate Now button in the General section (Section 3.1). The module builds the feed immediately and shows an inline confirmation such as “Main feed generated: 114 items.” — plus the local feed count when the Local Inventory feed is enabled.

For very large catalogues, use the cron or CLI instead (Section 4.4) so the admin request does not time out.

## 4.2 The generated feed (GoogleShopping.xml)

Open the feed in a browser at {base\_url}media/feed/{filename} — for example <https://your-store/media/feed/GoogleShopping.xml>. It is a standard Google Shopping RSS feed: a <channel> with your title and description, then one <item> per product carrying the g: fields you mapped.

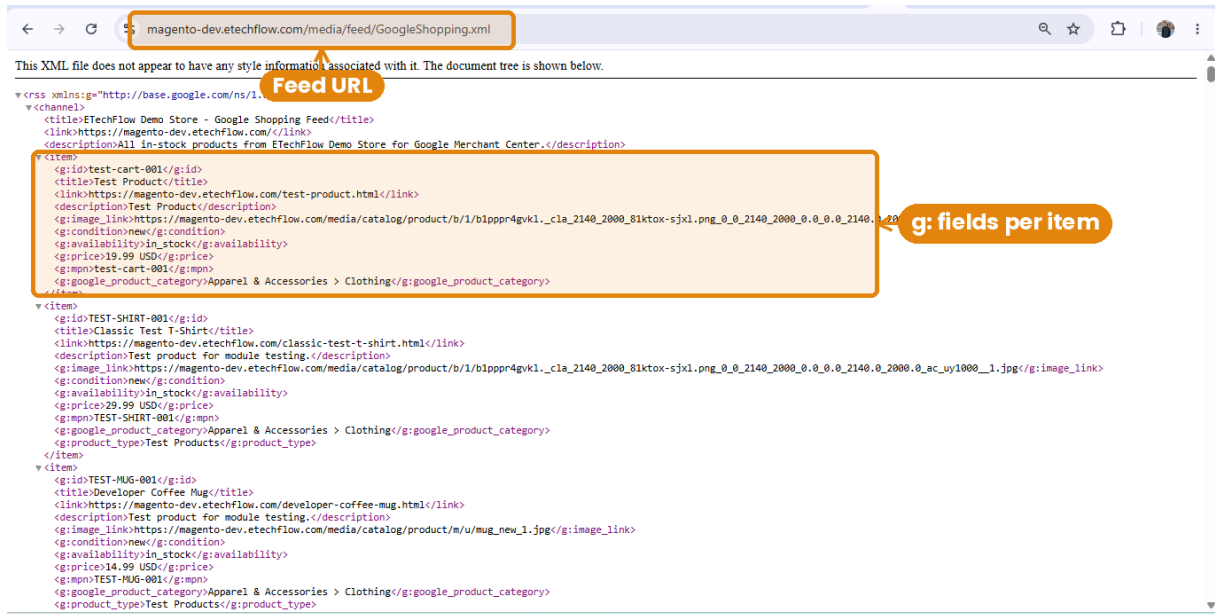


Figure 10 — The generated GoogleShopping.xml: each <item> carries g:id, g:title, g:link, g:image\_link, g:condition, g:availability, g:price, g:mpn and g:google\_product\_category.

Each item reflects your configuration: <g:id> is the SKU (id source), <g:condition> is your chosen condition, <g:price> follows the pricing rules, <g:google\_product\_category> uses the mapped or default category, and only products passing the Product Filter appear. Paste this URL into Google Merchant Center to import it.

### 4.3 The Local Inventory feed

When the Local Inventory Feed is enabled, a second file is written (e.g. GoogleShoppingLocal.xml). It is the same product list, with a <g:store\_code> added to every item for Local Inventory Ads.

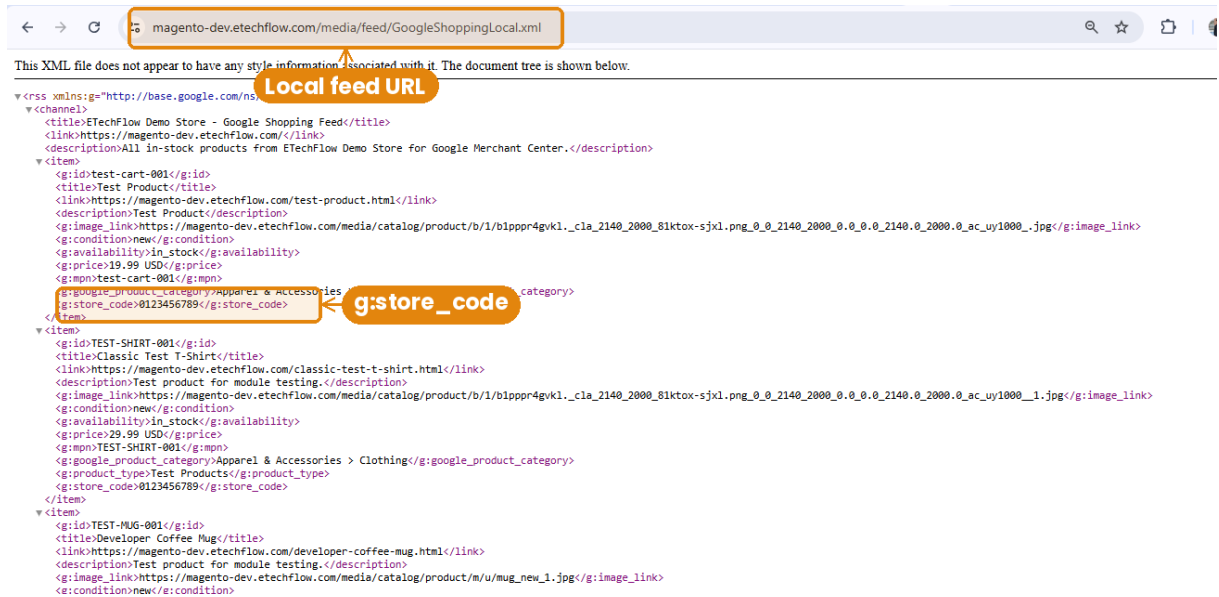


Figure 11 — The Local Inventory feed: every `<item>` includes `<g:store_code>` with your Google Business Profile store code.

## 4.4 Cron & CLI generation

You do not have to click Generate Now every time. With Generate on a Schedule enabled (Section 3.2), Magento cron rebuilds the feed automatically on your cron expression. You can also generate it from the command line in your Magento project root:

```
bin/magento etechflow:productfeed:generate
```

Add `--store=<id>` to generate for a specific store view. If the feed is disabled for that store, the command reports it and does nothing. This is the recommended way to generate feeds for very large catalogues.

## 5. Support & Contact

For help with Product Feed, licensing, or installation, contact eTechFlow support. We respond within one business day.

**Email:** [support@etechflow.com](mailto:support@etechflow.com)

**Website:** <https://etechflow.com>