



Loyalty Program Feed

Google Shopping for Magento 2

How to add Google Merchant Center loyalty program pricing to your feed with the [Google Shopping Feed](#) extension. Show member-only prices and loyalty points on your Shopping ads so logged-in members see the deals they qualify for. This guide explains what the loyalty program node is, how the configuration grid works, and how to map your member pricing to product attributes.

What Is a Loyalty Program in Google Shopping?

Google Merchant Center lets you advertise member-only benefits next to your regular Shopping ads. When a product carries loyalty data, the feed includes a `<g:loyalty_program>` node that tells Google:

- The **program** and **tier** a benefit belongs to (for example "Silver" members)
- A **member price** that differs from the regular price
- The number of **loyalty points** the product earns
- The **date range** the member price is valid for

Members who are signed in to your loyalty program then see the member price and points directly in Shopping results.

Requirements

Before the feed data has any effect:

1. **Google Merchant Center account** with products approved
2. **Loyalty program set up in Merchant Center** - Under **Growth** → **Loyalty program**, create your program and its tiers
3. **Matching labels** - The program and tier names you enter in Magento must match exactly what you configured in Merchant Center
4. **Product data** - The products that should carry a benefit need attributes holding the member price and/or loyalty points

For Google's policy and field reference, see [Google's loyalty program documentation](#).

How It Works

The loyalty feature uses a configuration grid. Each row defines one loyalty tier and is split into two kinds of fields:

- **Static labels** - The Program Label and Tier Label are typed once and are the same for every product the row applies to.
- **Mapped attributes** - The member price, loyalty points and effective date are read **per product** from the attribute you choose, so each product can carry its own values.

A `<g:loyalty_program>` node is only added to a product when that product actually has a benefit - that is, it has a value in the mapped **member price** or **loyalty points** attribute. Products with no member benefit are left untouched, so you can configure a single row for your whole catalog and only the products with member pricing get the node.

Configuration

Location: Stores → Configuration → Magmodules → Google Shopping → Additional Configuration → Loyalty Program

This setting is per store view, so configure it on the same scope you generate the feed for.

Click **Add** to create a row and fill in the columns:

Column	What it is	Goes into feed as
Program Label	The program ID exactly as set in Merchant Center	<code>g:program_label</code>
Tier Label	The tier ID exactly as set in Merchant Center	<code>g:tier_label</code>
Member Price Attribute	Product attribute holding the member price	<code>g:price</code>
Loyalty Points Attribute	Product attribute holding the loyalty points	<code>g:loyalty_points</code>
Member Price Effective Date Attribute	Product attribute holding the validity range	<code>g:member_price_effective_date</code>

Program Label and Tier Label are required - rows missing either are dropped when you save. Add more rows if you run multiple tiers (for example Silver and Gold), then **Save Config**.

Preparing Product Attributes

The price, points and date columns reference existing product attributes by their code. Decide which attribute holds each value, then make sure those attributes exist and are populated on the relevant products.

Value	Suggested attribute type	Notes
-------	--------------------------	-------

Member price	Price or decimal	The raw number, for example <code>19.99</code> . The currency is added automatically.
Loyalty points	Text or integer	Exported as-is, for example <code>150</code> .
Effective date	Text	Use Google's date-range format (see below).

You can reuse stock attributes (for example `cost`) or create dedicated ones under **Stores → Attributes → Product**. Only products that have a value in the member price or loyalty points attribute receive a loyalty node.

Effective Date Format

The effective date is a start/end range separated by a slash, using ISO 8601 timestamps:

```
2025-01-01T00:00:00+0000/2025-12-31T23:59:59+0000
```

The value is exported exactly as stored, so enter it in this format on the product.

Example Feed Output

A product with member pricing produces a node like this inside its item:

```
<g:loyalty_program>
  <g:program_label>my_loyalty_program</g:program_label>
  <g:tier_label>silver</g:tier_label>
  <g:price>19.99 EUR</g:price>
  <g:loyalty_points>150</g:loyalty_points>
  <g:member_price_effective_date>2025-01-01T00:00:00+0000/2025-12-31T23:59:59+0000</g:member_price_effective_date>
</g:loyalty_program>
```

The member price is formatted with the store's feed currency automatically. If a product has only points (no member price), the `g:price` line is omitted, and the other way around - only the values that exist are written.

Best Practices

Do's

- ☐ Match the Program Label and Tier Label to Merchant Center exactly, including case
- ☐ Store the member price as a plain number and let the feed add the currency

- Use a dedicated attribute for member price so it never clashes with regular pricing
- Keep the effective date in the slash-separated ISO 8601 range format
- Configure the grid on the store view you actually export

Don'ts

- Don't include a currency symbol in the member price attribute
- Don't expect a node on products that have no member price or points - that is by design
- Don't leave Program Label or Tier Label blank, the row will be dropped on save
- Don't rename the mapped attribute without updating the grid

Common Scenarios

Single Tier for the Whole Catalog

Use case: One membership level, member prices set on a handful of products.

Configuration:

- Program Label: your program ID
- Tier Label: your single tier ID
- Member Price Attribute: your member price attribute

Result: Every product that has a member price gets a node; everything else is skipped.

Multiple Tiers

Use case: Silver and Gold members with different pricing.

Configuration:

- Row 1: Tier Label `silver` , mapped to a silver-price attribute
- Row 2: Tier Label `gold` , mapped to a gold-price attribute

Result: Products carrying values for both attributes get two loyalty nodes, one per tier.

Troubleshooting

No Loyalty Node in the Feed

Symptoms: Products you expected to carry a benefit have no `<g:loyalty_program>` node.

Solution:

1. Confirm the product has a value in the mapped member price or loyalty points attribute - a node is only emitted when one of these exists
2. Check the configuration was saved on the store view you generate the feed for
3. Flush the configuration cache and regenerate the feed
4. Verify the attribute is included in the attribute set assigned to the product

Prevention: Populate the member price attribute before generating the feed.

Member Price Missing but Points Show

Symptoms: The node appears with points but no `g:price`.

Solution: The member price attribute is empty for that product. Add a numeric value to it. The price line is only written when a member price exists.

Merchant Center Rejects the Loyalty Data

Symptoms: Loyalty pricing is disapproved or ignored in Merchant Center.

Solution:

1. Make sure the Program Label and Tier Label match your Merchant Center program and tier exactly
2. Confirm the loyalty program is active in Merchant Center
3. Check the effective date uses the correct ISO 8601 range format

Wrong Currency on the Member Price

Symptoms: `g:price` shows an unexpected currency.

Solution: The currency follows the store's feed currency. Store only the number in the attribute and let the feed format it; adjust the store currency if needed.

Loyalty Program Checklist

Before submitting:

- Loyalty program and tiers created in Merchant Center
- Program Label and Tier Label match Merchant Center exactly
- Member price attribute populated with plain numbers
- Loyalty points attribute populated where relevant
- Effective date uses the slash-separated ISO 8601 range

- Configuration saved on the correct store view
- Feed regenerated and node verified

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 Google Shopping extension on magma.modules.eu

All articles for Google Shopping

Installation

1	Installation using Composer (recommended)
2	Installation using the Adobe Marketplace
3	Install through FTP and SSH
4	Update a Magento 2 Module using an Artifact Repository

Configuration

1	Quick Start Guide
2	Configuration Guide
3	Attribute Mapping
4	Product Types
5	Filters
6	SFTP Setup
7	Promotions Feed
8	Local Inventory Feed
9	Loyalty Program Feed (current)

Troubleshooting

1	Google Merchant Center Errors
2	CLI Commands
3	Google Merchant API vs Feed Upload

Background

1	Identifier Exists Attribute
2	Google Product Categories
3	Feed Preview
4	Best Practices
5	Troubleshooting
6	Google Merchant Center Setup

