



Common Issues

Pricesystem Documentation

Guide for Magento administrators, technical project owners, and implementation teams.

LAST UPDATED
2026-09-26

SOURCE
[Open this article online](#)

EXTENSION
[Magento 2 Extension Pricesystem](#)

Common Issues

Start with one customer, one SKU and one quantity. Large grids and storefront pages add noise before the calculation itself is understood.

A custom price is not used

Check, in this order:

1. Price System is active for the product's website.
2. The price record is assigned to the current customer, group or matching rule.
3. The website and date range apply now.
4. The requested quantity reaches the intended break.
5. The price application type has a valid base price when it is relative.
6. The global or customer-specific selection strategy allows that price type to win.
7. The Price System cache has been cleaned after the change.

Inspect one calculation

List the available price types:

```
php bin/magento pricesystem:get-final-price --list-price-types
```

Calculate one SKU for one customer:

```
php bin/magento pricesystem:get-final-price \  
  --customer-id=123 \  
  --product-sku=24-MB01 \  
  --qty=10
```

Use IDs and SKUs from your own test environment. The command can also accept a date and an explicit price type.

Capture a debug log

Debug logging is off by default. Turn it on only while you investigate a specific price calculation:

1. In the Magento Admin, open **Stores > Configuration > MageB2B > Pricesystem > Debug**.
2. Set **Enable log** to **Yes** and save the configuration. The underlying configuration path is `pricesystem/debug/enable_log`.
3. Reproduce the problem with one customer, one SKU and one quantity.
4. Review `var/log/pricesystem.log` for the price types considered, skipped or

selected during that request.

5. Set **Enable log** back to **No** when you have captured the calculation.

The log can contain a customer ID or **Guest**, the product SKU and the values used in pricing decisions. Treat it as store data when you share it with support, and include only the entries needed for the calculation you are investigating.

Warm the cache

Use a dry run before a broad warm-up:

```
php bin/magento pricesystem:cache:warmup \  
  --customer-id=123 \  
  --product-sku=24-MB01 \  
  --dry-run
```

The command also supports customer groups, websites, dates, chunk sizes and an all-records mode. Do not start with **--all** on a large catalog.

Storefront prices do not refresh

- Check the browser network response for the Price System AJAX request.
- Confirm the theme keeps Magento price-box identifiers intact.
- Review configured skipped routes and chunk size.
- For Hyvä, confirm the Price System Hyvä package is installed and enabled.
- Clean both Magento full-page cache and **pricesystem_cache**.

Useful evidence for support

Include the SKU, customer ID or group, website, quantity, calculation date, expected price type and the output of the diagnostic command. Do not send customer passwords, access tokens or an unrestricted database export.