Migrating from Itonomy_Katanapim

KatanaPIM for Magento 2

How to migrate from the old Itonomy_Katanapim module (v2.x) to KatanaPIM Connect. This guide walks through the full upgrade process, including transferring your existing attribute mappings so you don't have to redo them manually.

Before You Start

Make sure you have:

- A full database backup
- The old module version (itonomy/katanapim v2.x) currently installed
- Access to the Magento CLI

Important: Running `setup:upgrade` after disabling the old module will drop its database tables. If you want to migrate your existing attribute mappings, you must backup the old table **before** disabling the module.

Step 1: Backup the Old Attribute Mapping Table

If you want to migrate your existing attribute mappings to KatanaPIM Connect, backup the old table before proceeding. Running `setup:upgrade` with the old module disabled will remove its tables.

```
mysqldump -u <db_user> -p <db_name> katanapim_attribute_mapping > katanapim_attribute_mapping_backup.sql
```

If you do not need to migrate attribute mappings (for example, if you prefer to set them up fresh), you can skip this step and go directly to Step 2.

Step 2: Disable the Old Module

The old module must be disabled before enabling the new one. Both modules use the same admin routes and menu entries, which causes conflicts if both are active at the same time.

```
bin/magento module:disable Itonomy_Katanapim
bin/magento cache:flush
```

Step 3: Install KatanaPIM Connect

Install and enable the new module:

```
composer require katanapim/connect
bin/magento module:enable KatanaPim_Connect
bin/magento setup:upgrade
bin/magento cache:flush
```

Note: The `setup:upgrade` command will drop the old module's database tables. If you skipped Step 1 and still need the old data, restore from your database backup before continuing.

Step 4: Re-save the API Key

The old module stores the API key as plain text. KatanaPIM Connect stores it encrypted. After enabling the new module:

1. Go to **Stores → Configuration → Katana PIM → General → API**
2. The URL is shared and will carry over automatically
3. Re-enter the API Key and **Save Config**

This ensures the key is stored in the encrypted format that the new module expects.

Step 5: Run an API Import

Before migrating attribute mappings, the new module needs to know about your attributes. Run a full attribute import:

```
bin/magento katana:import:full --attributes --only-api
```

This fetches all attributes from the KatanaPIM API into the new `katanapim_attribute` table.

Step 6: Restore and Migrate Attribute Mappings

If you backed up the old attribute mapping table in Step 1, restore it now:

```
mysql -u <db_user> -p <db_name> < katanapim_attribute_mapping_backup.sql
```

Now transfer your Magento attribute mappings and configurable flags from the old table:

Preview first (dry run):

```
bin/magento katana:migrate:attributes --dry-run
```

This shows which attributes will be updated without making changes. Only attributes that have a Magento attribute code mapped or `is_configurable` set in the old module are migrated.

Apply the migration:

```
bin/magento katana:migrate:attributes
```

The command updates existing rows in the new table with:

- **Magento Attribute Code** — the mapped Magento attribute
- **Magento Attribute ID** — the linked attribute entity ID
- **Is Configurable** — whether the attribute is used for configurable products

Attributes that exist in the old table but not in the new table are skipped. If you see skipped attributes, run the API import (Step 5) first.

After a successful migration, you can drop the restored table:

```
mysql -u <db_user> -p <db_name> -e "DROP TABLE katanapim_attribute_mapping;"
```

If you did not backup the table in Step 1, skip this step and configure your attribute mappings manually in **Katana PIM → Attributes**.

Step 7: Verify the Migration

1. Go to **Katana PIM → Attributes** in the admin
2. Check that your Magento attribute mappings are present
3. Check that configurable flags are set correctly
4. Run a test sync to verify everything works:

```
bin/magento katana:import:full --attributes
```

Step 8: Remove the Old Module

After confirming the new module works correctly in production:

```
composer remove itonomy/katanapim itonomy/module-database-logger
bin/magento setup:upgrade
bin/magento cache:flush
```

What Gets Migrated

Data	Migrated?	How
Attribute mappings (Magento code + ID)	Yes	`katana:migrate:attributes` command
Is Configurable flags	Yes	`katana:migrate:attributes` command
API URL	Yes	Shared config path, carries over automatically
API Key	Manual	Must be re-saved (Step 4)
Products	No	Re-imported from API with full sync
Categories	No	Re-imported from API with full sync
Assets	No	Re-imported from API with full sync

Troubleshooting

"Source table does not exist"

The old module's tables were dropped during `setup:upgrade`. If you have a backup from Step 1, restore it with `mysql -u <db_user> -p <db_name> < katanapim_attribute_mapping_backup.sql` and run the migration command again. Otherwise, configure your attribute mappings manually.

Attributes skipped during migration

The attribute exists in the old table but not in the new one. Run `bin/magento katana:import:full --attributes --only-api` first to populate the new table from the API.

API connection fails after migration

The API key needs to be re-saved via the admin (Step 4). The old module stored it as plain text, the new module expects it encrypted.

Duplicate admin menu entries

Both modules register a "Katana PIM" menu. This is normal while both are enabled. The old module should be disabled first (Step 2) before enabling the new one to avoid this.

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 KatanaPIM extension on magmodules.eu

All articles for KatanaPIM

Installation

1	Installation using Composer (recommended)
2	Install through FTP and SSH

Configuration

1	Quick Start Guide
2	Configuration
3	Categories Grid
4	Attribute Mapping
5	Configurable Products Setup
6	Automation

Troubleshooting

1	Troubleshooting

Other

1	Category to Store Coupling
2	**Migrating from Itonomy_Katanapim (current)**

Grids

1	Stores Grid

2	Products Grid
3	Attributes Grid
4	Assets Grid
5	Sync Log Grid

Background

1	CLI Commands
2	Cron Jobs
3	Product to Website Matching
4	Understanding the Sync Process

