

Sqquid Catalog Sync for Magento 2

Sqquid – September 2017

Introduction

[Sqquid](#) enables stores to connect their catalog between their POS and their online stores. Effectively sharing your inventory between your brick and mortar and your online sales channels.

What is Sqquid?

Sqquid enables online retailers to offer true multi-channel across their brick and mortar stores and their web ecommerce site(s).

Sqquid supports integrations with POS and ERP systems such as RunIt POS, Retail Pro, and Sage. Once integrated to the retailer POS/ERP in the backend, Sqquid can help manage the product catalog across multiple web sites powered by Magento 1 and/or Magento 2.

Learn more: Sqquid.com

Installation Instructions:

Installing with Composer

It's a good way to stay up to date with the latest version.

1. Run:
`composer require sqquid/module-sync`
2. Update Magento2:
`php bin/magento update`
3. Setup and install the extension:
`php bin/magento setup:upgrade`
4. Clear the cache:
`php bin/magento cache:flush`

5. Compile:
`php bin/magento setup:di:compile`

Manual Installation

To install the extension manually follow these instructions:

1. Download the files from Git:
`https://github.com/Sqquid/Magento2-CatalogSync`
2. Copy the extension files to the following folder on the server:
`[webroot]/app/code/Sqquid/Sync/`
3. Setup and install the extension:
`php bin/magento setup:upgrade`
4. Clear the cache:
`php bin/magento cache:flush`
5. Compile:
`php bin/magento setup:di:compile`

Installation Video

Here's a short video on how to install our Sqquid Sync extension in under 5 minutes: [Sqquid Sync Installation on Magento 2 Video](#)

Uninstall

Uninstalling with Composer

Please follow the below steps:

1. Run the uninstall command:
`composer remove sqquid/module-sync`
2. Clear the cache:
`php bin/magento cache:flush`
3. Compile:
`php bin/magento setup:di:compile`

Uninstalling Manually

Please follow the below steps:

1. Delete the extension files under the following folder:
[webroot]/app/code/Squid/Sync/
2. Clear the cache:
php bin/magento cache:flush
3. Compile:
php bin/magento setup:di:compile

Usage:

Once the extension is installed, visit the admin panel of your store and head over to the System Configuration. Go to the Squid section. Enable the squid integration. See below:

Configuration

🔍 🔔 👤 ron ▾

Store View: Default Config ▾ ⓘ Save Config

SQUID ^

General Setting

GENERAL ▾

CATALOG ▾

CUSTOMERS ▾

SALES ▾

SERVICES ▾

ADVANCED ▾

Integration Setup ⌵

Enabled [store view] Yes ▾

Tax Class ⌵

Assign Tax Class [store view] None ▾

Overwrite Tax Class [store view] No ▾

Category Settings ⌵

Select Category [store view] Default Category ▾

Visibility ⌵

Visibility [store view] Catalog, Search ▾

Overwrite Visibility on Resynch [store view] No ▾

Advanced ⌵

Processing Batch Size [store view] 250

Log File name [store view] squid_sync.log

Go into the 'System' > 'Permissions' > 'All Users' and add an API user with full access. Make sure to assign it to a role with permissions. Give it a safe password and keep it handy. See Below:

Users

🔍 🔔 👤 ron ▾

Add New User

Search Reset Filter 2 records found

20 per page 1 of 1

ID	User Name	First Name	Last Name	Email	Status
1					Active
2	sqquid_api	Squid	Squid	sqquid@sqquid.com	Active

← Back Delete User Reset Force Sign-In Save User

USER INFORMATION

User Info

User Role

Search Reset Filter 1 records found

20 per page 1 of 1

Assigned	Role
Any	
<input checked="" type="radio"/>	Administrators

Setup an Account on Sqquid.com:

head over to app.sqquid.com and create an account. Once you are logged in, head over to 'Integrations' and add a Magento 2 integration. Add your store URL, the new API user/pass you just created, enable and save. See below:

Magento 2

Enabled?

Name

Magento Store Url
Example: https://www.mymagentoURL.com

Create a web services user on Magento Admin by selecting System > All Users > Add New User. For now give this user admin privileges. (we will restrict this privileges in the future.)

Web Services Username

Web Services Password

Product Batch #
How many to export at one time?

[Save!](#)

After you save, the Integrations page should look like this (in this screenshot, we also have RunIt POS setup – you may need to do that separately):

[Home](#) / [Integrations](#)

RunItPOS ✓



Cloud POS focused on fashion retail with inventory features for multiple locations.

[Edit](#)

Last updated 37 minutes ago

Magento 2 ✓



The latest generation of Magento. Customizable for advanced merchants.

[Edit](#)

What now?

That's it. At this point you can sit back and expect to see your products in Magento 2 within minutes. Your catalog: products, categories, and inventory will be in sync with your POS. Any orders from the store will now flow directly into Sqquid which will post it into the POS you're integrated with.

Troubleshooting:

1. In case that products do not show up in your store within minutes:
 - a. Make sure that the user/pass is accurate, enabled, and with proper permissions
 - b. Make sure you have products in Sqquid
 - c. Make sure there are no errors (sqquid has it's own log file same folder as system.log)
2. Magento 2 is connected but I have no products in Sqquid:
 - a. You need a POS to enable this system to work properly. We currently 100% support RunIt POS and will soon offer additional systems.
 - b. Make sure you have the correct credentials for accessing your POS catalog
3. If all else fails, please contact us at hello@sqquid.com - please include the details of the issue and your catalog ID (can be found in the My account section on Sqquid).

Learn More

- Github Repo - <https://github.com/Sqquid/Magento2-CatalogSync>
- Packagist - <https://packagist.org/packages/sqquid/module-sync>
- Product Page - <https://sqquid.com/>