Memsource Connector for Magento 2

This article provides a step-by-step guide to setting up and using the Memsource Connector extension for Magento 2.
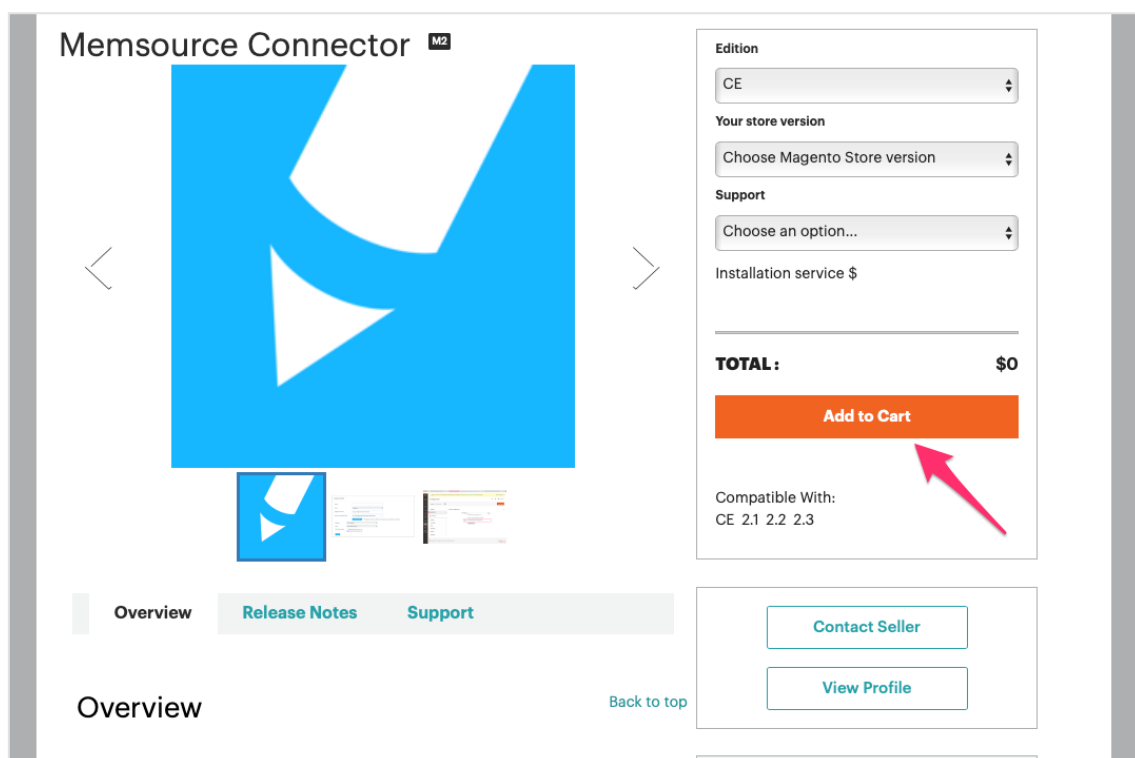
Requirements

Before you begin, make sure that you are using the [Enterprise edition](#) of Memsource. You will also need to be running Magento CE 2.1 or newer.
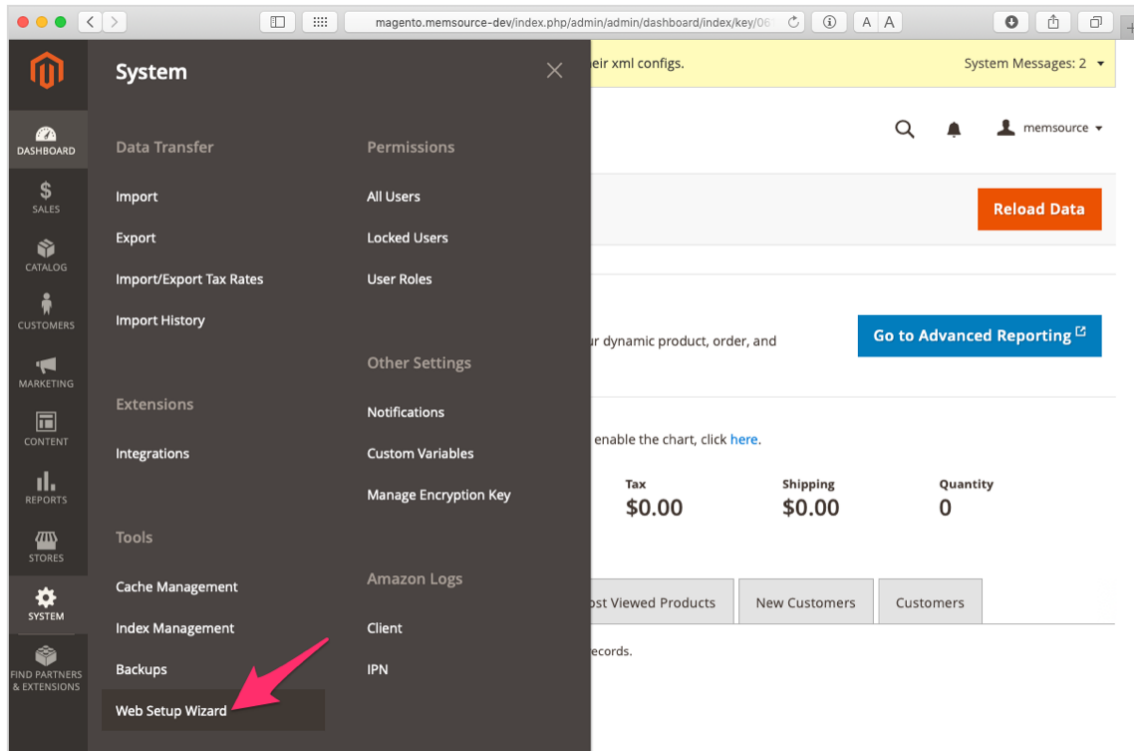
Installation

There are three ways to install the Memsource Connector extension into your Magento 2 instance: through the Magento Marketplace, by using Composer, or by installing it manually. Each of these three options will be explained below:
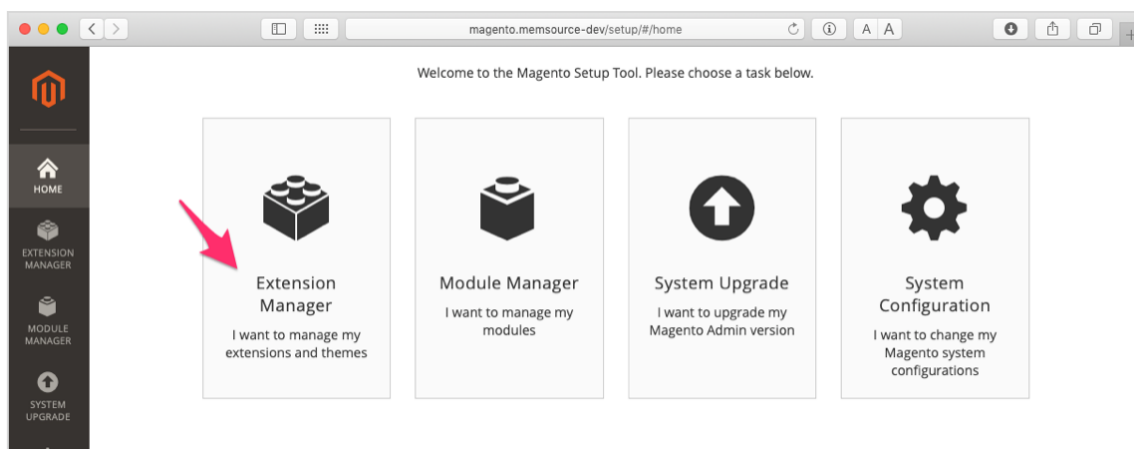
Magento Marketplace

Find the Memsource Connector on the [Magento Marketplace](#) and get the extension by clicking on *Add to Cart:*
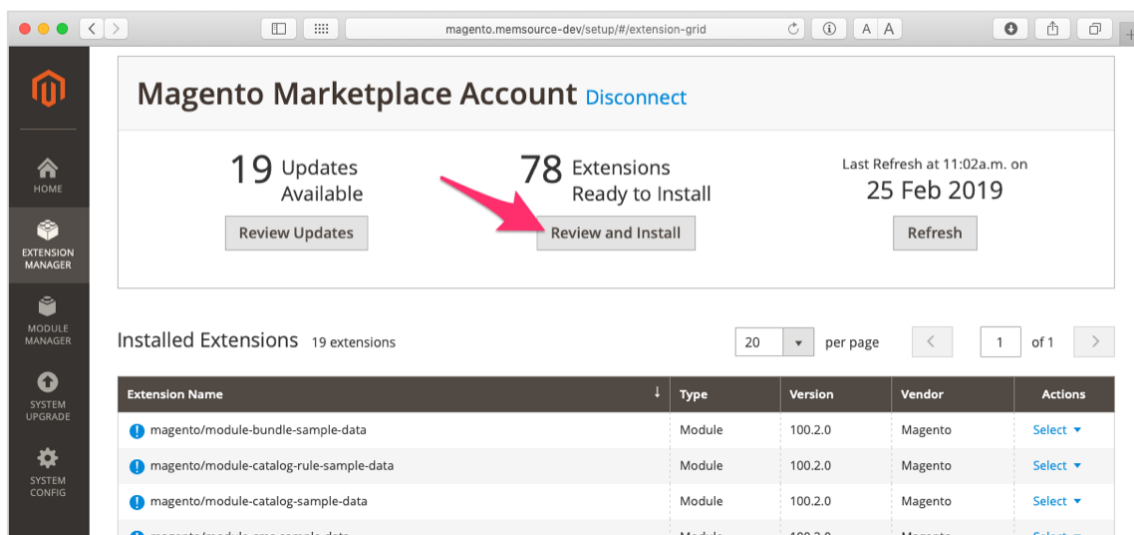
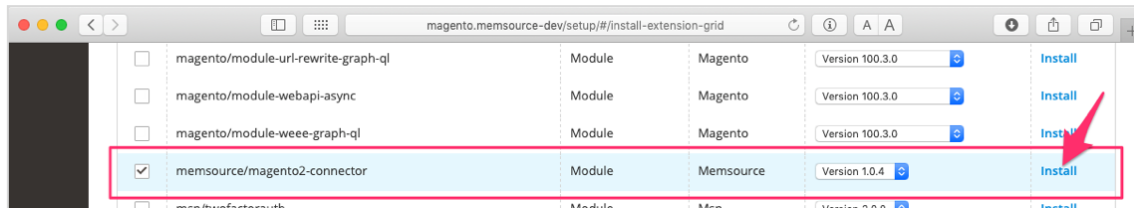In the Magento Administration, go to *System → Web Setup Wizard*:

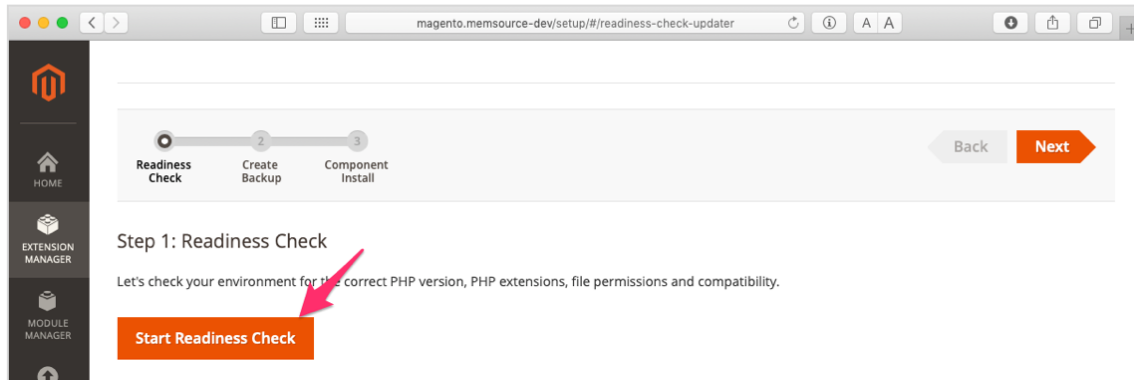Then, choose the *Extension Manager*:

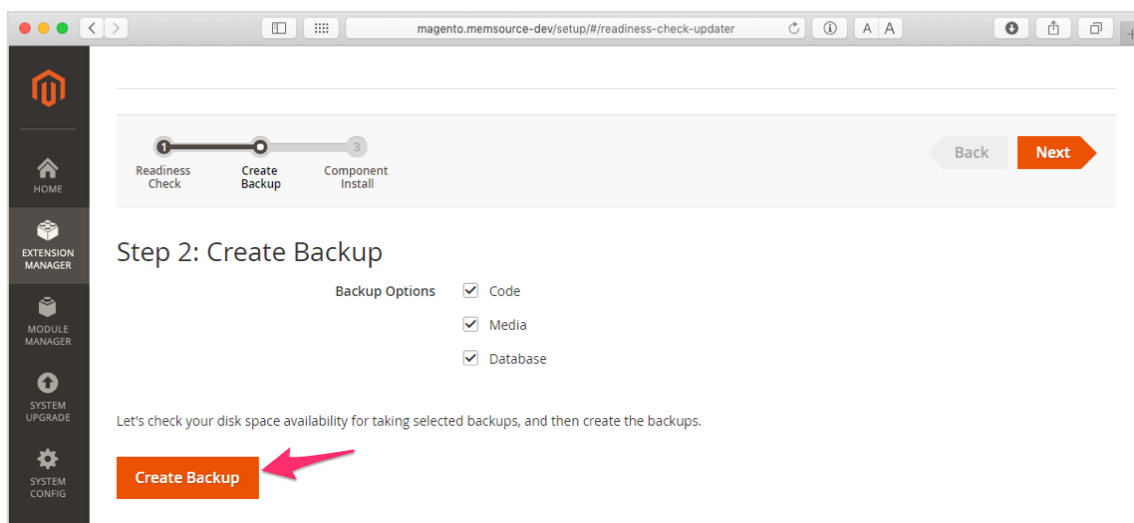Under the number of extensions to install, click on *Review and Install*:

On the list of extensions, find the Memsource Connector, and click on the Install link:
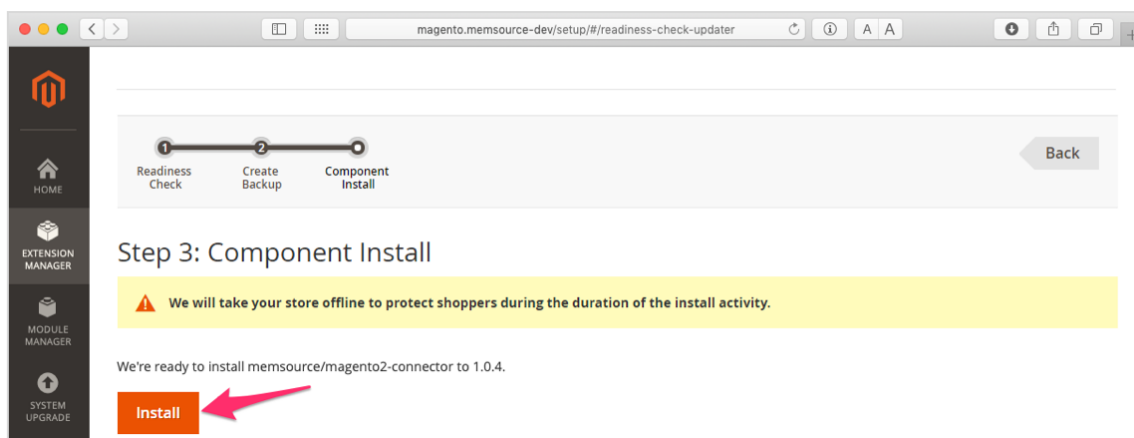

Now, we will go through the extension installation. Let's start with the readiness check:

Once you've passed the readiness check, you can continue the installation process by clicking the Next button. In the next step, you will be asked to create the data backup:

After the backup process is complete, you are ready to install the Memsource extension. Just click the Install button and the Memsource Connector will be installed:

If you are not sure about any of the steps above, check out the [Magento Docs](#) for further information.

Composer

Install the extension via Composer:

```
composer require memsource/magento2-connector
php bin/magento module:enable Memsource_Connector
php bin/magento setup:upgrade
php bin/magento setup:static-content:deploy
```

Manual Installation

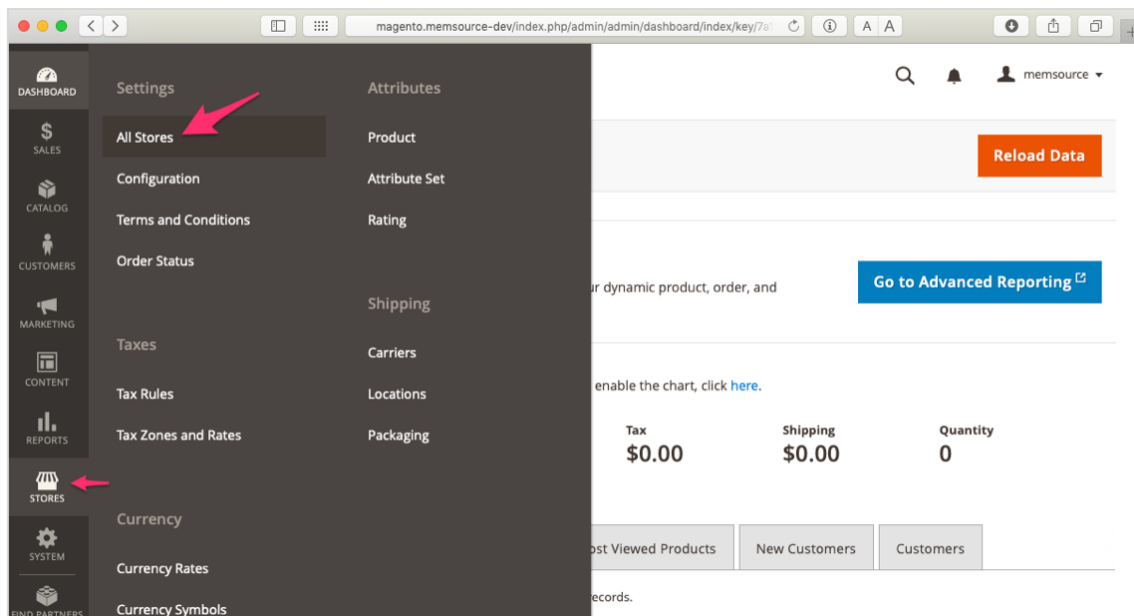If you are unable to use either Magento Marketplace or Composer:

1. Download the zip file from [GitHub releases](#)
2. Extract the downloaded zip file into the folder `app/code/Memsource/Connector`
3. Enable the extension and clear the cache:

```
php bin/magento module:enable Memsource_Connector
php bin/magento setup:upgrade
php bin/magento cache:flush
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
```
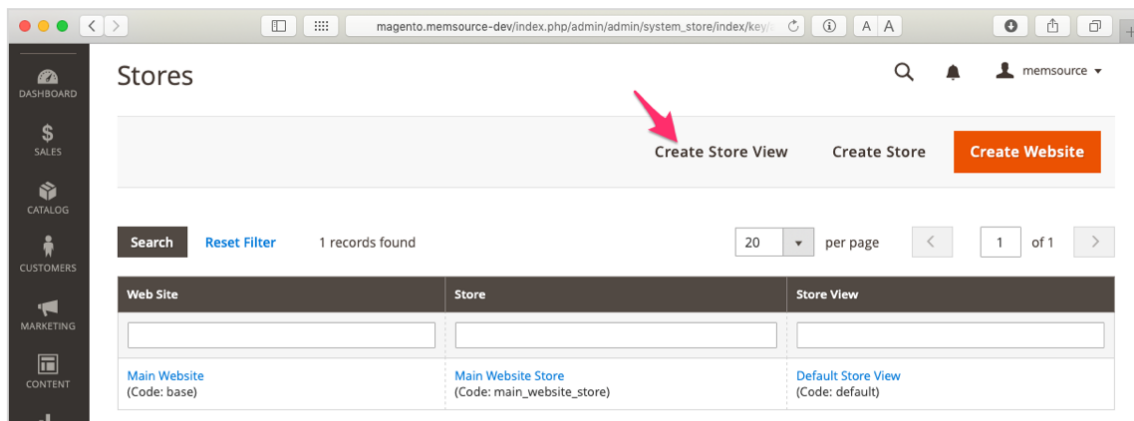
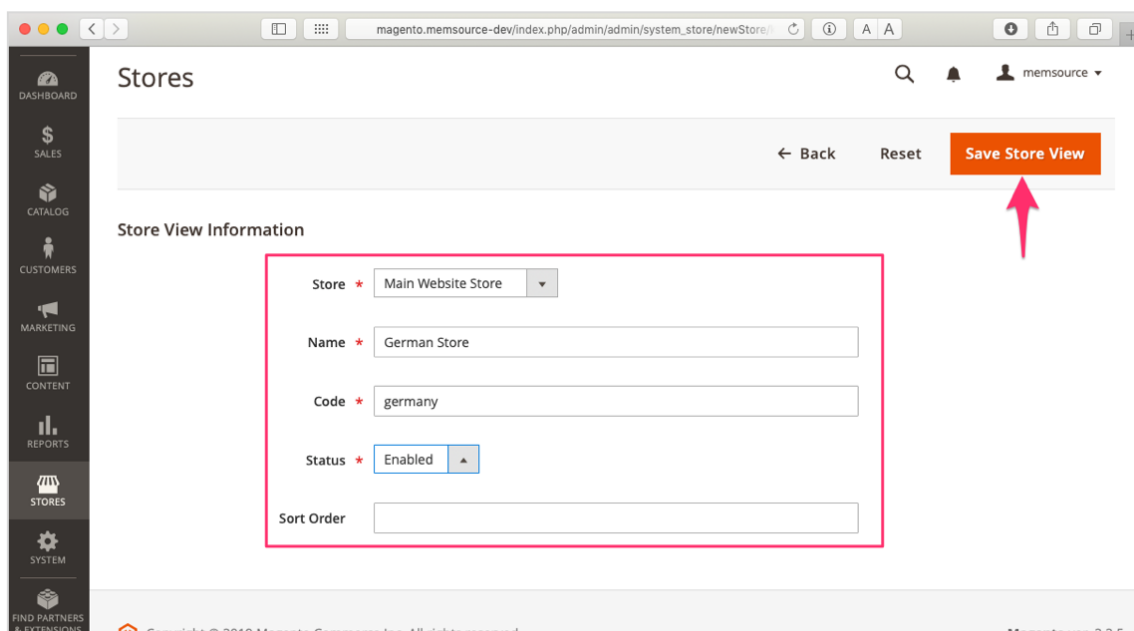# Configuration

Set up a Multilingual Website

Now it's time to connect Memsource with Magento. First, make sure that you have correctly set up your websites, stores and store views. Sign in to the Magento Administration and go to *Stores → All Stores*:
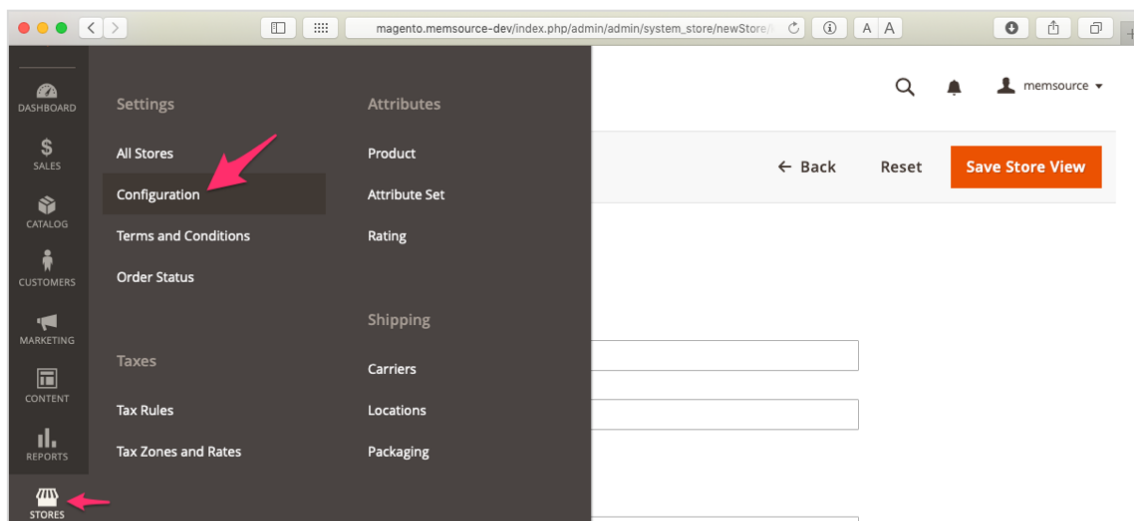
Every language is associated with a store view, so if you want to use two languages, you will need at least two store views. If you don't have enough store views, you can add more by clicking the *Create Store View* link:
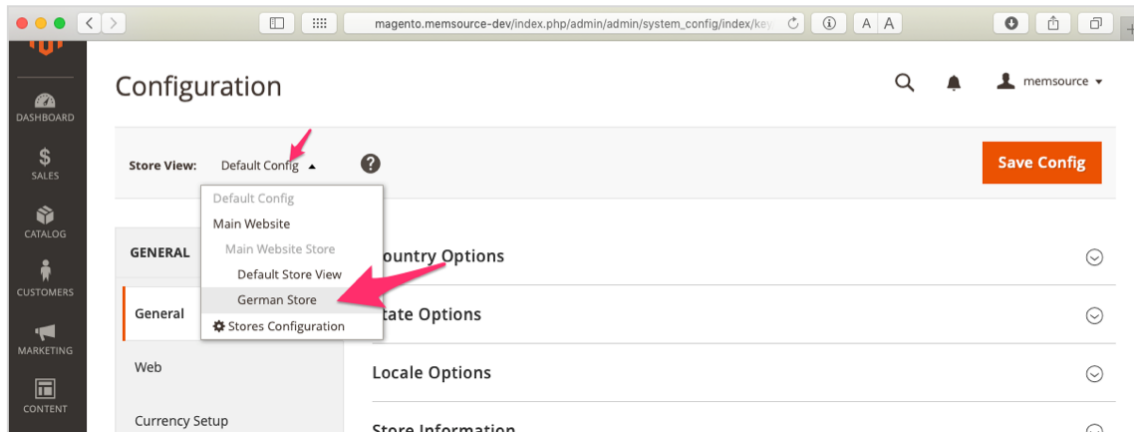

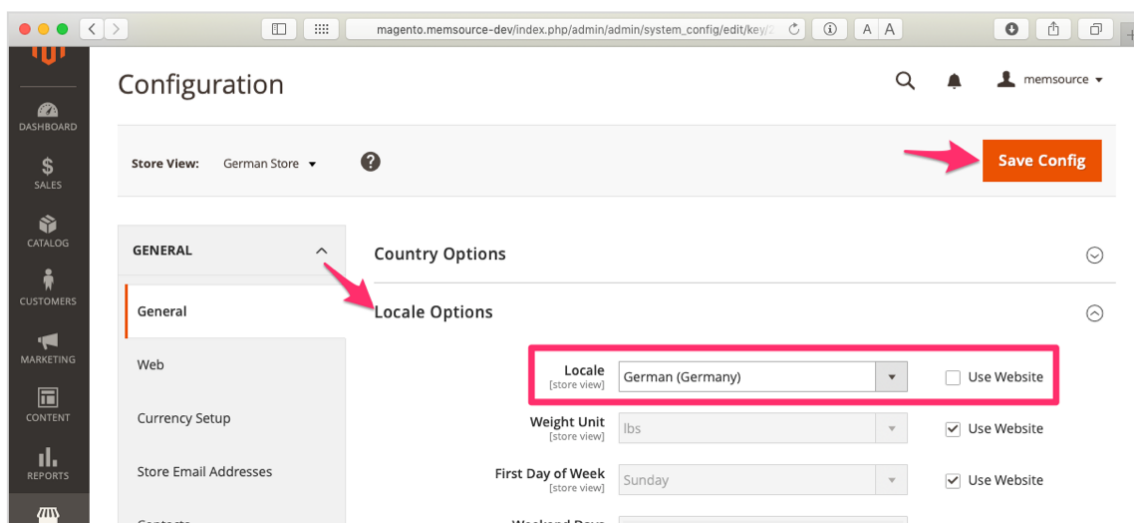Then, you have to provide some basic information about your new store view:

Every store view should have a language associated with it. To set them up, go to *Stores → Configuration*:
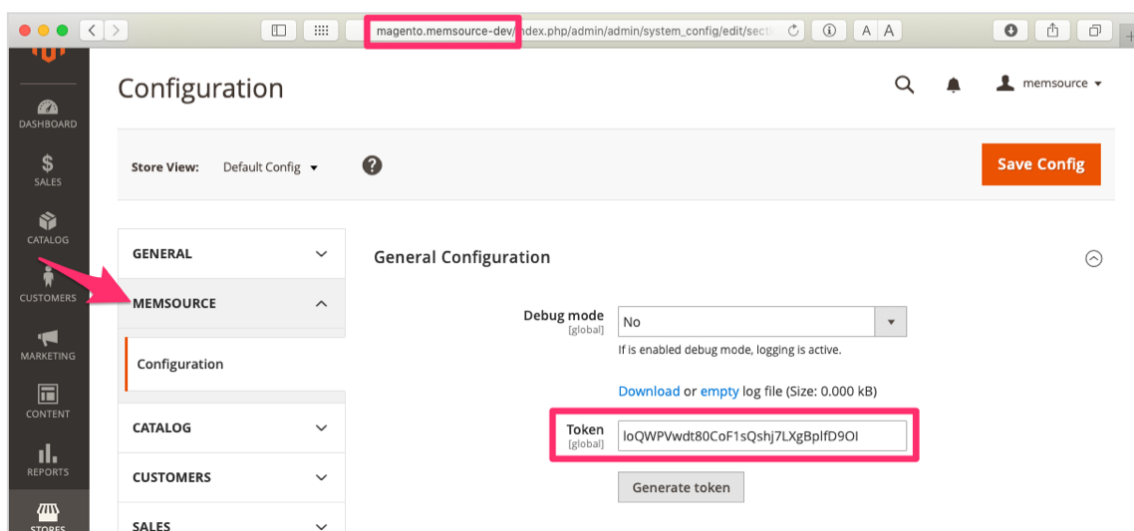
Select your store view from the dropdown menu:

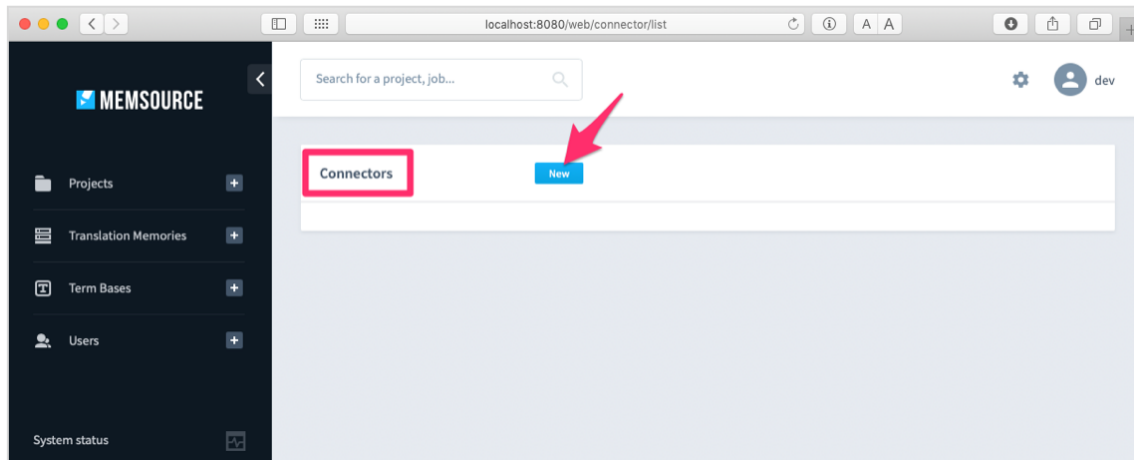Set the language for your store view. Update the Locale Options and click on *Save Config*:

Obtain the URL and token for API calls. Do this in Magento 2 by going to *Stores → Configuration → MEMSOURCE*.

On this page, you will see the URL and a token. The URL will be in the address bar, and the token can be found in the middle of the General Configuration section. Save these — you'll need these to configure Memsource.
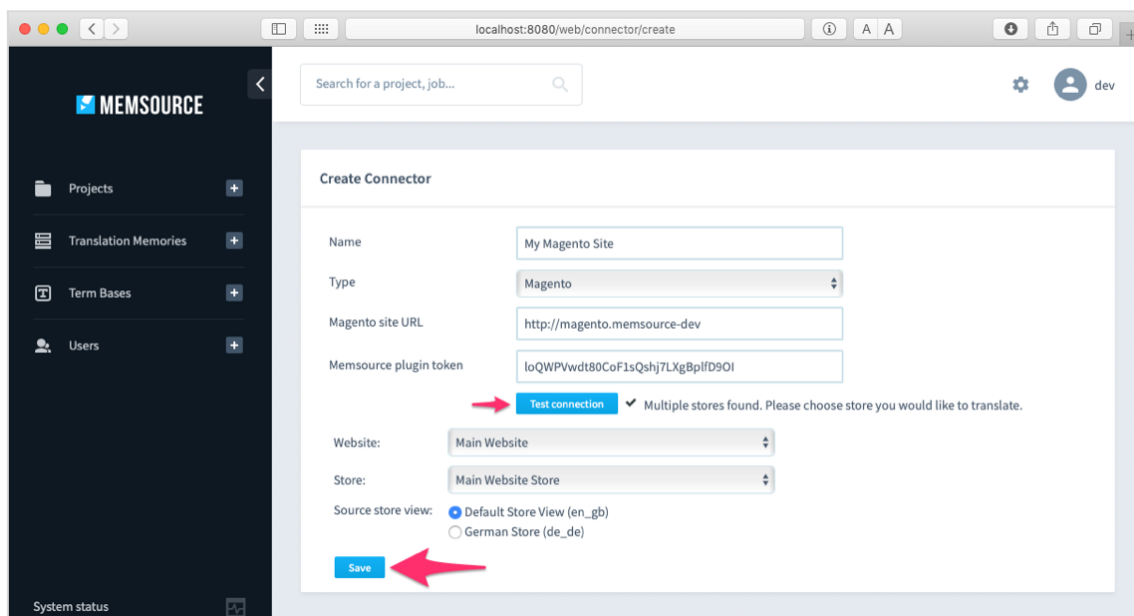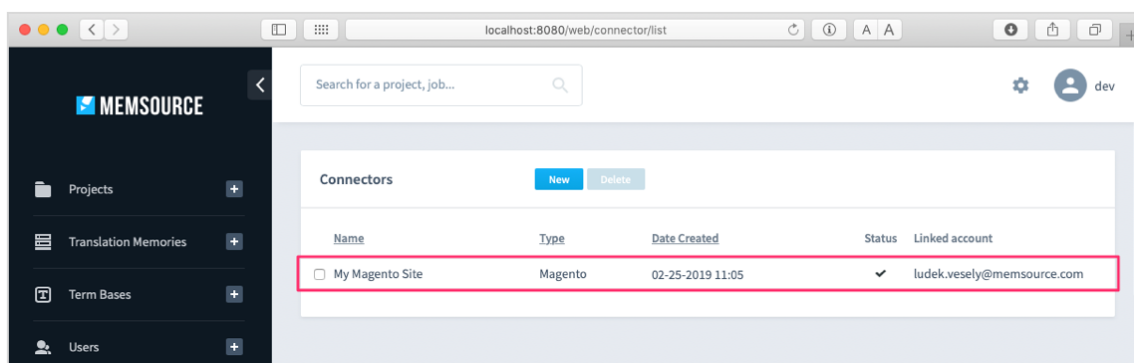
Memsource Configuration

Then, open Memsource and go to *Setup → Integrations → Connectors → New*:

Fill in the form with your URL and Token from the previous step and click on *Test connection*. Then, choose the source website, store, and store view:
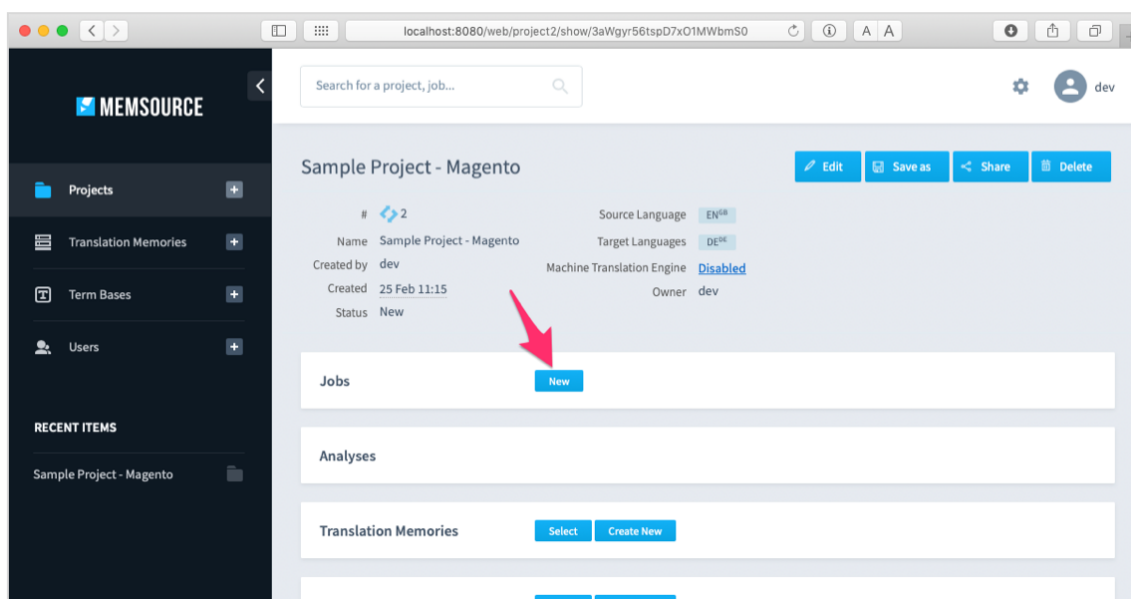

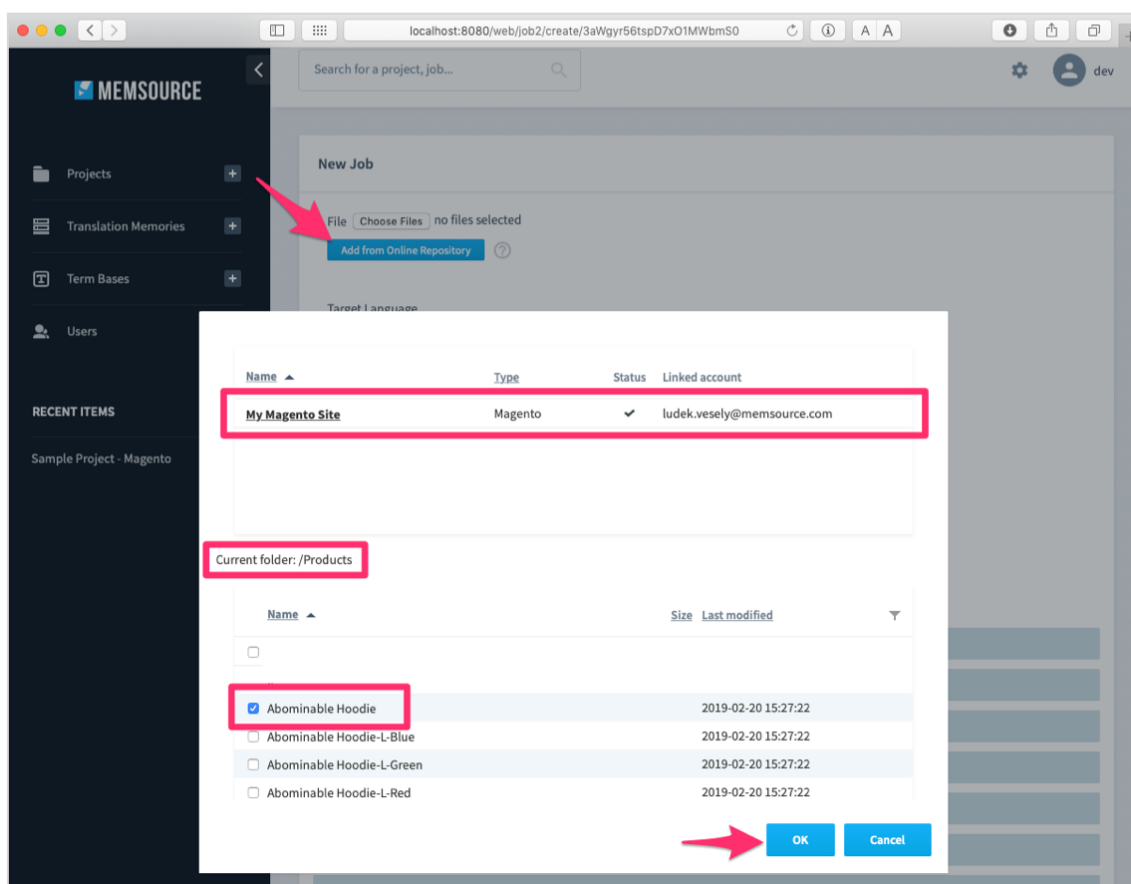If everything is successful, you should see your Magento account in the connectors list:

Usage

Now you can create a project (or choose an existing one) for content translation, and create a new job:

For translating Magento content, click on *Add from Online Repository* and choose your connector. Then, choose content you want to translate:

Finally, confirm the job creation:

After the translation, you can upload the translated content back to Magento:

In Magento Administration, go to the Products list and filter by the store view that was used as a target. You should then see your translated product: