

Zaius Magento 2 Connector

Integrate Zaius directly into your Magento instance using the Zaius Magento 2 Connector.

Getting Started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes. See deployment for notes on how to deploy the project on a live system.

Prerequisites

1. Magento 2 "^2.2.5"
2. PHP 5 >= 5.5.0, PHP 7
3. Composer

Installing

Composer

By far the quickest and easiest way to install and maintain the Zaius connector is to use Composer.

1. Require the Zaius Magento 2 package.
 2. Install the Zaius Magento 2 package.
 3. Finally, make sure the package is up-to-date.
- Add the required packages:

```
composer require zaius/zaius-magento-2:^1.0
composer install
composer update zaius/*
```

Alternative install: ZIP

- Download the Zaius Magento 2 module archive from Git:
<https://github.com/ZaiusInc/zaius-magento-2/archive/master.zip>
- Extract the contents of the ZIP file to
<MAGENTO_ROOT>/app/code/Zaius/Engage/<extract_here>.

The Zaius PHP SDK is required, and must be installed separately if you've chosen to install via ZIP archive.

- Add the required packages:

```
composer require zaius/zaius-php-sdk:^1.0
composer install
composer update zaius/*
```

Verify & Enable the Zaius Magento 2 Connector

- To verify that the extension installed properly, run the following command:

```
php bin/magento module:status
```

By default, the extension is probably disabled, and you will see output like this:

```
List of disabled modules:
Zaius_Engage
```

- Enable the extension and clear static view files:

```
php bin/magento module:enable Zaius_Engage --clear-static-content
```

You should see the following output:

```
The following modules have been enabled:
- Zaius_Engage

To make sure that the enabled modules are properly registered, run 'setup:upgrade'.
Cache cleared successfully.
Generated classes cleared successfully. Please run the 'setup:di:compile' command to generate static view files.
Generated static view files cleared successfully.
```

- Register the extension:

```
php bin/magento setup:upgrade
```

- Recompile your Magento project:

```
php bin/magento setup:di:compile
```

- Verify that the extension is enabled:

```
php bin/magento module:status
```

You should see output verifying that the extension is no longer disabled:

```
List of enabled modules:  
Zaius_Engage
```

```
List of disabled modules:  
None
```

- Clean the cache:

```
php bin/magento cache:clean
```

API Setup

In order to support the coupon code functionality Zaius utilizes, and to enable the Zaius support team to most effectively troubleshoot any issues, we require an API user be created with access to the Zaius APIs.

Create an appropriate role

In the Magento admin panel, navigate to System > User Roles (under the "Permissions" heading).

1. Click "Add New Role".
2. Enter the name "Zaius API".
3. Click "Role Resources".
4. Select the "Zaius Engage API" resource.
5. Click "Save Role".

Create the API user

Still in the Magento admin panel, navigate to System > All Users (under the "Permissions" heading).

1. Click "Add New User".
2. Enter the User Name "zaius_api"
3. Enter any details you like (Zaius recommends a maintainer's details) for name and email.
4. Create a new password for the user (and make sure to save this in a secure password manager or similar key store).
5. Click "User Role".
6. Select the "Zaius API" role you created in the previous section.
7. Click "Save User".

Save the API User in Zaius

Navigate to the Zaius integration ([Settings \(cogwheel\) > Integrations](#)) page, making sure you're editing the Zaius account this Magento store will be integrating with (Test or Prod, for the

appropriate brand if relevant).

1. Click the "Magento" card.
2. Select "Magento 2" from the Version dropdown.
3. Enter "zaius_api" as the Username.
4. Enter the password you configured in the previous section.
5. Enter the base Magento API URL, which usually matches your website's root URL.
6. Click Save.

Zaius does not recommend clicking "Start Bulk Import" at this time. Please consult with your Customer Success Manager to coordinate the bulk import process.

Configuration

After installing the module and setting up the API user, all configuration is done via Stores >> Configuration >> Zaius >> Engage.

Zaius Engage Status

Enabled: Enable or disable the Zaius Engage Connector functionality.

Version: The currently installed version of the Zaius Engage Connector.

Composer Installed?: Checks if Composer is installed. Will *ALWAYS* be installed for Magento 2 projects.

SDK Installed?: Checks if the [Zaius PHP SDK](#) is installed. The SDK is **REQUIRED** by the Zaius Engage Connector, and installed automatically with Composer.

Configuration

Zaius Tracker ID: Configuration field for the Zaius client Tracker ID. Found at [API Management](#) in the Zaius client Account.

Zaius Private API Key: Configuration field for the Zaius Private API Key. Found at [API Management](#) in the Zaius client Account. This is **REQUIRED** for Batch Updates to work.

Enable Amazon S3: Enable or disable the Amazon S3 Upload functionality.

Amazon S3 Key: Configuration field for the Zaius Client Amazon S3 Key. Found At [Integrations](#) in the Zaius Client Account. This is **REQUIRED** if Amazon S3 functionality is enabled.

Amazon S3 Secret: Configuration field for the Zaius Client Amazon S3 Secret. Found At [Integrations](#) in the Zaius Client Account. This is **REQUIRED** if Amazon S3 functionality is enabled.

Settings

Collect All Product Attributes: Enable or disable the functionality to collect all product attributes, or only the minimum. If your product feed has fields which are not yet captured in Zaius, you will likely need to turn this on and create a corresponding custom field. The Zaius support team can assist.

Track Orders on Frontend: Enable or disable the functionality to track orders on the frontend of the website. Disabled by default as backend order tracking is far more reliable.

Timeout: Specify a number of seconds to wait before timing out the connection to Zaius.

Zaius Localizations (Paid Beta)

Enabled?: Enable or disable the Zaius Localizations functionality. With Zaius Localizations functionality enabled, localized store_view data will be sent to Zaius. Please consult with your Zaius CSM before enabling.

Versioning

We use [SemVer](#) for versioning. For the versions available, see the [tags on this repository](#).

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details.