Sezzle Extension for Magento 2

Introduction

This document will help you in installing `Sezzle's Magento 2` extension. This extension is a certified one and listed [here](#) in the marketplace.

How to install the extension?

There are two ways of installing and upgrading the extension. * By composer. * Manual Process.

For all purposes assume [Magento] as your Magento 2 root directory.

Composer

- Open terminal and navigate to `Magento` root path.
- `composer require sezzle/sezzlepay`
- `php bin/magento setup:upgrade`
- `php bin/magento setup:di:compile`
- `php bin/magento setup:static-content:deploy`
- `php bin/magento cache:clean`

Manual

- Download the .zip or tar.gz file from `Sezzle's` github repository.
- Unzip the file and follow the following instructions.
- Navigate to `Magento` `[Magento]/app/code/` either through `SFTP` or `SSH`.
- Copy `Sezzle` directory from unzipped folder to `[Magento]/app/code/`.
- Open the terminal.
- Run the following command to enable `Sezzle`: `php bin/magento module:enable Sezzle_Sezzlepay`
- Run the `Magento` setup upgrade: `php bin/magento setup:upgrade`
- Run the `Magento` Dependencies Injection Compile: `php bin/`magento` setup:di:compile`
- Run the `Magento` Static Content deployment: `php bin/magento setup:static-content:deploy`
- Login to `Magento` Admin and navigate to `System > Cache Management`.
- Flush the cache storage by selecting `Flush Cache Storage`.

You can now directly navigate from the Configuration Page to get signed up for `Sezzle`. To do so, you need to click on `Register for Sezzle` which will redirect you to the `Sezzle Merchant Signup` Page. If you have the details already, you can simply click on `I've already setup Sezzle, I want to edit my settings` to move ahead.

How to upgrade the extension?

Composer

- Open terminal and navigate to `Magento` root path.
- `composer update sezzle/sezzlepay`
- `php bin/magento setup:upgrade`
- `php bin/magento setup:di:compile`
- `php bin/magento setup:static-content:deploy`
- `php bin/magento cache:clean`

Manual

- Download the .zip or tar.gz file from `Sezzle's` github repository.
- Unzip the file and follow the following instructions.
- Copy `Sezzle` directory from unzipped folder to `[Magento]/app/code/`. Make sure you are overwriting the files.
- Open the terminal.
- Run the following command to enable `Sezzle`: `php bin/magento module:enable Sezzle_Sezzlepay`
- Run the `Magento` setup upgrade: `php bin/magento setup:upgrade`
- Run the `Magento` Dependencies Injection Compile: `php bin/`magento` setup:di:compile`
- Run the `Magento` Static Content deployment: `php bin/magento setup:static-content:deploy`
- Login to `Magento` Admin and navigate to `System > Cache Management`.
- Flush the cache storage by selecting `Flush Cache Storage`.

Configure Sezzle

Payment Configuration

- Make sure you have the `Merchant UUID` and the `API Keys` from the [Sezzle Merchant Dashboard](#). Sign Up if you have not signed up to get the necessities.

- Navigate to `Stores > Configuration > Sales > Payment Methods > Sezzle > Payment Settings` in your `Magento` admin.
- Set the Payment Mode to `Live` for LIVE and set it as `Sandbox` for SANDBOX.
- Set the `Merchant UUID`, `Public Key` and `Private Key`.
- Set `Payment Action` as `Authorize` only for doing payment authorization only and `Authorize and Capture` for doing instant capture.
- Set the `Merchant Country` as per the origin.
- Set `Min Checkout Amount` to restrict Sezzle payment method below that amount.
- Set `Payment from Applicable Countries` to `Specific Countries`.
- Set `Payment from Specific Countries` to `United States` or `Canada` as Sezzle is currently available for US and Canada only.
- Set `Enable Customer Tokenization` to `Yes` for allowing Sezzle to tokenize the customer account if they approve it. If customer wish to tokenize their account, next time, they don't have to redirect to Sezzle Checkout for completing the purchase, rather it will happen in your website.
- Save the configuration and clear the cache.

In-Context Configuration

- Set `Enable In-Context Solution` to `Yes` for the In-Context Checkout to get activated.
- Set `In-Context Checkout Mode` to `IFrame` or `PopUp` depending on how you want Sezzle Checkout to get hosted.

Settlement Report Configuration

- Set `Enable Settlement Reports` to `Yes` for the Settlement Reports Dashboard to get activated.
- Set `Range` to a value based on which you want to fetch the Settlement Reports.
- Set `Enable Automatic Syncing` to fetch the Settlement Reports asynchronously.
- Set Schedule and Time of Day for the above automatic sync to run.

Note : *Automatic Syncing requires cron to be enabled.*

Widget Configuration

- Set `Enable Widget in PDP` to `Yes` for adding widget script in the Product Display Page which will help in enabling `Sezzle Widget` Modal in PDP.
- Set `Enable Widget in Cart Page` to `Yes` for adding widget script in the Cart Page which will help in enabling `Sezzle Widget` Modal in Cart Page.
- Set `Enable Installment Widget in Checkout Page` to `Yes` if you want to show the Sezzle Installment Widget under Sezzle Payment Option in Checkout Page.
- Set `Path to Price Element`. This is the path to the element in the Checkout Page where the order total text value will be detected.
- Save the configuration and clear the cache.

Developer Configuration

- Enable the log tracker to trace the `Sezzle` checkout process.
- Set `Send Logs to Sezzle` to `Yes` if you want the logs to be sent to Sezzle in a periodic basic. For this cron needs to be enabled.
- You can also download the latest logs by clicking on `Sezzle Log` if any.
- Save the configuration and clear the cache.

Your store is now ready to accept payments through Sezzle.

Frontend Functionality

- If you have correctly set up `Sezzle`, you will see `Sezzle` as a payment method in the checkout page.
- Select `Sezzle` and move forward.
- Once you click `Continue to Sezzle` or `Place Order`, you will be redirected to `Sezzle Checkout` to complete the checkout.
- In the final page of Sezzle Checkout, check the `Approve {Website Name} to process payments from your Sezzle account for future transactions.` You may revoke this authorization at any time in your `Sezzle Dashboard` to tokenize your account. And, then click on `Complete Order` to complete your purchase.
- If your account is already tokenized, order will be placed without redirection otherwise you will be redirected to Sezzle Checkout for completing the purchase.
- On successful order placement, you will be redirected to the order confirmation page.

Capture Payment

- If `Payment Action` is set to `Authorize and Capture`, capture will be performed instantly from the extension after order is created and validated in `Magento`.
- If `Payment Action` is set to `Authorize`, capture needs to be performed manually from the `Magento` admin. Follow the below steps to do so.
  - Go the order and click on `Invoice`.
  - Verify your input in the `Create Invoice` page and click on `Save` to create the invoice.
  - This will automatically capture the payment in `Sezzle`.
  - Payment can also be captured via Magento 2 Invoice API.

Refund Payment

- Go to `Sales > Orders` in the `Magento` admin.
- Select the order for which you want to refund the payment.
- Click on `Credit Memo` and verify your input in the `Create Credit Memo` page.
- Save it and the refunded will be initiated in `Sezzle`.
- In `Sezzle Merchant Dashboard`, `Order Status` as `Refunded` means payment has been fully refunded and `Order Status` as `Partially Refunded` means payment has been partially refunded.

- Payment can also be refunded via Magento 2 Refund API.

Release Payment

- Go to `Sales > Orders` in the `Magento` admin.
- Select the order for which you want to release the payment.
- Click on `Void` and confirm your action.
- In `Sezzle Merchant Dashboard`, `Order Status` as `Deleted due to checkout not being captured before expiration` means payment has been fully released.
- Only Full Release is supported from Magento.
- Payment can also be released via Magento 2 Void API.

Order Verification in Magento Admin

- Login to `Magento` admin and navigate to `Sales > Orders`.
- Proceed into the corresponding order.
- If Order Status is `Processing` and `Total Paid` is equals to `Grand Total`, payment is successfully captured by `Sezzle`.
- If Order Status is `Pending` and `Total Paid` is not equals to `Grand Total`, payment is authorized but yet not captured.
- If Order Status is `Closed`, payment is refunded.
- If Order Status is `Canceled`, payment is released.

Order Verification in Sezzle Merchant Dashboard

- Login to `Sezzle Merchant Dashboard` and navigate to `Orders`.
- Proceed into the corresponding order.
- Status as `Approved` means payment is successfully captured by `Sezzle`.
- Status as `Authorized, uncaptured` means payment is authorized but yet not captured.
- Status as `Refunded` means payment is refunded.
- Status as `Deleted due to checkout not being captured before expiration` means either payment was not captured in time or the payment is released.

Customer Tokenization Details

- Login to `Magento` admin and navigate to `Customers > All Customers`.
- Go inside a customer for which you want to see the tokenization details.
- `Sezzle` tab will appear if the customer is tokenized.
- `Customer UUID`, `Expiration Date` and `Status` will appear.

Settlement Reports

- Login to `Magento` admin and navigate to `Reports > Sales > Sezzle Settlement`.
- List of the latest Settlement Reports will be shown.
- To make a quick sync, enter the `From` and `To` Date and click on `Sync`.
- Click on `Download` from the `Action` column for downloading a Settlement Report.
- For viewing the details of a particular Settlement Report, click on `View` from `Action` column.
- Settlement Report details can also be downloaded by entering the Settlement Report view.
- Settlement Report can be downloaded via `CSV` or `Excel` and Settlement Report Details will be downloaded via `CSV`.

How Sandbox works?

- In the `Sezzle` configuration page of your `Magento` admin, enter the `Sandbox API Keys` from your `Sezzle Merchant Sandbox Dashboard` and set the `Payment Mode` to `Sandbox`, then save the configuration. Make sure you are doing this on your `dev/staging` website.
- On your website, add an item to the cart, then proceed to `Checkout` and select `Sezzle` as the payment method.
- To pay with Sezzle:
  - If customer is not tokenized, click `Continue to Sezzle`.
  - If customer is tokenized, click `Place Order`.
  - If In-Context checkout, click `Pay with Sezzle`.
- For In-Context checkout, the Sezzle checkout will be hosted in the configured mode, `iFrame` or `Popup`. Otherwise, you will be redirected to the Sezzle checkout.
- Sign In or Sign Up to continue.
- Enter the payment details using test data, then move to final page.
- Check the `Approve {Website Name} to process payments from your Sezzle account for future transactions. You may revoke this authorization at any time in your Sezzle Dashboard` to tokenize your account.
- If your account is already tokenized, order will be placed without redirection otherwise you will be redirected to Sezzle Checkout for completing the purchase.
- After payment is completed at Sezzle, you will be directed to your site's successful payment page.
- `Sandbox` testing is complete. You can login to your `Sezzle Merchant Sandbox Dashboard` to see the test order you just placed.

Troubleshooting/Debugging

- There is logging enabled by `Sezzle` for tracing the `Sezzle` actions.
- In case merchant is facing issues which is unknown to `Merchant Success` and `Support` team, they can ask for this logs and forward to the `Platform Integrations` team.

- Name of the log will be `sezzlepay.log` .It is always recommended to send the `system.log` and `exception.log` for better tracing of issues.

Docker Environment Set Up

Start

- Clone the repo.
- Execute `docker-compose up -d --build` to start the Magento server.
- Server will be up at `localhost:8085` . If you want to change that, edit the `docker-compose.yml` .

Install Magento

```
docker exec -it sezzle_magento2 process install
```

Sezzle will be installed alongside.

Sample Data Deploy

```
docker exec -it sezzle_magento2 process install-sampledata
```

Database Upgrade

```
docker exec -it sezzle_magento2 process upgrade
```

Compile

```
docker exec -it sezzle_magento2 process compile
```

Deploy Static Files

```
docker exec -it sezzle_magento2 process deploy
```

Set Developer Mode

```
docker exec -it sezzle_magento2 process developer
```

Cache Clear

```
docker exec -it sezzle_magento2 process clear
```

Cleanup Environment

```
docker-compose down --rmi local -v --remove-orphans
```