

# Openpay Installation and Configuration

Open Pay - Epi Server

Exported on 09/16/2021

## Table of Contents

<b>1</b>	<b>Notes.....</b>	<b>3</b>
<b>2</b>	<b>Installation .....</b>	<b>4</b>
2.1	Install Project/Package.....	4
2.2	QuickSilver Commerce Manager .....	6
2.3	QuickSilver Commerce Front-End Site .....	6
<b>3</b>	<b>Openpay Configuration .....</b>	<b>7</b>
3.1	Setting up the Openpay payment provider in the front-end site .....	7
3.1.1	To set up the Openpay payment provider, firstly browse the front-end site and follow these steps. ....	7
3.2	Setting up Openpay payment provider in Commerce-Manager .....	7
3.2.1	After deploying all necessary files, open the Optimizely Commerce Manager back-end site. Then, follow these steps.....	7

# 1 Notes

This guide show Openpay installation and configuration step in a standard QuickSilver site (Episerver commerce sample).

We provide a completed solution in [OpenpayAddonPaymentGateway](#)<sup>1</sup>.

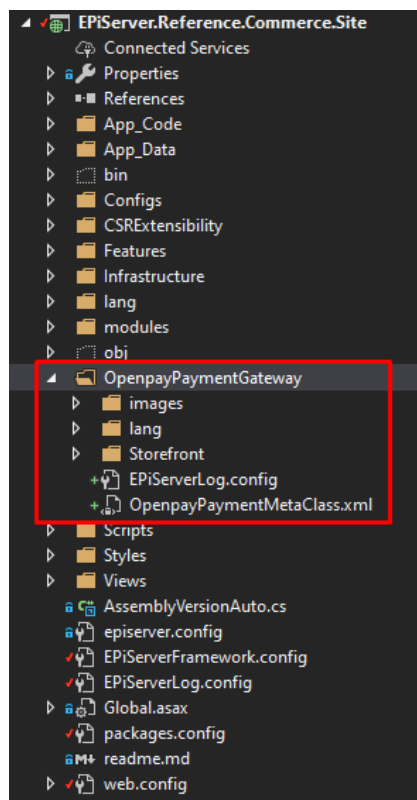
---

<sup>1</sup> <https://github.com/openpay-innovations/episerver-plugin>

## 2 Installation

### 2.1 Install Project/Package

1. Download & setup QuickSilver:
  - a. Follow this [installing-QuickSilver](#)<sup>2</sup>
2. In your QuickSilver solution:
  - a. Clone Openpay project at [OpenpayAddonPaymentGateway](#)<sup>3</sup>.
  - b. For another solution: Install package "Openpay.EpiCommerce.AddOns.PaymentGateway" in [nuget.episerver.feed](#)<sup>4</sup> to your "Commerce Site".
  - c. Related files will be created in **OpenpayPaymentGateway** folder.
  - d. Rebuild Solution.



3. The Openpay package contain these files:

File	Description
OpenpayPayment.cs	represents payment class for Openpay

<sup>2</sup> <https://world.episerver.com/documentation/installing-optimizely/installing-quicksilver/>

<sup>3</sup> <https://github.com/openpay-innovations/episerver-plugin>

<sup>4</sup> <http://nuget.optimizely.com/feed/packages.svc/>

File	Description
OpenpayPaymentMethod.cs	represents a payment method for Openpay
OpenpayPaymentGateway.cs	main payment gateway business
ConfigurePayment.ascx and its ascx.cs, ascx.designer.cs	Commerce Manager payment editor
OpenpayPaymentMetaClass.xml	contains payment meta class
OpenpayPaymentController.cs	page handler controller
Other controllers in Controller folder	widget controller
CommonConstants.cs	common global constants
OpenpayConfigurationConstants.cs	configuration constants
Files in BlockTypes folder	widget's block types
EnumHelper.cs OpenpayApiHelper.cs OpenpayConfiguration.cs Utilities.cs WidgetViewModelFactory.cs	helper classes
OpenpayPayment_EN.xml	language file
OpenpayConfigurationPage.cs	Exclude product configuration page type
OpenpayLandingPage.cs	Openpay landing page (widget)
OpenpayPaymentHandlerPage.cs	Openpay request handler page
GetOpenpayPurchaseLimits.cs MaintainOrderStatus.cs	scheduled jobs
_Openpay.cshtml	Openpay payment option in checkout page
_OpenpayConfirmation.cshtml	Openpay confirmation result when finish an order
Other *.cshtml files in Storefront folder	widget block view
EPiServerLog.config	Openpay custom logger
OpenpayPaymentMetaClass.xml	Meta fields

## 2.2 QuickSilver Commerce Manager

1. Find the Commerce Manager Apps folder, located under the Optimizely Commerce Manager project folder.
2. Deploy the **ConfigurePayment.ascx** file to the *Apps\Order\Payments\Plugins\Openpay* folder. If the **Openpay** folder does not exist, create it.
3. In file **EPiServerLog.config** copy & paste two node element `<appender>` & `<logger>` to **EPiServerLog.config** file in your website.
4. Deploy file **Openpay.EpiCommerce.AddOns.PaymentGateway.dll** to the bin folder of your Commerce Manager site.
5. In your solution's Commerce Manager project, add references to the deployed .dll and rebuild.

## 2.3 QuickSilver Commerce Front-End Site

1. Deploy files in folder **images** to your website's *styles\images* folder.
2. Deploy files \*.cshtml to your website's *Views\Shared* folder.
3. Deploy file **OpenpayPayment\_EN.xml** to lang folder in your website's *lang* folder.
4. In file **EPiServerLog.config** copy & paste two node element `<appender>` & `<logger>` to **EPiServerLog.config** file in your website.
5. Deploy file **Openpay.EpiCommerce.AddOns.PaymentGateway.dll** to the bin folder of your Commerce site.
6. In your solution's Commerce Manager project, add references to the deployed .dll and rebuild.

## 3 Openpay Configuration

### 3.1 Setting up the Openpay payment provider in the front-end site

#### 3.1.1 To set up the Openpay payment provider, firstly browse the front-end site and follow these steps.

Create Payment handler page:

1. Create a page using page type **Openpay Payment Handler Page** under your content tree.
2. Pick the destination for order complete page & order cancel page.

New Page: Openpay Payment Page

Start

Name

---

Required properties

Order Complete Page	<input type="text" value="Order confirmation"/> <input type="button" value="x"/> <input style="border: none; background: none;" type="button" value="..."/>
Order Cancel Page	<input type="text" value="Checkout"/> <input type="button" value="x"/> <input style="border: none; background: none;" type="button" value="..."/>

Create exclude product page:

1. Create a page using page type **Openpay Configuration Page** under your content tree.
2. Edit above page to pick Catalog, Package, Bundle or a Specified Product to the field **Product Catalog to be exclude**.
3. Open **Openpay payment provider** in Commerce Manager.
4. Open tab Parameters, **Excluded Product Type Config Item ID** - fill in the Content ID of the Openpay Configuration Page created above.

### 3.2 Setting up Openpay payment provider in Commerce-Manager

#### 3.2.1 After deploying all necessary files, open the Optimizely Commerce Manager back-end site. Then, follow these steps

1. Go to **Administration > Order System > Payments > English (United States)**. The last option is the language in which you want to make the Openpay payment available.

2. Click **New** to create a new payment method.
  - a. For **System keyword**, input "**Openpay**", the name of the folder created during deployment.
  - b. For **Class Name**, select **Openpay.EpiCommerce.AddOns.PaymentGateway.OpenpayPaymentGateway**.
  - c. For **Payment Class**, select **Openpay.EpiCommerce.AddOns.PaymentGateway.OpenpayPayment**.
  - d. For **IsActive** & **IsDefault** choose **Yes**.
  - e. Screen sample =>

The screenshot shows the 'Overview' tab of the Openpay configuration interface. The form contains the following fields and values:

- ID:** e2c9319a-34e1-4c38-9cd5-4906a1401c02
- Name:** Openpay
- Description:** Openpay
- System Keyword:** Openpay
- Language:** English
- Class Name:** Openpay.EpiCommerce.AddOns.PaymentGateway.OpenpayPaymentGateway
- Payment Class:** Openpay.EpiCommerce.AddOns.PaymentGateway.OpenpayPayment
- Sort Order:** 0
- IsActive:** Yes (selected)
- IsDefault:** No (selected)
- Supports Recurring:** No (selected)

Below the form, there is a section for **Restricted Shipping Methods** with two tabs: **Available Shipping Methods** and **Chosen Shipping Methods**.

3. Click OK to save the Openpay payment method.
4. Open the just created **Openpay** payment method for additional editing.
5. Go to the **Parameters** tab and enter the following:
  - a. **Region** - Select the Openpay's region to be used.
  - b. **Description** - Description.
  - c. **Environment** - Choose type of environment.
  - d. **Authorization Token** - Openpay authorize token provided by Openpay. (format in {username}|{password})
  - e. **Excluded Product Type Config Item ID** - Content ID of the Item created base on page type **OpenpayConfigurationPage**, which is used to define the product list needed to be exclude from Openpay payment.
  - f. **Call Back Item ID** - Content ID of the Item created base on page type **Openpay Payment Handler Page**.
  - g. **Min Purchase Limit** - The minimum purchase price allowed.
  - h. **Max Purchase Limit** - The maximum purchase price allowed.
  - i. **Run Purchase Limits API** - Button to manually get the min-max purchase limits from Openpay.
  - j. Screen sample =>



The screenshot shows the 'Configure Openpay' dialog box with the 'Parameters' tab selected. The fields are as follows:

- Region : Australia (dropdown)
- Description : (text input)
- Environment : ☐ Sandbox, ☐ Production
- Openpay Username : (text input)
- Openpay Password : (text input)
- Excluded Product Type Config Item ID : (text input)
- Call Back Item ID : (text input)
- Minimum Purchase Limit : 0 (text input)
- Maximum Purchase Limit : 0 (text input)
- Get Min/Max Values : Run Min/Max (button)

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

6. Open the **Markets** tab and add the expected markets for this payment → Click **OK**
7. In Commerce Manager, go to **Administration > Order System > Meta Classes**.
8. Click **Import/Export**, select **Import MetaData**.
9. To populate the MetaData import screen, drag and drop the meta class file to upload, **OpenpayPaymentMetaClass.xml**.
10. Select the OpenpayPaymentMetaClass.xml in the MetaData import screen. Click **Start Import**.