Pointspay

WooCommerce Plugin Installation Guide

Version 1.3

1 Introduction

Pointspay is an Alternative Payment Method (APM) that lets shoppers pay with their favourite loyalty program miles or points at participating web shops.

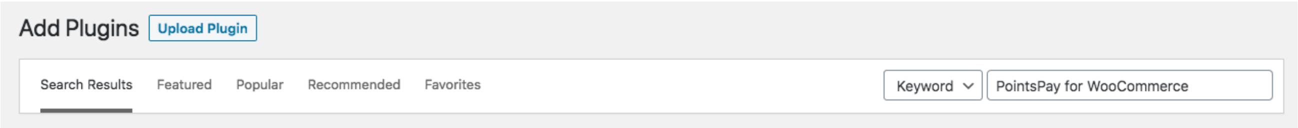
The Pointspay plugin for WooCommerce enables merchants who are on WooCommerce to easily install Pointspay as an additional payment method at their checkout and process payments and refunds seamlessly.

With the Pointspay plugin merchants have the ability to enable one available Pointspay payment method variants, like Flying Blue+ or Etihad GuestPay. **At the moment multiple variants simultaneously are not supported.**

2 Installation

The Pointspay Woocommerce plugin is compatible with WordPress Version: 4.0 and WooCommerce Version**:** 3.7

Automatic installation

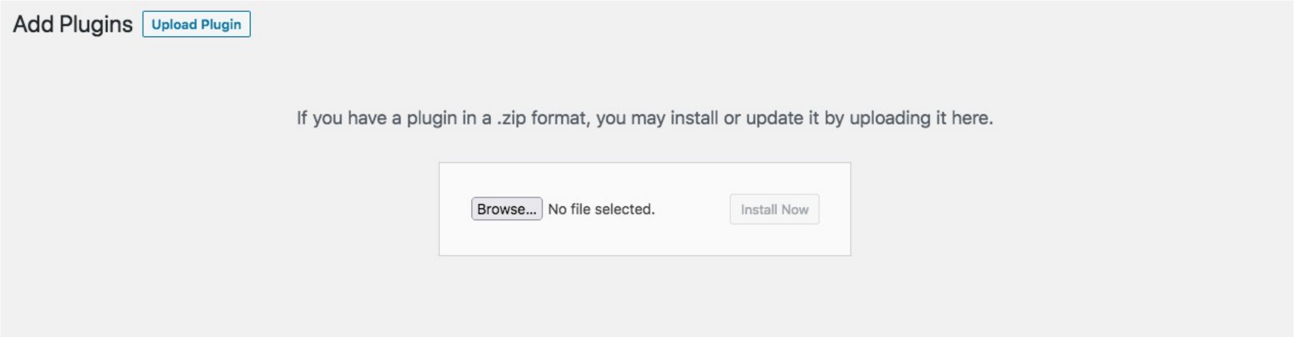


PointsPay for WooCommerce

Automatic installation is the easiest option as WordPress handles the file transfers itself and you don’t need to leave your web browser. To do an automatic install of our plugin, log in to your WordPress dashboard, navigate to the **Plugins** menu and click **Add New**. You will see Add Plugins screen.

In the search field type “PointsPay for WooCommerce”. Once you’ve found our plugin you can view details about it such as the version, rating and description. Most importantly of course, you can install it by simply clicking **Install Now**.

Manual installation



If you have a plugin in a .zip format, you may install or update it by uploading it here.

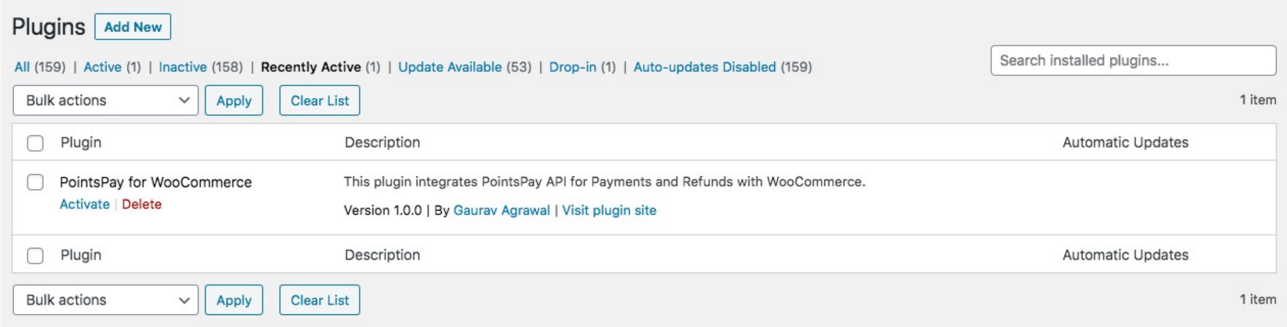
No file selected.

For installing the plugin manually, you have two options:

1. Extract the plugin zip file and drop its contents in the /wp-content/plugins/ directory of your WordPress installation.

2. Navigate to the **Plugins** menu and click **Add New**. Click on **Upload Plugin** button on *Add Plugins* screen. Now, either **Browse** the plugin zip or drag-and-drop it in the file upload container box. Then click on **Install Now**.

Plugin activation



Plugins Add New

All (159) | Active (1) | Inactive (158) | Recently Active (1) | Update Available (53) | Drop-in (1) | Auto-updates Disabled (159)

Search installed plugins...

Bulk actions | Apply | Clear List — 1 item

□ Plugin	Description	Automatic Updates
□ PointsPay for WooCommerce Activate | Delete	This plugin integrates PointsPay API for Payments and Refunds with WooCommerce. Version 1.0.0 | By Gaurav Agrawal | Visit plugin site	
□ Plugin	Description	Automatic Updates

Bulk actions | Apply | Clear List — 1 item

Once the plugin is installed, you can activate the plugin by clicking on **Activate** link below the plugin name, in *Plugins* listing.

3 Plugin Configurations

After activating the plugin, **Configure** link will appear below the plugin name in *Plugins* listing. You can click on this link to reach the plugin configuration settings. Alternatively, you can go to **WooCommerce** menu, and click on **Settings**; then go to **Payments** tab, and click on **Manage** button next to PointsPay method to reach this page.

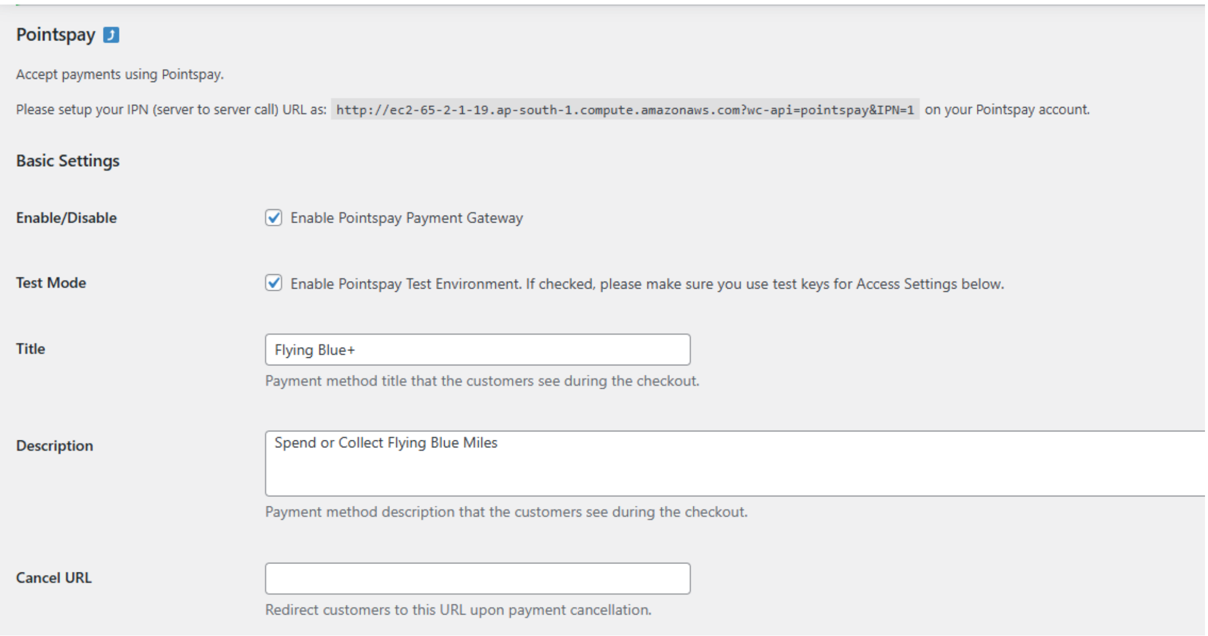
NOTE – Please convey the IPN URL listed on settings page to PointsPay team, as they would need to configure it at backend for your PointsPay account.

Please note that only one version of the available variants of Pointspay can be used at the same time. The available variants are:

- Flying Blue+
- Etihad GuestPay
- PointsPay

Basic settings

Please note the below example relates to configuring Flying Blue+



Pointspay

Accept payments using Pointspay.

Please setup your IPN (server to server call) URL as: `http://ec2-65-2-1-19.ap-south-1.compute.amazonaws.com?wc-api=pointspay&IPN=1` on your Pointspay account.

Basic Settings

Enable/Disable	☑ Enable Pointspay Payment Gateway
Test Mode	☑ Enable Pointspay Test Environment. If checked, please make sure you use test keys for Access Settings below.
Title	Flying Blue+ Payment method title that the customers see during the checkout.
Description	Spend or Collect Flying Blue Miles Payment method description that the customers see during the checkout.
Cancel URL	 Redirect customers to this URL upon payment cancellation.

Basic Settings relate to usability and display configuration of the payment gateway. The options under Basic Settings are as below:

1. **Enable / Disable –** Use this option to enable or disable PointsPay Payment Gateway on your WooCommerce store.
2. **Test Mode** – Enable this option if you want to test the payment gateway without charging any amount to the customers. Remember that orders will still be placed, so use this method only on your test/staging site, and not on live/production site. If you enable Test Mode, remember to use test credentials in *Access Settings* fields.
3. **Title** – This is the payment method title that customers see during the checkout. You can change it to a different relevant text such as "Flying Blue+".
4. **Description –** This is the description for payment method that customers see during the checkout. You can change it to a different relevant text "Spend or collect Flying Blue miles".
5. **Cancel URL** – This is an optional field in which you can specify the URL to redirect the user to if they cancel the payment on PointsPay. By default, the user is redirected to order payment page in case of payment cancellation.

To configure Etihad GuestPay instead of Flying Blue+ please use the following parameters instead:

1. **Title** – This is the payment method title that customers see during the checkout. You can change it to a different relevant text such as "Etihad GuestPay".

2. **Description –** This is the description for payment method that customers see during the checkout. You can change it to a different relevant text "Spend or collect Etihad Guest points".

Access settings

Access Settings pertain to the access and security configuration of the plugin.
The options under Access Settings are as below:

1. **API Username** – API Username is generated at the time of activation of your account and helps to uniquely identify your account to PointsPay.
2. **API Password** – API Password is generated at the time of activation of your account and helps to validate your API Username.
3. **API Access Token** – API Access Token is like a secret key that helps to validate transaction status notifications from PointsPay to your store.
4. **Merchant Code (Shop ID)** – Merchant Code is the unique identity of this WooCommerce store, as associated with your account with PointsPay. You will have one Merchant Code for each WooCommerce store associated with your account on PointsPay.
5. **Private Key** – Private Key is used for signing all transaction requests sent from your WooCommerce store to PointsPay. A method for generating a set of Private and Public keys is described next.

NOTE – Click on **Save changes** button after making any changes to PointsPay settings, for changes to take effect.

Generating Private and Public Keys

In order to securely transact with PointsPay, all transaction requests are required to be digitally signed. For this purpose, you need to generate a private-public key pair on your server, and obtain a certificate corresponding to public key. We recommend to use an 8192-bit key to accommodate the payload size. Certificate can be either self-signed or signed by Certificate Authority (CA).
You can generate a self-signed certificate using below *openssl* command:

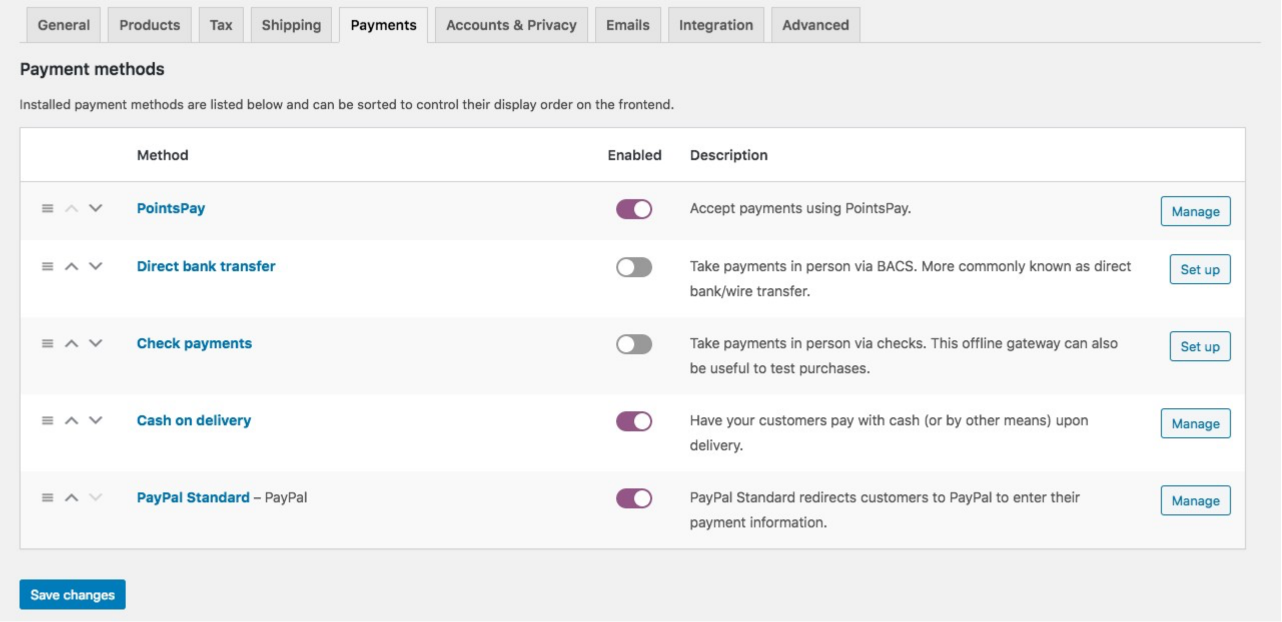
```
openssl  req  -newkey  rsa:8192  -nodes  -keyout  key.pem  -x509  -days
1095  -out
```

This plugin assumes that the generated key has no password and SHA256 encryption has been used. The above command complies to both the assumptions.

Once you've generated the keypair as above, submit *certificctee.cer* file to PointsPay to associate with your merchant code, whereas copy the contents of *keye.pem* file in **Private Key** input box under *Access Settings*.

Position among Payment Methods on Checkout Page

You can change the order of payment methods on Checkout Page by changing their order on
Payments tab under **WooCommerce > Settings**.



Payment methods

Installed payment methods are listed below and can be sorted to control their display order on the frontend.

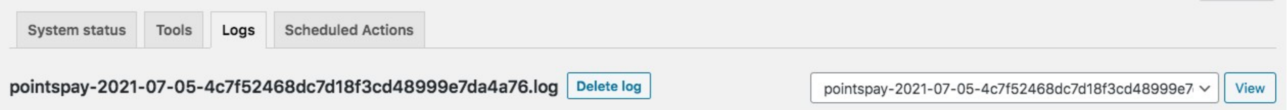
Method	Enabled	Description	
PointsPay	●	Accept payments using PointsPay.	Manage
Direct bank transfer	○	Take payments in person via BACS. More commonly known as direct bank/wire transfer.	Set up
Check payments	○	Take payments in person via checks. This offline gateway can also be useful to test purchases.	Set up
Cash on delivery	●	Have your customers pay with cash (or by other means) upon delivery.	Manage
PayPal Standard – PayPal	●	PayPal Standard redirects customers to PayPal to enter their payment information.	Manage

You can click *Up* and *Down* arrows to move the payment methods above or below another payment method. Alternatively, you can drag and drop the payment method at any position after grabbing it bythe list icon on the left.

Once you've arranged the payment methods in desired order, click on **Save changes** button for thenew order to reflect on your store's *Checkout* page.

Accessing log files

In case you are facing any issues with the plugin, you would want to contact our support team for help. In such cases, you should submit log files along with your support request. Log files are generated for the plugin on daily basis, and can be accessed by going to **WooCommerce** menu and clicking on **Status** link, and then going to **Logs** tab.



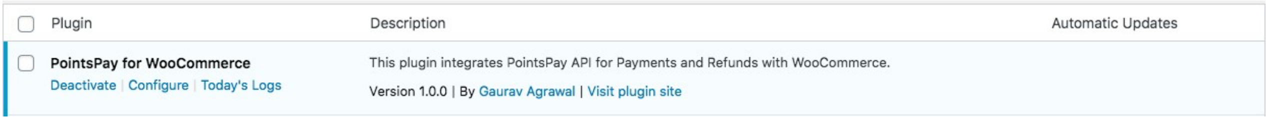
pointspay-2021-07-05-4c7f52468dc7d18f3cd48999e7da4a76.log

pointspay-2021-07-05-4c7f52468dc7d18f3cd48999e7

Select the log file for a particular day for our plugin by selecting it from the dropdown menu on the right, and click on **View** button. Names for all the log files for our plugin will begin with ‘pointspay-’. Log file for a specific day will have the date in YYYY-MM-DD format in the log file name.

You can also access today’s log file by clicking on **Today’s Logs** link under the plugin name in *plugins*

listing.



□ Plugin	Description	Automatic Updates
□ PointsPay for WooCommerce	This plugin integrates PointsPay API for Payments and Refunds with WooCommerce. Version 1.0.0 | By Gaurav Agrawal | Visit plugin site	

Multi – currency plugin compatibility

This plugin relies on WooCommerce’s get_currency() method. As most multi-currency plugins override this method to setup the transaction currency, this plugin should be compatible with most such plugins out of the box. Our customers have found this plugin to be compatible with the following multi-currency plugins:

- CURCY – WooCommerce Multi Currency *by* VillaTheme
- WOOCS – Currency Switcher for WooCommerce *by* realmag777