



## Gateway Integration Guide

Version 2.7

## Table of Contents

<b>1.Introduction</b>	<b>3</b>
1.1 Purpose	3
1.2 Scope	3
<b>2. Integration</b>	<b>3</b>
2.1 Payment Workflow	4
2.2 Request Parameters	4
2.3 Response Parameters	5
2.4 Gateway Status Codes	5
2.4.1 Response Status Codes	5
2.4.2 Error Codes	5
2.5 Transaction Request	6
2.5.1 Gateway ID's	7
2.5.2 EMI Gateway ID's	7
2.6 Transaction Response	9
<b>3 Sample Code (PHP)</b>	<b>10</b>
<b>4 WEBXPAY Plugins &amp; SDK</b>	<b>12</b>
4.1 Opencart Integration	12
4.2 WooCommerce Integration	12
4.3 Shopify Integration	13
4.4 Prestashop Integration	15
4.5 Magento Integration	16
4.6 WHMCS Integration	17
5 Test Cards	17

## 1. Introduction

WEBXPAY Gateway offers a highly secure, mobile-responsive, and user-friendly payment solution. It supports transactions from any gateway integrated with a merchant's website, accommodating all international card brands and various local payment methods such as eZ Cash, mCash, Frimi, Upay, Genie, and DFCC Wallet. To enhance security, alerts are implemented to detect any fraudulent activities and manage chargeback processes.

This integration guide provides developers with a comprehensive understanding of the integration process and the necessary software requirements. It includes sample code snippets to assist in seamlessly integrating the gateway functionalities with the merchant's website.

### 1.1 Purpose

The purpose of this document is to guide developers through the integration of the WEBXPAY Gateway with merchant websites. It aims to provide clear, concise instructions and sample code snippets to ensure a smooth and efficient integration process. By following this guide, developers will be able to implement a secure, mobile-responsive, and user-friendly payment gateway that supports a wide range of international and local payment methods, enhancing the overall transaction experience for both merchants and customers.

### 1.2 Scope

This document encompasses the necessary steps and requirements for integrating the WEBXPAY Gateway with merchant websites. It covers the following aspects:

1. Integration Overview: An introduction to the WEBXPAY Gateway and its key features.
2. Supported Payment Methods: Details on the international card brands and local payment methods accepted by the gateway.
3. Security Measures: Information on security alerts and fraud detection mechanisms.
4. Technical Requirements: A comprehensive list of software and hardware prerequisites for successful integration.
5. Integration Process: Step-by-step instructions for integrating the gateway with a merchant's website, including sample code snippets.
6. Testing and Validation: Guidelines for testing the integration to ensure proper functionality.
7. Troubleshooting: Common issues and solutions to assist developers in resolving potential problems.

By adhering to this guide, developers will ensure a seamless integration of the WEBXPAY Gateway, providing a secure and efficient payment experience for users.

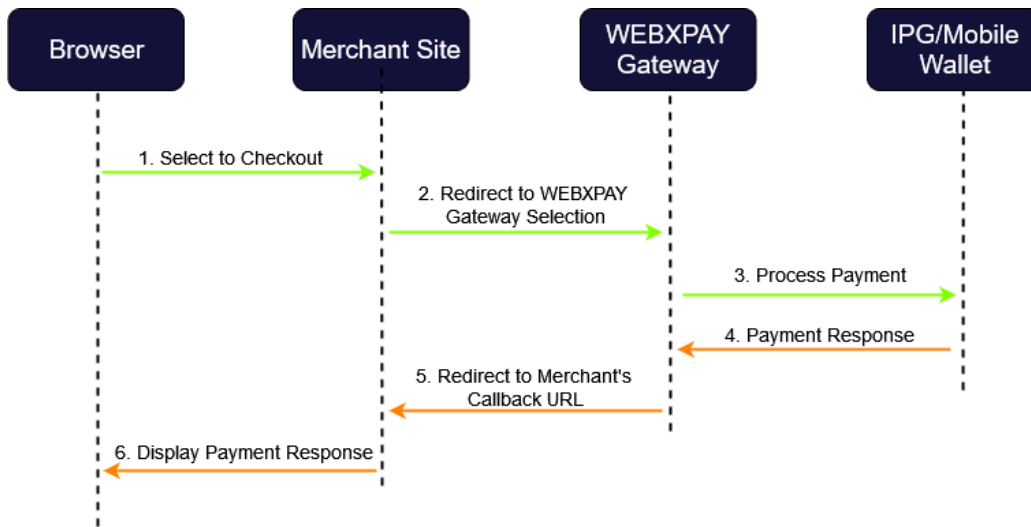
## 2. Integration

Customer and order data are transmitted to the following endpoints for the customers to proceed with payments in the merchant's website.

Interaction Point	URL
Payment Request URL (LIVE)	<a href="https://webxpay.com/index.php?route=checkout/billing">https://webxpay.com/index.php?route=checkout/billing</a>
Payment Request URL (STAGING)	<a href="https://stagingxpay.info/index.php?route=checkout/billing">https://stagingxpay.info/index.php?route=checkout/billing</a>

## 2.1 Payment Workflow

The sequence of events associated with WEBXPAY Gateway initializing from customer to the point of checkout is depicted below.



## 2.2 Request Parameters

The request has to be a HTTP POST request, and parameters required by WEBXPAY Gateway are as follows.

Parameter Name	Description	Mandatory	Max Character Count
first_name	Customer's first name	Yes	30
last_name	Customer's last name	Yes	30
email	Customer's email address	Yes	Valid email format
contact_number	Customer's contact number	Yes	Max 20, Min 9 Characters (Numbers and plus (+) sign only)
address_line_one	Customer's address	Yes	
address_line_two	Customer's address	No	
city	Customer's city	No	
state	Customer's state	No	
postal_code	Customer's postal code	No	
country	Customer's country	No	
secret_key	Provided upon registration. The key can be retrieved from the merchant backend under Settings\Integrations menu.	Yes	
payment	Refer section 3.5	Yes	
cms	Magento, PrestaShop, WooCommerce, OpenCart, PHP, etc.	Yes	
process_currency	LKR or USD	Yes	
custom_fields	Refer section 3.5	No	
payment_gateway_id	Refer section 3.5.1	No	

## 2.3 Response Parameters

Parameter Name	Description
Payment	Refer Section 2.5 Transaction Request
Signature	The checksum of payment parameter to validate the response is not tampered.
Custom Feilds	Variables passed to gateway side rather than main variables

The customer is redirected to the merchant's website with the following parameters when the payments is being processed.

## 2.4 Gateway Status Codes

### 2.4.1 Response Status Codes

Response status codes are used to track down the transaction statuses. The status codes depend on the gateways chosen by the merchant.

Code	Status	Description
0	Transaction Approved	The Transaction was Successful
15	Transaction Declined	The Transaction was Unsuccessful

### 2.4.2 Error Codes

The following is some of the most common error codes displayed on merchant websites when the required information result in failure.

Error Code	Description
401	Invalid access
402	Can't identify product
403	Invalid Secret Key
405	First name is required
406	Last name is required
407	Email Address is required
408	Contact Number is required
409	Total amount less than 1 USD / 1 LKR
410	LKR total amount exceed
411	USD total amount exceed
412	Not supported currency code
413	Can't find the Gateways
414	Selected gateway is not found
415	Error gateway ID
416	Bank Response not received
417	Currency code not defined
418	Return URL missing X Gateway
419	Transaction Blocked for this IP Address
420	Transaction Blocked for this E-mail address
421	Transaction Blocked for this merchant

423	An error has occurred while processing your payment
424	Invalid Request URL

### Note:

The merchant should set the response messages on their websites regarding the statuses of each transaction. For example, when a transaction is completed, the customer should view a response as “Transaction Successful”. And when a transaction is not completed, the customer should view a response saying, “Transaction Incomplete”, along with other details.

## 2.5 Transaction Request

# Define ‘payment’ parameters

The format of “payment” parameter would be as follows:

**unique\_order\_id|total\_amount** (Eg : 12001|2567.50)

The above format will then be encrypted using the public key provided by WEBXPAY. This is achieved by using openssl\_public\_encrypt method and then needs to be encoded with base64\_encode method. The public key can be retrieved from the merchant backend under Settings.

(Public and secret keys will be generated once the merchant account is activated and can be accessed via Dashboard > Settings > Integration Information > Generate keys)

```
// unique_order_id|total_amount
$plaintext = '525|1000';
$publickey = "-----BEGIN PUBLIC KEY-----
MIGfMA0GCsQGIb3DQEBAQUAA4GNADCBiQKBgQC9l2HykxDIDVZeyDPJU4pA0imf
3nWsvyJgb3zTsnN8B0mFX6u5squ5NQcnQ03L8uQ56b4/isHBgiyKwfMr4cpEpCTY
/t1WSdJ5EokCI/F7hCM7aSSSY8557IY0iC6pKR4WbaOYMvAMKn5gCobEPtosmPLz
gh8Lo3b8UsjPq2W26QIDAQAB
-----END PUBLIC KEY-----";
//load public key for encrypting
openssl_public_encrypt($plaintext, $encrypt, $publickey);

//encode for data passing
$payment = base64_encode($encrypt);
```

# Define custom parameter

The format of the ‘custom\_fields’ parameter is as follows.

**parameter1|parameter2|parameter3|parameter4**

The “custom\_fields” parameter consists of above variables which then need to be encoded using base64\_encode.

```
//custom fields
//cus_1|cus_2|cus_3|cus_4
$custom_fields = base64_encode('cus_1|cus_2|cus_3|cus_4');
```

# Define the payment request URL

```
//checkout URL
$url = 'https://webxpay.com/index.php?route=checkout/billing';
```

## 2.5.1 Gateway ID's

Following is the available gateway ID's

Gateway ID	IPG	Display
2	eZ Cash (LKR)	eZ Cash (LKR)
3	mCash (LKR)	mCash (LKR)
4	Nations Trust Bank - AMEX (LKR)	AMEX (LKR)
8	Nations Trust Bank - AMEX (USD)	AMEX (USD)
5	Sampath Viswa (LKR)	Sampath Viswa (LKR)
47	Hatton National Bank (USD)	HNB (USD)
46	Hatton National Bank (LKR)	HNB (LKR)
16	DFCC Wallet (LKR)	DFCC - Wallet (LKR)
35	FriMi (LKR)	FriMi (LKR)
36	Seylan Bank (LKR)	Seylan Bank (LKR)
37	Seylan Bank (USD)	Seylan Bank (USD)
38	Commercial Token (LKR)	Commercial Token (LKR)
39	Commercial Token (USD)	Commercial Token (USD)
40	Commercial Bank MPGS (LKR)	Commercial Bank MPGS (LKR)
41	Commercial Bank MPGS (USD)	Commercial Bank MPGS (USD)
42	Genie Visa Master (LKR)	Genie Visa Master (LKR)
43	Cargils Bank Visa Master (LKR)	Cargils Bank Visa Master (LKR)
44	Cargills Bank Token (LKR)	Cargills Bank Token (LKR)
45	UPay (LKR)	UPay (LKR)
52	Promotional Gateway	V/M Promotional (LKR)
96	Lanka QR	Lanka QR

## 2.5.2 EMI Gateway ID's

Gateway ID	Bank	Tenor
23	NTB - AMEX	EMI 03 Month (LKR)
24	NTB - AMEX	EMI 06 Month (LKR)
25	NTB - AMEX	EMI 12 Month (LKR)
26	NTB - AMEX	EMI 20 Month (LKR)
27	NTB - AMEX	EMI 24 Month (LKR)
28	NTB - AMEX	EMI 36 Month (LKR)
48	HNB	EMI 3 Month (LKR)
49	HNB	EMI 6 Month (LKR)
50	HNB	EMI 12 Month (LKR)
51	HNB	EMI 24 Month (LKR)
53	DFCC	EMI 3 Month (LKR)
54	DFCC	EMI 6 Month (LKR)
66	DFCC	EMI 7 Month (LKR)
67	DFCC	EMI 9 Month (LKR)
55	DFCC	EMI 12 Month (LKR)
68	DFCC	EMI 15 Month (LKR)
69	DFCC	EMI 18 Month (LKR)
56	DFCC	EMI 24 Month (LKR)

70	DFCC	EMI 27 Month (LKR)
71	DFCC	EMI 36 Month (LKR)
72	DFCC	EMI 41 Month (LKR)
61	Commercial Bank	EMI 3 Month (LKR)
57	Commercial Bank	EMI 6 Month (LKR)
62	Commercial Bank	EMI 9 Month (LKR)
58	Commercial Bank	EMI 12 Month (LKR)
59	Commercial Bank	EMI 18 Month (LKR)
60	Commercial Bank	EMI 24 Month (LKR)
63	Commercial Bank	EMI 36 Month (LKR)
64	Commercial Bank	EMI 48Month (LKR)
65	Commercial Bank	EMI 60 Month (LKR)
75	NDB	EMI 6 Month (LKR)
76	NDB	EMI 9 Month (LKR)
77	NDB	EMI 12 Month (LKR)
78	NDB	EMI 18 Month (LKR)
79	NDB	EMI 24 Month (LKR)
80	NDB	EMI 36 Month (LKR)
81	NDB	EMI 60 Month (LKR)
90	Seylan	EMI 6 Month (LKR)
91	Seylan	EMI 12 Month (LKR)
92	Seylan	EMI 24 Month (LKR)
97	Union Bank	EMI 3 Month (LKR)
98	Union Bank	EMI 6 Month (LKR)
99	Union Bank	EMI 12 Month (LKR)
100	Union Bank	EMI 18 Month (LKR)
101	Union Bank	EMI 24 Month (LKR)
102	Union Bank	EMI 36 Month (LKR)

Existing parameter values will be posted via the html form below.

```
<form action="<?php echo $url; ?>" method="POST">
  First name: <input type="text" name="first_name" value="John"><br>
  Last name: <input type="text" name="last_name" value="Doe"><br>
  Email: <input type="text" name="email" value="customer_email@email.com"><br>
  Contact Number: <input type="text" name="contact_number" value="0773606370"><br>
  Address Line 1: <input type="text" name="address_line_one" value="46/46, Green Lanka Building"><br>
  Address Line 2: <input type="text" name="address_line_two" value="Nawam Mawatha"><br>
  City: <input type="text" name="city" value="Colombo"><br>
  State: <input type="text" name="state" value="Western"><br>
  Zip/Postal Code: <input type="text" name="postal_code" value="10300"><br>
  Country: <input type="text" name="country" value="Sri Lanka"><br>
  currency: <input type="text" name="process_currency" value="LKR"><br>
  Selected Gatewayid: <input type="text" name="payment_gateway_id" value=""><br>
  Bulk Payment Gateway IDs<input type="text" name="multiple_payment_gateway_ids" value="" placeholder="| seperated ids" > <br/>
  CMS : <input type="text" name="cms" value="PHP">
  custom: <input type="text" name="custom_fields" value="<?php echo $custom_fields; ?>">
  Mechanism: <input type="text" name="enc_method" value="JCs3J+6oSz4V0LgE0zi/Bg==">
<br/>
<!-- POST parameters -->
<input type="hidden" name="secret_key" value="630be963-59e2-447a-8f3b-93b3d7a3bf25" >
<input type="hidden" name="payment" value="<?php echo $payment; ?>" >
<input type="submit" value="Pay Now" >
</form>
```

Note:



If the “payment\_gateway\_id” parameter is set in the gateway request, corresponding payment option will be selected automatically, and the gateway payment selection page will be skipped.

## 2.6 Transaction Response

After payment process is completed along with the whole transaction process, the customer will be redirected to the callback URL. (You can set the call back URL from WEBXPAY Dashboard > Settings > Website Integration > on Add Return URL field)

*The callback process posts the following data to display the transaction status.*

1. payment
2. signature
3. custom\_fields

The “payment” parameter should be decode using base64\_decode method and then should be decrypted with openssl\_public\_decrypt methoc. The signature is included to avoid man-in-the middle attacks.

The format of the “payment” parameter would be as follows after decoding and decrypt methods are used.

order\_id|order\_reference\_number|date\_time\_transaction|status\_code|comment|payment\_gateway\_used  
E.g.

**12001|T372016I24522| 2016-04-20 09:57:52|15| No such issuer|1**

You need to compare the decoded payment value against the decoded and decrypted signature in order to confirm the posted parameters’ authenticity. (signature status)

```
//decode & get POST parameters
$payment = base64_decode($_POST ["payment"]);
$signature = base64_decode($_POST ["signature"]);
$custom_fields = base64_decode($_POST ["custom_fields"]);
//load public key for signature matching
$publickey = "-----BEGIN PUBLIC KEY-----
MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKBgQC9l2HykxDIDVZeyDPJU4pA0imf
3nWsvyJgb3zTsnN8B0mFX6u5squ5NQcnQ03L8uQ56b4/isHBgiyKwfMr4cpEpCTY
/t1WSdJ5EokCI/F7hCM7aSSSY85S7IYOiC6pKR4WbaOYMvAMKn5gCobEptosmPLz
gh8Lo3b8UsjPq2W26QIDAQAB
-----END PUBLIC KEY-----";
openssl_public_decrypt($signature, $value, $publickey);

$signature_status = false ;

if($value == $payment){
    $signature_status = true ;
}
```

If the signature status is true, you can proceed to further steps.

### 3 Sample Code (PHP)

```
<?php

// unique_order_id|total_amount
$plaintext = '525|1000';
$publickey = "-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC9I2HykxDIDVZeyDPJU4pA0imf
3nWsvyJgb3zTsnN8B0mFX6u5squ5NQcnQ03L8uQ56b4/isHBgiyKwfMr4cpEpCTY
/t1WSdJ5EokCI/F7hCM7aSSSY85S7IYOiC6pKR4WbaOYMvAMKn5gCobEPtosmPLz
gh8Lo3b8UspQ2W26QIDAQAB
-----END PUBLIC KEY-----";

//load public key for encrypting
openssl_public_encrypt($plaintext, $encrypt, $publickey);

//encode for data passing
$payment = base64_encode($encrypt);

//checkout URL
$url = 'https://webxpay.com/index.php?route=checkout/billing';

//custom fields
//cus_1|cus_2|cus_3|cus_4
$custom_fields = base64_encode('cus_1|cus_2|cus_3|cus_4');

?>
```

The following sample code depicts a sample request form with the required parameters.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="author" content="John Doe">
<title>WebXPay | Sample checkout form</title>
</head>

<body>
<form action="<?php echo $url; ?>" method="POST">
    First name: <input type="text" name="first_name" value="John"><br>
    Last name: <input type="text" name="last_name" value="Doe"><br>
    Email: <input type="text" name="email" value="customer_email@email.com"><br>
    Contact Number: <input type="text" name="contact_number" value="0773606370"><br>
    Address Line 1: <input type="text" name="address_line_one" value="46/46, Green Lanka Building"><br>
    Address Line 2: <input type="text" name="address_line_two" value="Nawam Mawatha"><br>
    City: <input type="text" name="city" value="Colombo"><br>
    State: <input type="text" name="state" value="Western"><br>
    Zip/Postal Code: <input type="text" name="postal_code" value="10300"><br>
    Country: <input type="text" name="country" value="Sri Lanka"><br>
    currency: <input type="text" name="process_currency" value="LKR"><br>
    Selected Gatewayid: <input type="text" name="payment_gateway_id" value=""><br>
    Bulk Payment Gateway IDs<input type="text" name="multiple_payment_gateway_ids" value="" placeholder="|
    seperated ids" > <br>
    CMS : <input type="text" name="cms" value="PHP">
    custom: <input type="text" name="custom_fields" value="<?php echo $custom_fields; ?>">
    Mechanism: <input type="text" name="enc_method" value="JCs3J+6oSz4V0LgE0zi/Bg==">
```

## WEBXPAY Gateway Integration Guide v2.7

```

<br/>

<!-- POST parameters -->
<input type="hidden" name="secret_key" value="630be963-59e2-447a-8f3b-93b3d7a3bf25" >
<input type="hidden" name="payment" value="<?php echo $payment; ?>" >
<input type="submit" value="Pay Now" >

</form>
</body>
</html>

```

The following sample code depicts the response from WEBXPAY Gateway along with the POST parameters. The same library is used to validate the checksum of the response string using the public key provided to the merchant by WEBXPAY.

```

<?php

//decode & get POST parameters
$payment = base64_decode($_POST ["payment"]);
$signature = base64_decode($_POST ["signature"]);
$custom_fields = base64_decode($_POST ["custom_fields"]);

//load public key for signature matching
$publickey = "-----BEGIN PUBLIC KEY-----
MIGfMA0GCsQGSib3DQEBAQUAA4GNADCBiQKBgQc9l2HykxDIDVZeyDPJU4pA0imf
3nWsvyJgb3zTsnN8B0mFX6u5squ5NQcnQ03L8uQ56b4/isHBgiyKwfMr4cpEpCTY
/t1WSdJ5EokCI/F7hCM7aSSSY85S7IYOiC6pKR4WbaOYMvAMKn5gCobEPtosmPLz
gh8Lo3b8UsjPq2W26QIDAQAB
-----END PUBLIC KEY-----";

//Decrypt data
openssl_public_decrypt($signature, $value, $publickey);

//Check signature status (Default False)
$signature_status = false ;

if($value == $payment){
    $signature_status = true ;
}

//get payment response in segments
//payment format: order_id|order_reference_number|date_time_transaction|payment_gateway_used|status_code|comment;
$responseVariables = explode('|', $payment);

if($signature_status == true)
{
    //display values
    echo $signature_status;
    $custom_fields_variable = explode('|', $custom_fields);
    var_dump($custom_fields_variable);
    echo '<br/>';
    var_dump($responseVariables);
}else
{
    echo 'Error Validation';
}

?>

```

## 4 WEBXPAY Plugins & SDK

WEBXPAY supports Opencart, Magento, Wordpress (Woo Commerce), Prestashop, WHMCS, and Shopify. You can download the WEBXPAY plugins and payment gateway modules from your merchant dashboard or by using the following links.

[https://webxpay.co/System\\_Support/Downloads.zip](https://webxpay.co/System_Support/Downloads.zip)

Or visiting the following link.

<https://webxpay.co/Integration/Integration-pack.zip>

### 4.1 Opencart Integration

1. Download WEBXPAY plugin for OpenCart from the Dashboard > Go to Downloads.
2. Copy the files in the zip file into your OpenCart server directory.
3. Sign in to your OpenCart admin.
4. Go to Extensions > Payment's tab on tab filter.



5. Find WEBXPAY
6. Click "Edit" to manage WEBXPAY plugin settings
7. Enter your public and secret keys.
8. Enter the redirection URL which the customer will be redirected to after a transaction is completed
9. Save your plugin settings.

from the list and click "Install".

	Disabled		 
	Disabled		 
	Disabled		 

\* Title

Card / Mobile Wallet Payment

\* Redirect Link

<https://webxpay.com/index.php?route=checkout/billing>

\* Order Status

Canceled

\* Secret Key

D15AA1E2482BCD6D56E5F3A4B5DE4

\* Public Key

-----BEGIN PUBLIC KEY-----

MIGfMA0GCQsQGSib3DQEBAQUAA4GNADCBiQKBgQChet+9WAmmlEmuhanqG2i6Rh4M

+BbsvFYUSLLE4fd1sbqg4Lxw6a4gMXNV1/ot4Qbo1MMY7PA/s1feCW/wZ0vouL/

zQsbhArFDGz3bB5+yATG5BZfWPlmXpahGmWjEUMAh61QVsuHBe8qsPz/jmxSRT+

yAmaqEstrn8Uqk7Uj9QIDAQAB

-----END PUBLIC KEY-----

Status

Disabled

### 4.2 WooCommerce Integration

## WEBXPAY Gateway Integration Guide v2.7

1. Download WEBXPAY plugin for WooCommerce from the Dashboard > Go to Downloads.
2. Install the plugin via your admin panel as other Wordpress plugin installations. Activate the plugin.
3. On your dashboard, go to WooCommerce > Settings and then “Payments” tab.
4. Find the WEBXPAY plugin and enable it and click “Manage” button to manage plugin integration.

<div> <div> <div></div> <div></div> <div></div> </div> <div>Direct bank transfer</div> </div>	<div> <div></div> <div>Take payments in person via BACS. More commonly known as direct bank/ wire transfer.</div> </div>	<div>Finish setup</div>
<div> <div> <div></div> <div></div> <div></div> </div> <div>Check payments</div> </div>	<div> <div></div> <div>Take payments in person via checks. This offline gateway can also be useful to test purchases.</div> </div>	<div>Finish setup</div>
<div> <div> <div></div> <div></div> <div></div> </div> <div>Cash on delivery</div> </div>	<div> <div></div> <div>Have your customers pay with cash (or by other means) upon delivery.</div> </div>	<div>Finish setup</div>
<div> <div> <div></div> <div></div> <div></div> </div> <div>WebXpay - WEBXPAY</div> </div>	<div> <div></div> <div></div> </div>	<div>Manage</div>

5. Provide relevant public and secret keys.

PG Domain:	<input type="text" value="https://webxpay.com/index.php?route=checkout/billing"/>
	Payment process URL
Secret Key:	<input type="text" value="3a1f027d-6573-4455-bc3a-e15ceaa49a98"/>
	Secret key given by WebXpay.
Public Key:	<div> <div>-----BEGIN PUBLIC KEY-----</div> <div>MIGfMA0GCsQGSib3DQEBAQUAA4GNADCBiQKBgQC3OINoDv8NGgUoRdYmT9mbeT9y w2jNNueNevaTSes+5Qnle5G7rk6T0TPwgKkQLQo+KTZy6+f7gma5r6oX3mIHSe8g</div> </div>
	Public key given by WebXpay.
Checkout Message:	<input type="text" value="Thank you for your order. Please click the button below to pay with WebXpay."/>
	Message display when checkout

6. Provide the redirection URL where you want the customer to be redirected to after transaction. (On your WEBXPAY Dashboard > Settings> Website Integration> Redirect URL field)

\*\*\* Redirection URL Format should be as follows:

<https://yourdomain.com/checkout/order-received/>

7. Save Changes.

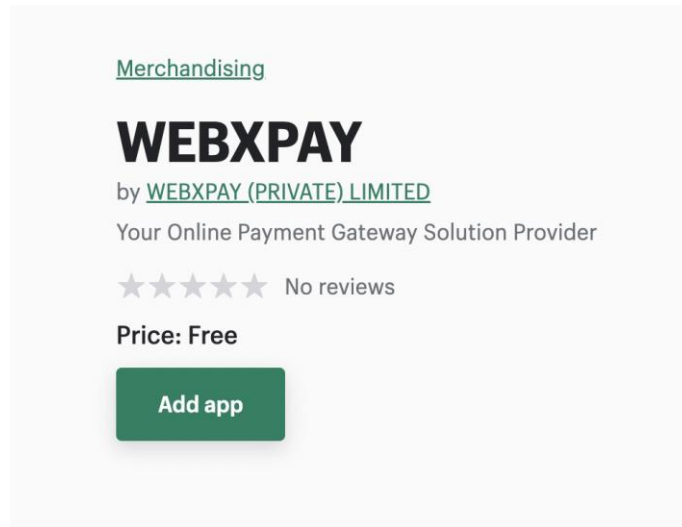
## 4.3 Shopify Integration

1. Access to shopify plugin for WEBXPAY from the following link.

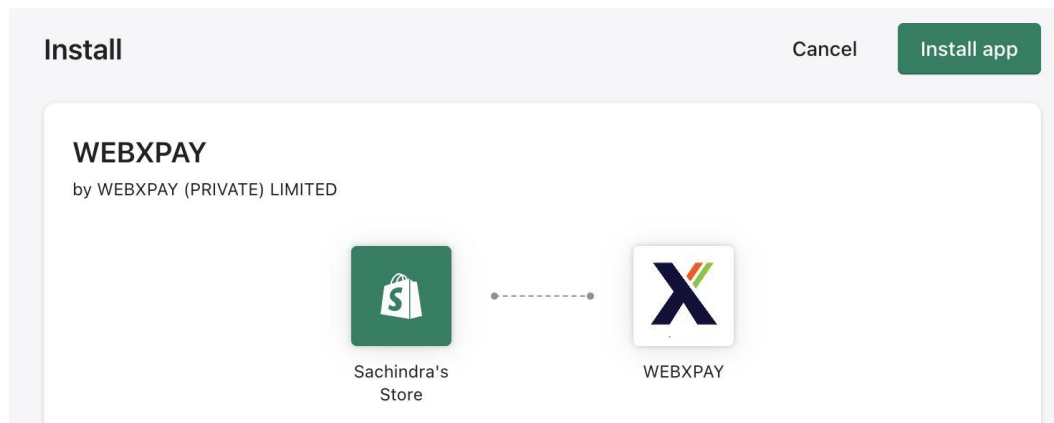
<https://apps.shopify.com/webxpay>

2. Add the plugin to your Shopify account.

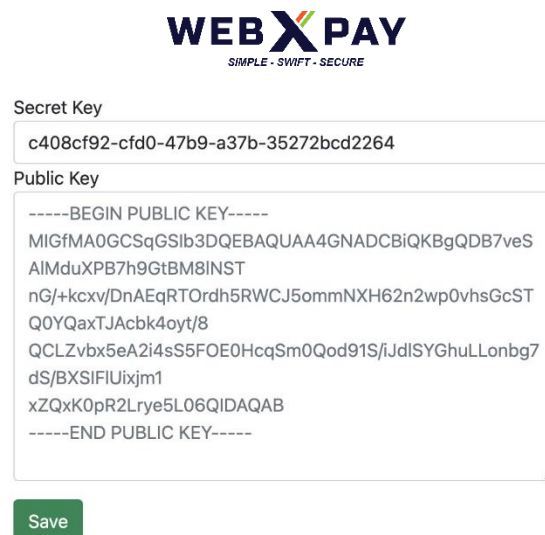
## WEBXPAY Gateway Integration Guide v2.7



3. Install the app.



4. Click manage to provide the public and secret keys from your WEBXPAY account and save.



**WEBXPAY**  
SIMPLE · SWIFT · SECURE

Secret Key

c408cf92-cfd0-47b9-a37b-35272bcd2264


Public Key


-----BEGIN PUBLIC KEY-----  
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDB7veS  
AlMduXPB7h9GtBM8INST  
nG/+kcxv/DnAEqRTOrdh5RWCJ5ommNXH62n2wp0vhsGcST  
Q0YQaxTJAcbk4oyt/8  
QCLZvbx5eA2i4sS5FOE0HcqSm0Qod91S/iJdlSYGhuLLonbg7  
dS/BXSIUixjm1  
xZQxK0pR2Lrye5L06QIDAQAB  
-----END PUBLIC KEY-----


[Save](#)


## WEBXPAY Gateway Integration Guide v2.7


5. Select the payment methods and activate the plugin. (Enable test mode if necessary)


☒  Visa


☒  Mastercard

☒  American Express

☒  Discover

☒  Diners Club

☒  AliPay

☒  WeChat Pay

**Test mode**

Test your WEBXPAY setup by simulating successful and failed transactions.

☐ Enable test mode

Uninstall WEBXPAY




Activate WEBXPAY

6. Add the following return URL to your account. Go to Dashboard > Settings > Website integration.

<https://webxpay.secureapps.center/reply.php>

## 4.4 Prestashop Integration

1. Download WEBXPAY plugin for PrestaShop from the Dashboard > Go to Downloads.
2. Upload the plugin file into your PrestaShop server's "Modules" folder.
3. Log into your admin dashboard and go to Modules > Module Catalog and then from Categories tab, choose "Payment" tab. Find the WEBXPAY payment method and install.
4. Go to Payment > Payment methods. Under WEBXPAY payment, click "Configure".

	<b>Bank transfer v2.1.1 - by PrestaShop</b> Accept wire payments by displaying your account details during the checkout.	<a href="#">Configure</a>
	<b>PrestaShop Checkout v2.17.1 - by PrestaShop</b> Provide the most commonly used payment methods to your customers in this all-in-one module, and manage all your sales in a centralized interface.	<a href="#">Configure</a>
	<b>WEBXPAY v1.0.0 - by WEBXPAY (PVT) LTD</b> Accept payments for your products via WEBXPAY.	<a href="#">Configure</a>

5. Provide your Public and Secret keys. Also set the Redirect URL to your preferred redirect URL which your customers will be redirected to after each payment transaction. (The same redirect URL has to be included in WEBXPAY Dashboard > Settings > Return URL)

## WEBXPAY Gateway Integration Guide v2.7

\* Test Mode ☐ False  
While on test mode, the transaction amount will be set as 10.

**IMPORTANT:** Please note that, making test mode payment is actually controlled by enabling the *Sandbox* Activation checkboxes on your WEBXPAY settings at <https://webxpay.com -> Integrations>

\* WEBXPAY Public Key

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCsGSIb3DQEBAQUAA4GNADCBiQKBgQDkrmZEtKuuNrpU4Ck9PIYumwD
3eYbeKc5aA8BOCIECmIgcOstUjzzd1sf/16/aaJdDAOpAxkEk1xhn7nPRtbN6DKpP
Wm8TPNbl38QikGdWkerCcO2s5sd2TCkndsSbc+F1EtFX+OHdG4eNdN5EcIm5H4LC
+LMaIeRAPp81Lx/drW/DAQAB
-----END PUBLIC KEY-----
```

\* WEBXPAY Secret Key

b83b77ff-25e4-4717-b6b9-8223be5c8ae1

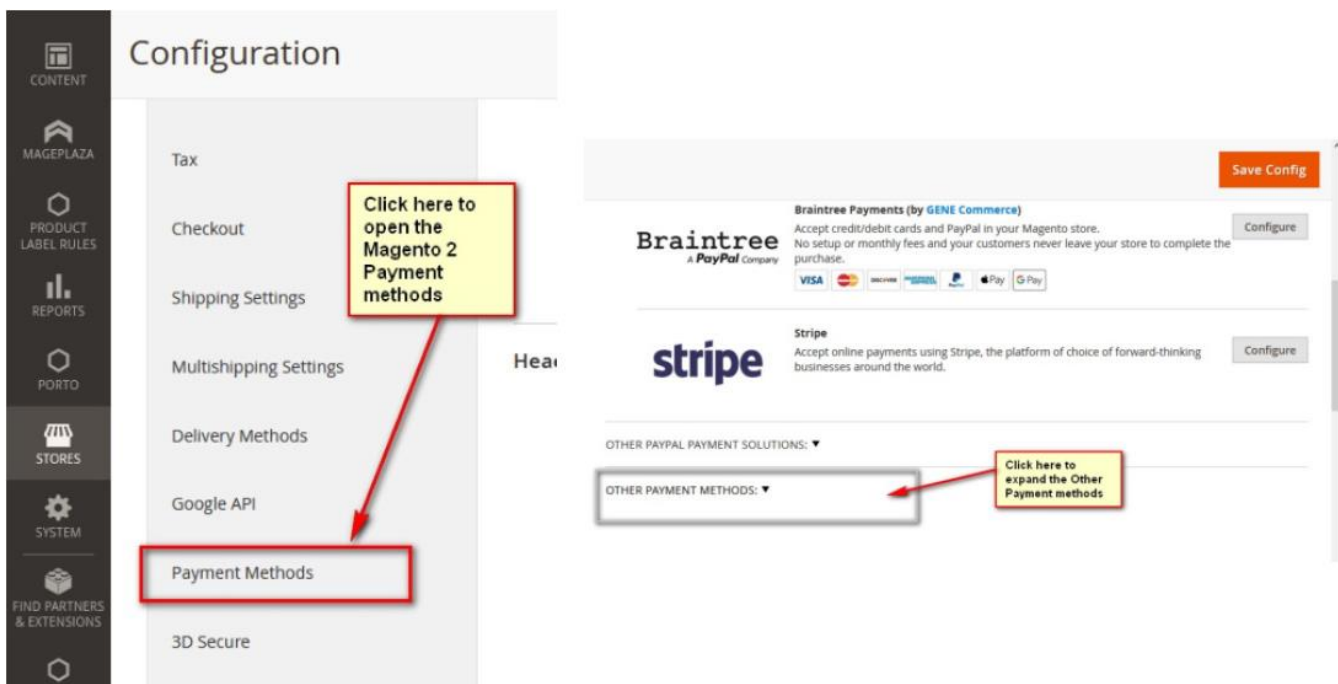
Set your **Callback / Response** URLs at WEBXPAY to:  
<http://localhost:8888/pesttrashop/en/module/webxpay/webxpaysuccess>

\* Failed Payment Order ☐ False  
*If enabled:* Cart will become empty, and converted to new order for both successful & failed payment.  
*If disabled:* Cart will be intact (not converted to new order) for failed payment; hence, the failed attempt will not be redirected to order-information page.

6. Save changes.

## 4.5 Magento Integration

1. Download WEBXPAY plugin for Magento from the downloads link.
2. Upload the plugin file to your server. (Directory: app/code).
3. Run the following command to install the plugin.  
**bin/magento setup:upgrade**
4. In your admin panel, go to Stores > Configuration > Sales > Payment Methods.



5. Provide your Public and Secret keypair along with Redirect URL on your merchant dashboard. (Dashboard > Settings > Website Integration > Return URL)



## WEBXPAY Gateway Integration Guide v2.7

\*\*\*\*\*

<b>WEBXPAY Live Public Key</b> <small>[website]</small>	----BEGIN PUBLIC KEY---- <div></div>
<b>WEBXPAY Live Secret Key</b> <small>[website]</small>	<div></div>
<b>WEBXPAY Test PUBLIC Key</b> <small>[website]</small>	----BEGIN PUBLIC KEY---- <div></div>
<b>WEBXPAY Test Secret Key</b> <small>[website]</small>	<div></div>

### 4.6 WHMCS Integration

1. Download WEBXPAY payment gateway module for WHMCS from the downloads link.
2. Upload the plugin file into your WHMCS server's "Modules > Gateways" folder.
3. Log into your admin dashboard. Go to System Settings > Payments > Payment Gateways. Find the gateway for WEBXPAY from the "All Payment Gateways" tab and activate it.
4. Enter public and secret keys provided in your merchant dashboard under Dashboard > Settings > Website Integration > Integration. Also include the return URL in your Dashboard > Settings > Website Integration > Add Return URL.
5. Save settings in both the Merchant Dashboard and the WHMCS admin panel.

### 5 Test Cards

**4111 1111 1111 1111 - Visa With 3DS**

**5111 1111 1111 1118 - Master Without 3DS**

**4564 4564 4564 4564 – Union Pay**

**3400 0009 990 0051 – Amex With 3DS**

**4508 7500 1574 1019 - Visa With - 3DS**

**4012 0000 3333 0026 - Visa Without 3DS**

**5123 4500 0000 0008 - Master With - 3DS**

**(with any future expiry and 3 digits of CVV)**

*For further queries, please contact us via the following details.*

*Phone: 0117 441 900*

*E mail: [technical@webxpay.com](mailto:technical@webxpay.com)*

\*\*\*\*\*