



Verifone Payment gateway for Prestashop eCommerce platform

Table of contents

1. Introduction
2. Installation and Configuration
3. Payment service key handling
4. Server configuration
5. Late shop payment success page URL
6. Troubleshooting
7. Order states and inventory handling
8. Refund feature
9. Configuration summary
10. Payment method name in invoice pdf file

1. Introduction

This document is the Verifone payment module installation and configuration guide.

Verifone payment module offers Prestashop integration towards the Verifone payment service.

2. Installation and Configuration

1. Go to the **Modules > Modules & Services** screen.
2. At the top, there is a button labelled **Upload A MODULE**. Click that button, and a new area of the page appears.
3. Into that area, you can either drag-and-drop or click the **select file** link to use your system's file browser to find it.
4. After installation, the **Configure** button can be clicked to set the configuration for the module.
5. You can also configure module using this path: **Modules > Modules & Services > Installed modules > Verifone**, and click the **Configure** button
6. Enter values for Configuration (on screenshot values for a **test environment**):
7. Save configuration.
8. Click "**Refresh Payment methods**" to get all payment methods. By using the "**Available payment method**" you may show payment method as separated Payment Method and the customer will be redirected directly to the bank or card data form. If you wish to use

Verifone Gateway then select “Verifone Payment – All in One”. TIP: use CTRL + [click] to select multiple options.

9. For check stored configuration, please click on **Display configuration summary**. More information in section 9.
10. Due to PSD2 regulation phone number is mandatory in payment request. As a default, this field is optional and must be configured in the **Admin Panel > Customers > Addresses**
11. Click on **Set required fields for this section** button. From list available below select **phone** fields and click the **Save** button

Display configuration summary [📄 Display configuration summary](#)

Title Verifone Payment

Environment (Test/Production) Test

Select environment for the payment module

Verifone Payment production merchant agreement code EC1234567890

Verifone Payment production merchant agreement code

Verifone Payment test merchant agreement code demo-merchant-agreement

Verifone Payment test merchant agreement code

Payment service key handling Automatic (Simple)

Generate new keys 🔑 Generate live keys

Uses preset keys by default, only needed if using custom test agreements

🔑 Generate test keys

When you generate live keys, you will need to upload the new public key to Verifone Payment portal

Pay page URL 1 https://epayment1.point.fi/pw/payment

Required. URL to the payment system

Pay page URL 2 https://epayment2.point.fi/pw/payment

Optional. Second redundant URL to the payment system

Pay page URL 3

Optional. Third redundant URL to the payment system

Payment page language English

Select language which will be use on payment page

Check payment node availability YES NO

Make a check that payment node is available

Skip confirmation page YES NO

Return directly to shop after payment

Style code

Use of custom payment page template needs first to be uploaded and to be approved by Verifone

Basket Item Sending Send for all payment method

Select for which type of order should send items.

Combine Invoice Basket Items YES NO

Currently only available for invoice payment methods

Use external customer id.

If you want to use other field to match user in Verifone Payment service, please set this value (for example billing_phone)

Refresh Payment Methods



Available payment methods

- All in one
- Master Card
- Enterpay Lasku
- Diners
- American Express
- Handelsbanken
- Aktia
- Nordea
- POP Pankki
- Säästöpankki
- Ålandsbanken
- Oma Säästöpankki
- Collector Lasku
- Visa
- Euroloan Lasku
- Svea Lasku

Allow to save Credit Cards YES NO

Save masked PAN number YES NO

Remember me info

Optional. Note in checkout after Remember payment method checkbox - for cards only

Minimum Order Total

Leave empty to disable limit.

Maximum Order Total

Leave empty to disable limit.

Disable rsa blinding YES NO

Define CRYPT_RSA_DISABLE_BLINDING as true in case of custom PHP build or PHP7 (experimental)

Pending payment order status Verifone Payment - payment

Success order status Verifone Payment - payment

Fail/Cancel order status Canceled

Secret Token f7fc063e11e683d0ba0ff91e03a257e2ea9bce67

Make this long and hard to guess

Cron job URL http://prestashop1761.local/en/module/verifone/checkCron?token=f7fc063e11e683d0ba0ff91e03a257e2ea9bce67

Set a cron job to call this URL

Configuration setting	Description
Display configuration summary	Possibility to display summary for shop configuration. Available information about which merchant agreement or keys is used. Before check configuration for test/live you need first save the configuration. See section 9 for more information.
Title	Name of the payment module. This name is visible to the customer as the name of the payment.
Environment (Test/Production)	Select the environment for the payment module
Verifone Payment production merchant agreement code	Unique seller identification obtained from Verifone.
Verifone Payment test merchant agreement code	Seller identification used for the test.
Payment service key handling	Select which key handling mode you will use. See section 3 for more information.
Generate new keys	Functionality for creating a new key pair for the shop. See section 3 for more information.
Directory for store keys	A path to the directory for generated files Important: Do not place keys in a folder accessible by the WWW-server.
Shop private key filename	The filename for the shop key file. Important: Do not place keys in a folder accessible by the WWW-server.
Shop private key filename (Test)	The filename for the shop key file used for the test. Important: Do not place keys in a folder accessible by the WWW-server.
Pay page URL 1,2,3	Payment system endpoint address (payment node). For redundancy, it is possible to have up to three separate endpoints. The payment module will choose the endpoint used randomly.
Payment page language	Select the language which will be used on the payment page

Check payment node availability	Before payment, the payment node is polled in order to check its availability. In case a certain payment node is not available, another one is checked. In case no payment node is available an error is raised. The availability check can be bypassed with this configuration setting.
Skip confirmation page	In the Verifone payment system, it is possible to skip the last page and direct user directly to the shop. With this selection, the functionality can be turned on/off.
Style code	Code identifying the style sheet used in payment page or an empty string if the default style sheet is used. Use of custom payment page template needs first to be uploaded and to be approved by Verifone.
Basket Item Sending	Select for which type of order the items should be sent. When you select the option for send items, then the product list is provided for Verifone payment.
Combine Invoice Basket Items	If yes, then all products are sent to Verifone as one product with name Tilaus %ORDERNUMBER%
Refresh Payment methods	Option to get payment methods from Verifone gate and redirect user directly to the payment page, dodging portal step
Available payment methods	Select visible payment methods. To fill the list click „ Refresh Payment methods ” button.
Allow to save Credit Cards	Allow customers to store credit cards in Verifone for future payments.
Save masked PAN number	Save masked pan number to order
Remember me info	Information visible when Allow to save Credit Cards is set to Yes
Minimum Order Total	Leave empty if you do not want to limit order minimum
Maximum Order Total	Leave empty if you do not want to limit order maximum

Disable rsa blinding	Define CRYPT_RSA_DISABLE_BLINDING as true in case of custom PHP build or PHP7 (experimental)
Pending payment order status	Set status for order when payment is pending
Success order status	Set status for order when payment is done and received
Fail order status	Set status for order when payment is cancelled
Secret Token	Generated automatically
Cron Job URL	Set a cron job to call this URL (for example in DirectAdmin)

3. Payment service key handling

For testing payments, the module contains test keys, and uses it, when the simple mode is configured.

Automatic (Simple)

- When everything is configured, please run **Generate live keys** (for production) or **Generate test key** (for development/test). Confirmation for this action is required.
- Generation a new test key is required only when custom merchant agreement code is configured in field **Verifone Payment test merchant agreement code**
- After creating a new key, remember to copy the public key to payment operator configuration settings. The public key is available in the **Configuration summary**.
- **Generate new keys to** override old keys. Please don't click this link if you are not sure.

Manual (Advanced)

- This configuration could be used for keys created in the old way.
- Please configure the directory path where keys are stored.
- Please configure file name for production and/or test environment.

By clicking on Display configuration summary you can check which keys are being used. If you encounter any problems, please see section 6. Troubleshooting

4. Server configuration

Required configuration for the http-server:

- SSL-support installed and enabled (e.g. OpenSSL for Apache web server)
- allow_url_fopen in PHP configuration must be set on.

Cron is required for automatic check payment status for orders which has a pending status. Url

for cron is provided in the module configuration in the **Cron Job URL** field. Please add this URL into crontab list on the server or in the hosting admin panel.

This configuration depends on the provider, and please check in the hosting provider help page

for more information.

5. Late/delayed shop payment success page URL

The late shop payment success page URL is:

https://<yourdomain>/module/verifone/paymentDelayed or
http://<yourdomain>/module/verifone/paymentDelayed

depending on whether SSL is used or not.

Verifone may request this information for production use web shops.

Full delayed url is available in the module information view, after a click on **“Display configuration summary”**

6. Troubleshooting

Problem	Resolution
Verifone payment is not visible in payment selection	Check configuration values. Check that the module is enabled.
Error: Payment system not available.	The payment system availability check has failed. Check Pay Page URL parameters.
Error: Problem with generating new keys.	Check directory configuration. The directory must be accessible and writable.
Error: Problem with directory configuration.	Check directory configuration. The directory must be accessible and writable.
Error: Key file for provided merchant agreement code does not exist	This problem appears, when is set merchant agreement code, or custom key file name, but the key does not exist. Check configuration, and if required, then create new keys. Remember about upload into payment service.
Info: Default test key file uses	The plugin uses default keys, provided with the module. It is using for the default configuration.
Error: Key file could not be read.	The payment module has not been able to read the key file. Possible causes are: <ul style="list-style-type: none"> • The key file is not stored to the server • Key file path or filename is wrong • Key file does not have reading rights. Check that the key file exists in the server, check that the path and file name is correct and check file permissions.
Error: Payment verification failed	The payment verification has failed. Check that the key file is correct.
Error: Configuration error: allow_url_fopen must be enabled	Change allow_url_fopen true in PHP

7. Order states and inventory handling

Order state before payment has been accepted will be gotten from '**Pending Payment order status**'.

For successfully paid orders, the order state will be gotten from '**Processing order status**'.

For cancelled orders, the state will be gotten from '**Canceled order status**'. Order cancellation will be done, when the customer cancels payment in the Verifone payment system and returns to the shop, or payment is not finalized.

Note that the order is created when the customer begins the payment process. The sole existence of an order does not indicate, that the order is actually ordered and paid. Only orders, which are on the state will be gotten from '**Processing order status**', are to be delivered.

When stock management is enabled, the ordered quantity is reserved from the stock at the time of order creation. For orders, where the payment process is never completed nor cancelled, the order with associated stock reservation remains indefinitely at

'Pending Payment order status' configuration. Manually cancelling the orders will return items to the stock.

79	USGSQGVKF	No	Finland	S. Lamia Testi	€37.82	Verifone	Verifone ePayments - payment pending
78	BQWNQBNUG	No	Finland	S. Lamia Testi	€37.82	Verifone	Canceled
77	CKDEGYMZM	No	Finland	S. Lamia Testi	€20.47	Verifone	Verifone ePayments - payment success

At the example above order:

- #79 is waiting for payment.
- #77 has been paid and can be delivered,
- #78 has been cancelled.

8. Refund feature

Refund is available by functionality provided by PrestaShop, on the order details page after clicking the **Partial refund** button.

To refund money, you need to set **Quantity**. If you want, you can change the refunded amount, for example for deduct shipping fee.

Generate a credit slip checkbox must be checked, otherwise, refund will be not proceed into the customer account.

When you fill everything, you need to click the **Partial refund** button (the blue one).

If necessary, it is possible to create multiple transactions, for example for each product.

Display configuration summary  Display configuration summary

Title Verifone

Environment (Test/Production) Test ▼

Select environment for the payment module

1. Default test configuration - use for testing with default keys.

Configuration summary ✕

Mode	Test
Verifone Payment merchant agreement code	demo-merchant-agreement
Default test merchant agreement uses	
Delayed success url	http://prestashop.local/en/module/verifone/paymentDelayed
This is the url that you need to copy to payment provider settings in their portal.	
Key handling mode	Automatic (Simple)
Path and filename of Verifone Payment public key file	/var/docker_storage/workspace/verifone-shops/prestashop/modules/verifone/keys/verifone-e-commerce
Key file is available	
Path and filename of shop private key file	Key file stored in database
Default key file is used	

Close

2. Custom test configuration - use for testing with custom merchant agreement code and keys

Configuration summary



Mode	Test
Verifone Payment merchant agreement code	test-merchant-agreement
Delayed success url	http://prestashop.local/en/module/verifone/paymentDelayed This is the url that you need to copy to payment provider settings in their portal.
Key handling mode	Manual (Advanced)
Path and filename of Verifone Payment public key file	/var/docker_storage/workspace/verifone-shops/prestashop/modules/verifone/keys/verifone-e-commerce Key file is available
Directory for store keys	/var/docker_storage/workspace/keys Directory configured properly
Path and filename of shop private key file	/var/docker_storage/workspace/keys/test-merchant-private.pem Key file is available

Close

3. Production configuration

Configuration summary



Mode	Production
Verifone Payment merchant agreement code	EC1234567890
Delayed success url	http://prestashop.local/en/module/verifone/paymentDelayed This is the url that you need to copy to payment provider settings in their portal.
Key handling mode	Automatic (Simple)
Path and filename of Verifone Payment public key file	/var/docker_storage/workspace/verifone-shops/prestashop/modules/verifone/keys/verifone-e-commerce Problem with load key file. Please contact with customer service
Path and filename of shop private key file	Key file stored in database Key file is available
Public key file	-----BEGIN PUBLIC KEY----- MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCekbn1nbw1HX4ZtCttaXnYEcE Mmm0g2cjJuiYFnzBgH8p814V40b+nf+6clXkQvuCsOARKFhsqUYlVuZE3fgYu9JP nZxwJNgRIKcrEWK4QaXoc63qgwT+G35cZCNebrBDCR5agCX4bwqF6dRqfTvqzZGu KbWRxbe8T5ulgHDVtwIDAQAB -----END PUBLIC KEY----- Please, copy this key to payment operator configuration settings, otherwise, the payment will be broken.

Close

10. Payment method name in invoice pdf file

In case, when you want to display full payment method on the PDF invoice file, you need to modify template file.

Default template file with name **invoice.payment-tab.tpl** is available in **pdf** directory. In case when you are using custom template, you need edit tpl file there.

You need to edit the file, and replace line 32 (in the original file):

```
<td class="right small">{$payment->payment_method}</td>
```

You have a couple of possibilities to display. In example for the Visa payment method you can display it like:

- **Verifone Payment [Visa]**

```
{assign var='payment_method' value=Verifone::getPaymentName($payment)}
```

```
<td class="right small">{$payment_method}</td>
```

- **Visa**

```
{assign var='payment_method' value=Verifone::getPaymentName($payment, false)}
```

```
<td class="right small">{$payment_method}</td>
```

- **prefix_Visa** (for example Online: Visa)

```
{assign var='payment_method' value=Verifone::getPaymentName($payment, 'Online: ')}
```

```
<td class="right small">{$payment_method}</td>
```

Instead of one line, you need to have 2 lines. This modification works just for Verifone payments.

Example implementation:

```
<table id="payment-tab" width="100%">
  <tr>
    <td class="payment center small grey bold" width="44%">{l s='Payment Method' d='Shop.Pdf' pdf='true'}</td>
    <td class="payment left white" width="56%">
      <table width="100%" border="0">
        {foreach from=$order_invoice->getOrderPaymentCollection() item=payment}
          <tr>
            {assign var='payment_method' value=Verifone::getPaymentName($payment, false)}
            <td class="right small">{$payment_method}</td>
            <td class="right small">{displayPrice currency=$payment->id_currency price=$payment->amount}</td>
          </tr>
        {/foreach}
      </table>
    </td>
  </tr>
</table>
```