Verifone Payment gateway for Drupal Commerce 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. Configure customer identifier field

1. Introduction

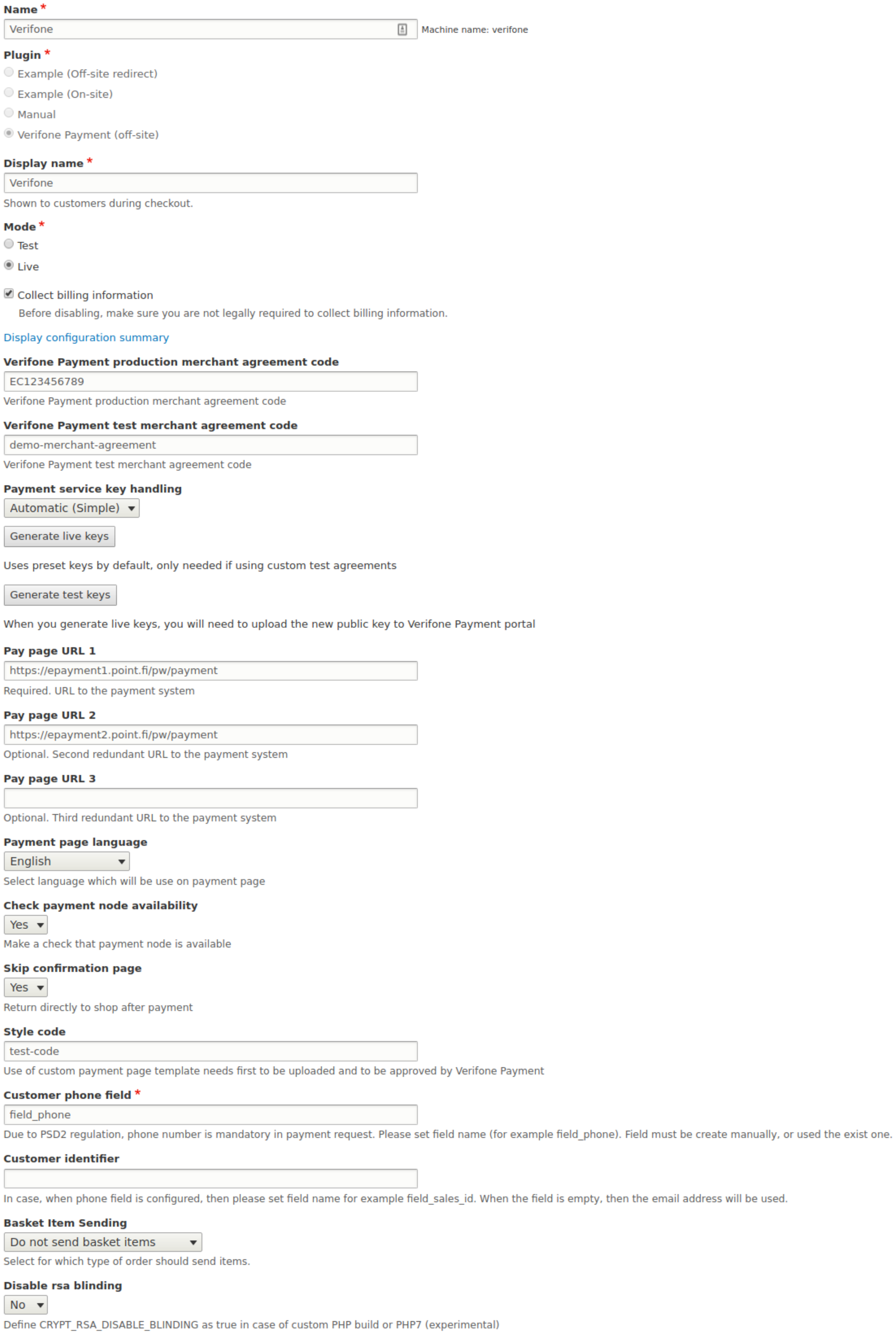
This document is the Verifone payment module installation and configuration guide.

Verifone payment module offers Drupal Commerce integration towards the Verifone payment service.

2. Installation and Configuration

1. Navigate to your Drupal Commerce project directory and update your composer.json file: **composer require 'drupal/commerce_verifone:^2.1'**
2. Wait for Composer to finish updating your project dependencies and make sure there aren’t any errors.
3. Go to the admin panel **Extend > Commerce tab** and check the checkbox next to the **Commerce Verifone** label.
4. Click the **Install** button.
5. After installation go to the **Commerce > Configuration > Payment Gateways** and click **Add new payment gateway**
6. Check radio input next to the **Verifone (off-site)**
7. Enter values for Configuration (on screenshot below values for a **test environment**)

8. Save configuration.
9. For check stored configuration, please click on **Display configuration summary.** More information in section 9.



Name *

Verifone — Machine name: verifone

Plugin *

○ Example (Off-site redirect)

○ Example (On-site)

○ Manual

● Verifone Payment (off-site)

Display name *

Verifone

Shown to customers during checkout.

Mode *

○ Test

● Live

☑ Collect billing information

Before disabling, make sure you are not legally required to collect billing information.

Display configuration summary

Verifone Payment production merchant agreement code

EC123456789

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 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 ▾

Make a check that payment node is available

Skip confirmation page

Yes ▾

Return directly to shop after payment

Style code

test-code

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

Customer phone field *

field_phone

Due to PSD2 regulation, phone number is mandatory in payment request. Please set field name (for example field_phone). Field must be create manually, or used the exist one.

Customer identifier

In case, when phone field is configured, then please set field name for example field_sales_id. When the field is empty, then the email address will be used.

Basket Item Sending

Do not send basket items ▾

Select for which type of order should send items.

Disable rsa blinding

No ▾

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

Configuration setting	Description
Name	Name of the payment module. This name is visible to the customer as the name of the payment.
Display name	Name of the payment module. This name is visible to the customer as the name of the payment.
Mode	Select if the extension is on production mode (Live) or test mode (Test)
Collect billing information	When checked, then billing address will be collected - this is required by PSD2 regulation
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.
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 node is used 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.
Customer phone field	Due to PSD2 regulation, phone number is mandatory in payment request. Please set field name (for example field_phone). See section 10 for more information
Customer identifier	In case, when phone field is configured, then please set field name for example field_sales_id. When the field is empty, then the email address will be used. See section 10 for more information
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.
Disable rsa blinding	Define CRYPT_RSA_DISABLE_BLINDING as true in case of custom PHP build or PHP7 (experimental)
Conditions	Set condition(s) for payment, like minimum order amount.

Status	Set status for payment gateway

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 check payment status for pending orders. For information about configuring it, please check Drupal documentation:
https://www.drupal.org/docs/8/administering-a-drupal-8-site/cron-automated-tasks

5. Late/delayed shop payment success page URL

The late shop payment success page URL is: **https://<yourdomain>/commerce-verifone/successDelayed** or **http://<yourdomain>/commerce-verifone/successDelayed** 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: • 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

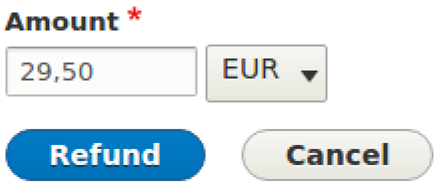
7. Order states and inventory handling

Order before payment has been accepted will be available in **Cart** tab on **Commerce > Orders** admin page.
For successfully paid orders, the order state will be set to **Completed**.
For cancelled orders, the order will be available in **Cart** view. 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 paid. The order is store as a Cart before paid (or payment cancellation).

8. Refund feature

Refund is available on order details page in tab **Payments** > **Refund** from the order list view.
To refund money, you need put the amount to be refunded into the field and click **Refund** button.
If necessary, it is possible to create multiple transactions, for example for each product



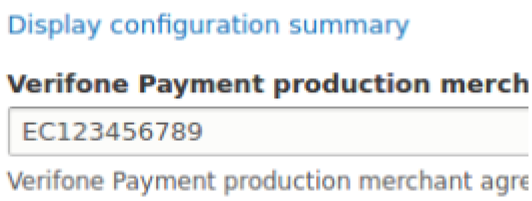
Amount *

29,50 EUR

9. Configuration summary

Below are available screenshots with example configuration summary for different states.

Configuration summary is available after click **Display configuration summary** link.



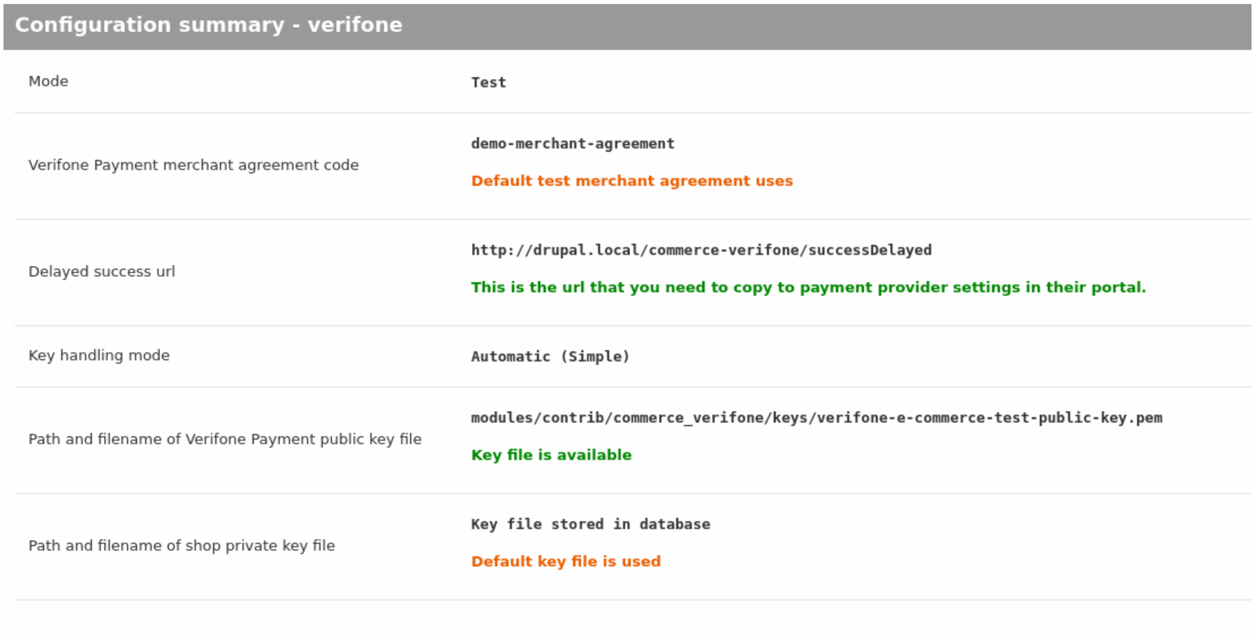
Display configuration summary

Verifone Payment production merch

EC123456789

Verifone Payment production merchant agre

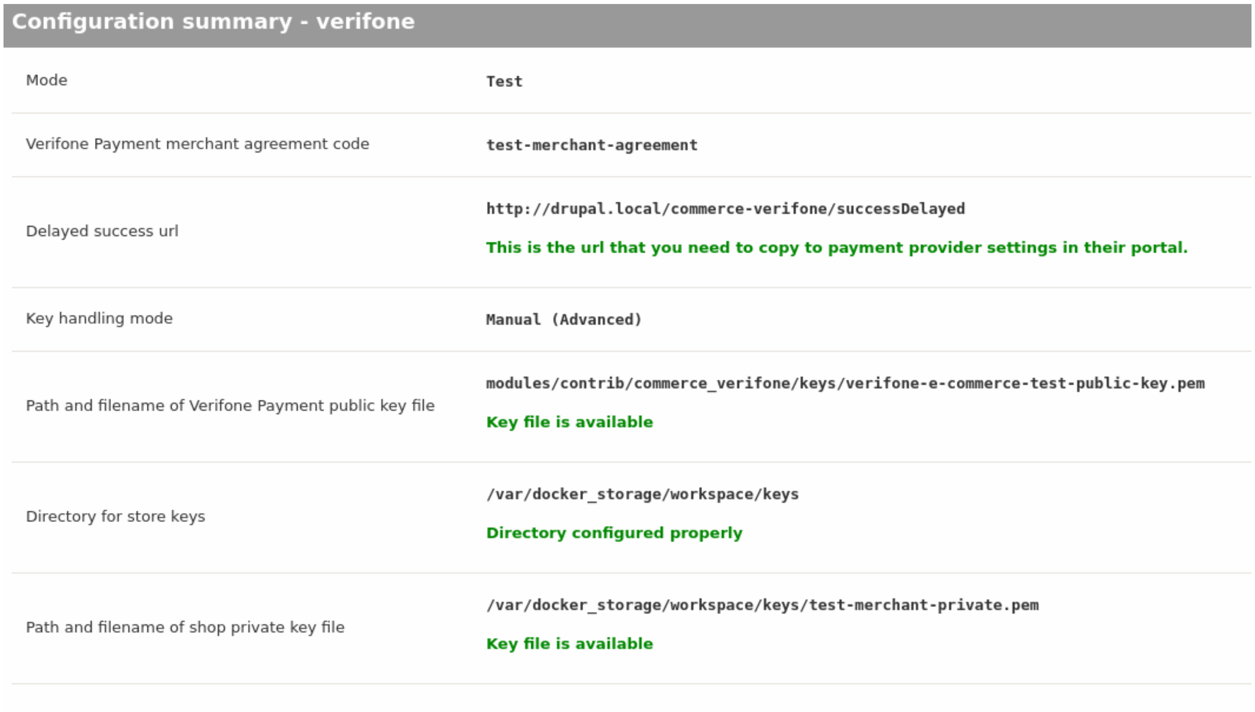
1. Default test configuration - use for testing with default keys.



Configuration summary - verifone

Mode	Test
Verifone Payment merchant agreement code	demo-merchant-agreement **Default test merchant agreement uses**
Delayed success url	http://drupal.local/commerce-verifone/successDelayed **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	modules/contrib/commerce_verifone/keys/verifone-e-commerce-test-public-key.pem **Key file is available**
Path and filename of shop private key file	Key file stored in database **Default key file is used**

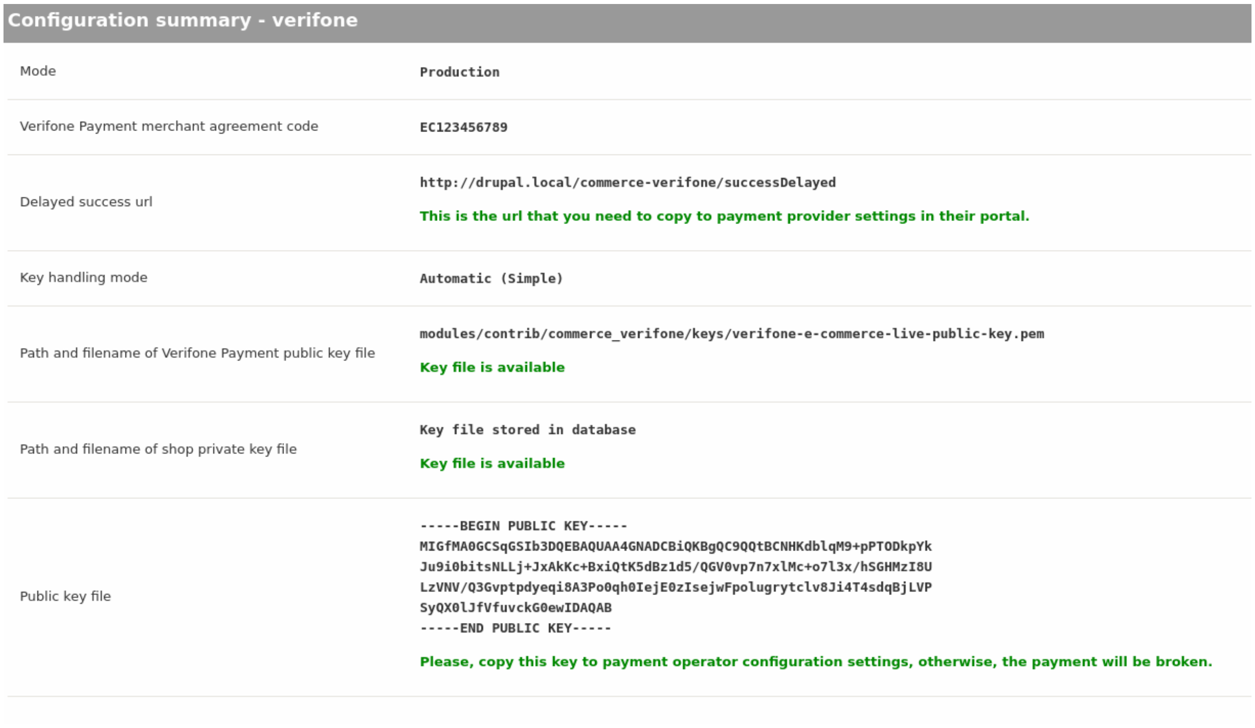
2. Custom test configuration - use for testing with custom merchant agreement code and keys



Configuration summary - verifone

Mode	Test
Verifone Payment merchant agreement code	test-merchant-agreement
Delayed success url	http://drupal.local/commerce-verifone/successDelayed 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	modules/contrib/commerce_verifone/keys/verifone-e-commerce-test-public-key.pem 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

3. Production configuration



Configuration summary - verifone

Mode	Production
Verifone Payment merchant agreement code	EC123456789
Delayed success url	http://drupal.local/commerce-verifone/successDelayed 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	modules/contrib/commerce_verifone/keys/verifone-e-commerce-live-public-key.pem Key file is available
Path and filename of shop private key file	Key file stored in database Key file is available
Public key file	-----BEGIN PUBLIC KEY----- MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC9QQtBCNHKdblqM9+pPTODkpYk Ju9i0bitsNLLj+JxAkKc+BxiQtK5dBz1d5/QGV0vp7n7xlMc+o7l3x/hSGHMzI8U LzVNV/Q3Gvptpdyeqi8A3Po0qh0IejE0zIsejwFpolugrytclv8Ji4T4sdqBjLVP SyQX0lJfVfuvckG0ewIDAQAB -----END PUBLIC KEY----- Please, copy this key to payment operator configuration settings, otherwise, the payment will be broken.

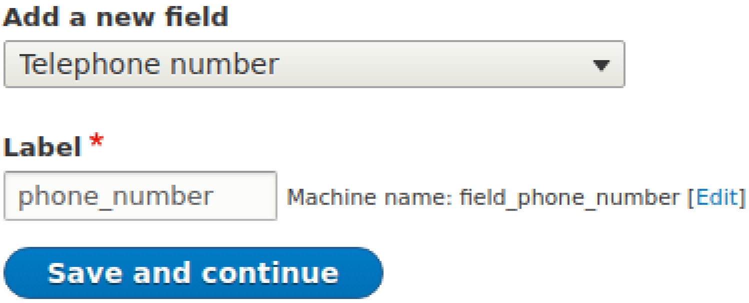
10. Configure customer identifier field

1. Go to the **Configuration > People > Profile types**
2. Click **Manage fields** next to the **Customer** profile type

3. Click **Add field**, select a **Telephone number** from the **Add a new field** list. Set label for the field. Store/Copy **Machine name** value (next to the label)
4. Click the **Save and continue** button.
5. On the next page, you can set **Allowed number of values**. If you are not sure what configure, leave the default value and click the **Save field setting** button
6. Go to the **Commerce > Configuration > Payment Gateways** and click **Edit** next to the created payment method.
7. Put **Machine name** value into **Configuration** field
8. Save configuration

Example configuration for the phone number field. Machine name for this configuration is:

field_phone_number



Add a new field
Telephone number

Label *
phone_number
Machine name: field_phone_number [Edit]