



Maksuturva/eMaksut
Payment Gateway Module
PrestaShop 1.4. -



Revision 121

Table of Contents

1 Changes.....	3
2 System requirements.....	4
3 Limitations and notes.....	4
4 Package contents.....	5
5 Installation.....	6
5.1 Package installation.....	6
5.2 PrestaShop setup.....	7
6 Configuration.....	8
6.1 Seller ID.....	8
6.2 Secret key.....	8
6.3 Secret key version.....	9
6.4 Communication encoding.....	9
6.5 eMaksut.....	9
6.6 Sandbox mode.....	9
6.7Communication URL.....	9
7 Language customization.....	10
8 Administration.....	11
8.1 Delayed payments (if available).....	11
9Testing the installation.....	12
9.1Sandbox testing.....	12
9.2Testing in Maksuturva test environment.....	13
10Paying through Maksuturva payment gateway.....	13

1 Changes

Date	Module revision	Description
16.04.12	101	Initial release.
13.07.12	108	Handling of multibyte characters fixed. Previous module versions may run into hash calculation problems with some characters or character encodings.
17.07.12	110	The module is now compatible beginning with Prestashop version 1.4.3
15.03.13	120	Fixes to handling product name and description (use other one if the other is blank). Also some changes in the common code. No effect on functionalities in this implementation.
06.08.14	121	<p>Bugfixes:</p> <ul style="list-style-type: none">• the module now supports Prestashop up to version 1.6.• the module now removes quotation marks from product names and description – previously it caused errors in hash calculation• product SKU is added when available, shipping method name used as shipping name when available• Optional parameters (e.g. pre-selected payment method, buyer's identification code) are now included in the hash calculation when available (currently not available, though, but enabling is possible in further development - see <code>_construct</code> -method in <code>MaksuturvaGatewayImplementation.php</code> and Maksuturva Payment interface description).• Module directory structure updated to enable installation through Prestashop user interface. <p>Documentation updated.</p>

2 System requirements

The Maksuturva Payment Gateway module for PrestaShop was tested on and requires the following set of applications in order to fully work:

Requirement	Supported versions
PrestaShop	<ul style="list-style-type: none">1.4.3. - 1.6.x
PHP	<ul style="list-style-type: none">4.x5.x
PHP cURL Module	<ul style="list-style-type: none">Any
PHP mbstring module	<ul style="list-style-type: none">n/a

There is no guarantee that the module is fully functional in any other environment which hasn't such requirements.

3 Limitations and notes

The following limitations have been found in this revision of the payment module or in the web store software in question:

- The order payment amount in Maksuturva response must match order payment amount sent from PrestaShop to Maksuturva. That means that no extra handling or billing fees can be added to payments in Maksuturva payment page. If such extra fees are needed, they need to be added to the order already in PrestaShop before the buyer is directed to Maksuturva service page. Currently the module does not support pre-selection of payment methods so adding extra fees depending on selected payment method is not supported either.

4 Package contents

This guide assumes that all the package files were correctly downloaded by the user. The uncompressed folder structure is presented in Illustration 1.

```
—maksuturva
  config.xml
  emaksut.png
  en.php
  index.php
  logo.gif
  logo.png
  maksuturva.gif
  maksuturva.php
  MaksuturvaGatewayAbstract.php
  MaksuturvaGatewayImplementation.php
  payment.php
  payment.tpl
  payment_execution.tpl
  payment_return.tpl
  validation.php
  Installation_Guide.odt
  Installation_Guide.pdf
  LICENSE.TXT
  README.md
  TESTING_INSTRUCTIONS.md
```

Illustration 1: Package content.

File	Description
maksuturva/validation.php	Callback url used to verify a payment (ok, error, cancel or giro payment).
maksuturva/MaksuturvaGatewayAbstract.php	Base class used to wrap Maksuturva's features.
maksuturva/MaksuturvaGatewayImplementation.php	Implementation of base maksuturva's class based on OsCommerce
maksuturva/maksuturva.php	The main payment gateway file, called from checkout and administration areas.
maksuturva/config.xml	Module general information.
maksuturva/index.php	Module access restriction.
maksuturva/payment.php	Payment preparation process, with form generation and presentation.
maksuturva/payment.tpl	Payment button template.
maksuturva/payment_execution.tpl	Payment form template (with hidden fields which comprise the products and store credentials).
maksuturva/payment_return.tpl	Success or error message templates for when

File	Description
	coming back from the payment in Maksuturva.
maksuturva/en.php	Empty language file.
maksuturva/emaksut.png maksuturva/logo.gif maksuturva/logo.png maksuturva/maksuturva.gif	Image files used by the module.

5 Installation

Prior to any change in your environment, it is strongly recommended that a backup of the entire existing PrestaShop installation is performed.

5.1 Package installation

The simplest way of installing module is to use PrestaShop administration user interface. Compress the maksuturva directory as an archived file maksuturva.zip (or in .tar, .tar.gz or .tgz format). From the main menu choose “Modules” and “Add a new module” (“Add a module from my computer” in PrestaShop 1.4). Upload file maksuturva.zip to PrestaShop.

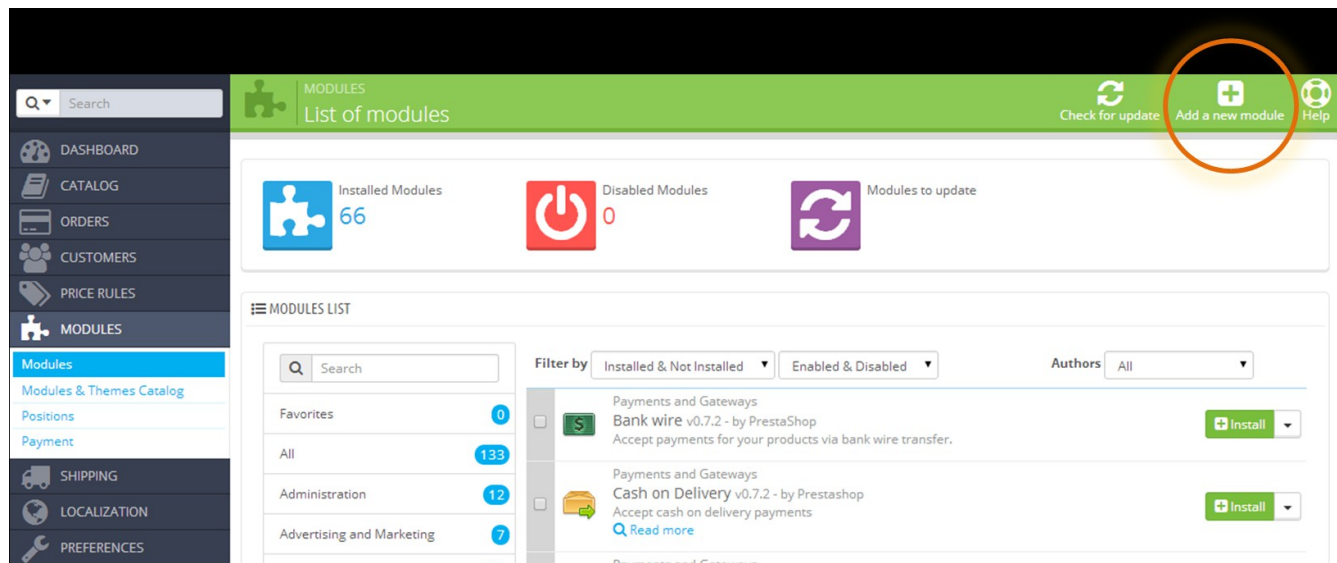


Illustration 2: Add a new module option in PrestaShop 1.6

Another option is to copy uncompressed maksuturva directory with its content to the modules directory in your PrestaShop installation (see Illustration 3).

```
—PrestaShop root
├── modules
│   └── maksuturva
│       ├── config.xml
│       ├── emaksut.png
│       ├── en.php
│       ├── index.php
│       ├── logo.gif
│       ├── logo.png
│       ├── maksuturva.gif
│       ├── maksuturva.php
│       ├── MaksuturvaGatewayAbstract.php
│       ├── MaksuturvaGatewayImplementation.php
│       ├── payment.php
│       ├── payment.tpl
│       ├── payment_execution.tpl
│       ├── payment_return.tpl
│       └── validation.php
```

Illustration 3: Module location in PrestaShop directory

5.2 PrestaShop setup

When the files are uploaded into PrestaShop or its modules directory, open the administrative panel of your PrestaShop installation and go to the payment modules section. Choose “install” as shown in Illustration 4. The module is now shown as installed, and the configuration can be performed.

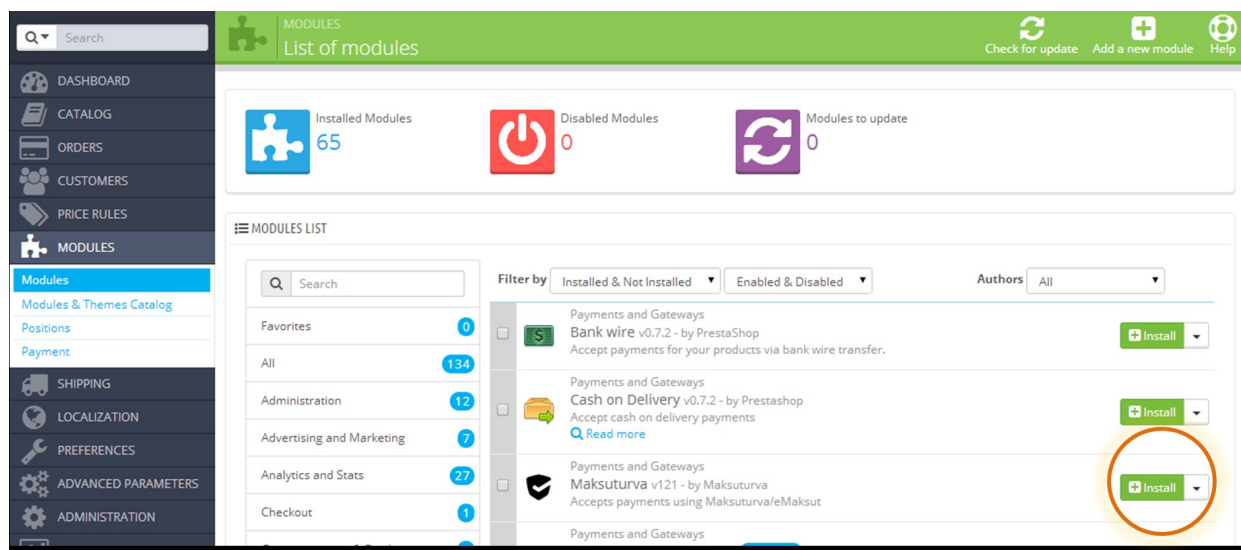


Illustration 4: Install the uploaded payment module in PrestaShop 1.6

6 Configuration

The configuration section for Maksuturva's module is reached by accessing the installed payment modules, as shown in the setup section. The parameters and their possible configurations are presented in Illustration 5. All the configuration entries are detailed below. Remember to save the changes after your actions.

It is strongly encouraged to test the module and payment service before using the module in production mode with actual money transfers. The settings varies depending on if the module is used in production mode, for testing in Maksuturva test environment or for sandbox testing. Read more about testing in chapter 9 Testing the installation.

Maksuturva/eMaksut configuration

Settings

Seller ID:

Secret Key:

Secret Key Version:

Communication Encoding: ☒ UTF-8 ☐ ISO-8859-1

eMaksut: ☐ Active ☒ Inactive

Sandbox mode (tests): ☒ Active ☐ Inactive

Communication URL:

Illustration 5: Maksuturva payment module configuration options

6.1 Seller ID

Open your Maksuturva account and check your seller ID in account details. The seller id is your identifier, used in every transaction inside your store. Copy the value and paste it in this field.

When used for testing in *Maksuturva test environment* use your test environment seller id.

When used in *sandbox mode*, general test account is used and the value filled in this option will be ignored.

6.2 Secret key

Together with the Seller ID, the secret key is also provided by Maksuturva. Check your secret key and input this value into the configuration field. Please note that this key must not be shared with any person, since it allows many operations to be done in your Maksuturva account.

When used for testing in *Maksuturva test environment* use your test environment secret key.

When used in *sandbox mode*, general test account is used and the value filled in this option will be ignored.

6.3 Secret key version

This parameter is also provided by Maksuturva. Check your secret key version and input this value into the configuration field. Please note that this key must not be shared with any person, since it allows many operations to be done in your Maksuturva account.

6.4 Communication encoding

There are two possible character encodings that are accepted by Maksuturva. Even if you are using a different encoding the module will attempt to convert the data from your PrestaShop encoding to the encoding set on this configuration. You might need to test which of them works better for your encoding by using the sandbox mode. The possible values are 'ISO-8859-1' and 'UTF-8' and they are selected from radio button options. The default value for this setting is 'UTF-8'.

Wrong encoding will fail the hash verification of the payment message sent to Maksuturva service.

6.5 eMaksut

This configuration option enables payments to be done using eMaksut's service, instead of using Maksuturva's one. Activate this option only if your payment service is an eMaksut service.

6.6 Sandbox mode

It is possible to set the module in sandbox test mode through activating the Sandbox mode. When this option is selected, default test parameters (*Seller ID* and *Secret key*) will be used in your store payments. Purchases made during sandbox will be directed to sandbox test page to verify information module is forwarding to Maksuturva as well as responses from Maksuturva service to PrestaShop.

When set to "Inactive", the production mode will be used. If *Maksuturva production service* credentials and communication URL parameters are used real commercial transactions will be performed when payments are done. Using *Maksuturva test credentials* and *test environment URL* makes possible to test the full payment process without actual money transactions.

Note that the default installation value for this configuration entry is "Active".

6.7 Communication URL

This configures which URL the payment and query status operations will send the data to Maksuturva. The default value '<https://www.maksuturva.fi>' refers to Maksuturva *production environment*.

The communication URL for Maksuturva *test environment* is '<http://test1.maksuturva.fi>' (note, that transfer protocol for test environment is *http*, not *https*).

7 Language customization

It is possible to customize the language and messages used in the module to the languages enabled in PrestaShop language settings. The customization takes effect by changing the text entries using the administrative tool, accessed by the modules configuration page (see Illustration 6) or through Localization menu (see Illustration 7) by selecting “Installed modules translations” from “Type of translation” menu. Look for the section “Module: maksuturva”, and apply all the changes for the selected language.

In PrestaShop 1.6, when translations are accessed through Localization menu it is also possible to make different translations for each theme (See “select your theme” -option in Illustration 7) that overrule core translations.

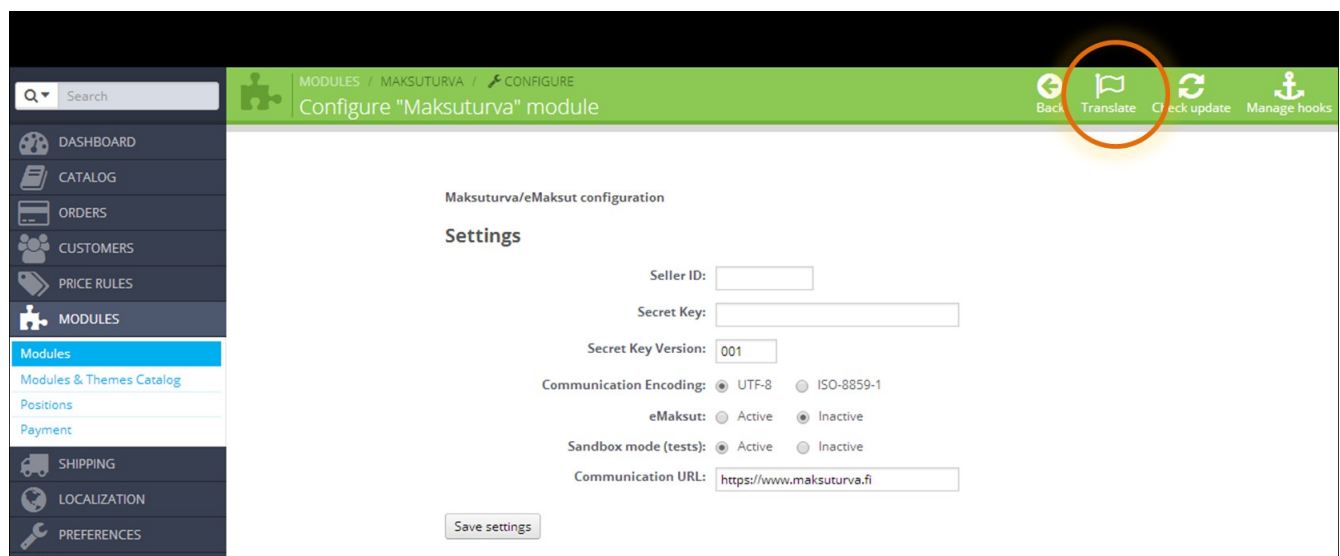


Illustration 6: Accessing module translations through module configuration page

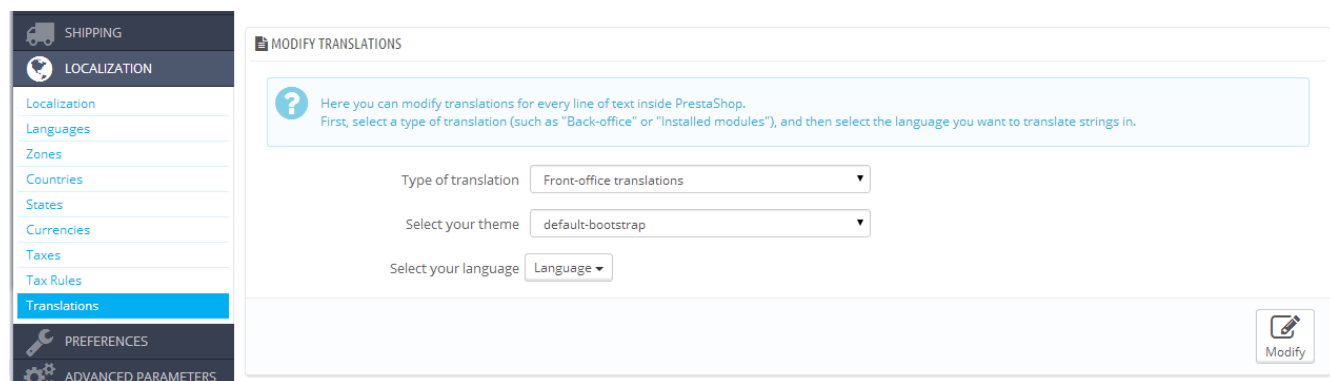
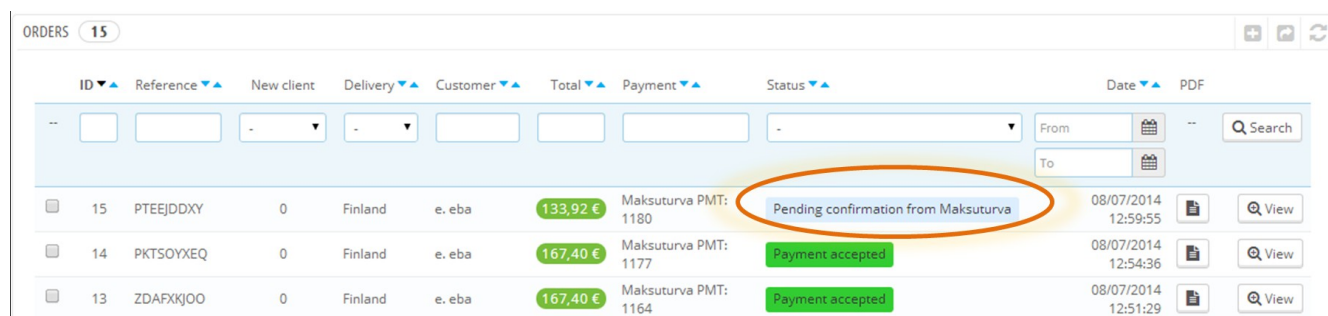


Illustration 7: Accessing module translations through Localization menu

8 Administration

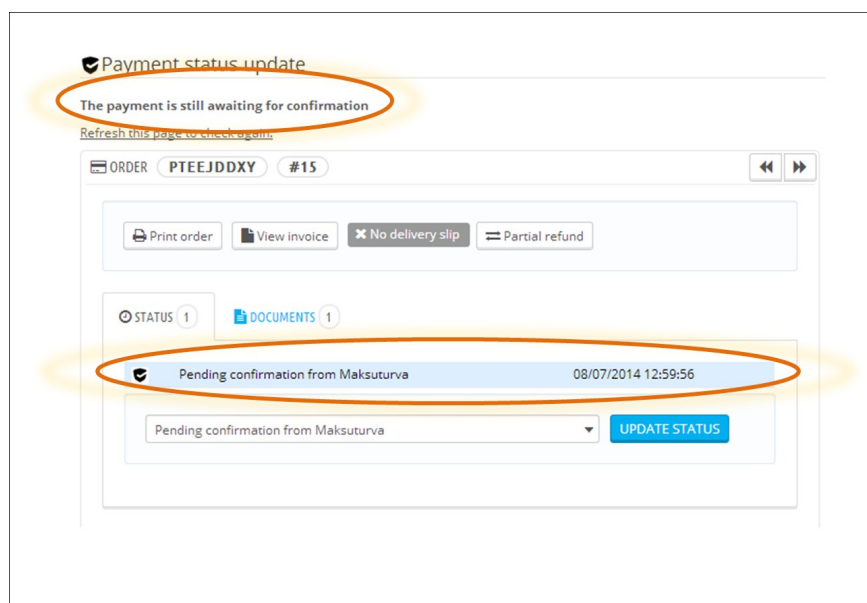
8.1 Delayed payments (if available)

When the buyer has paid the order by credit card, online bank payment, invoice, part payment etc. Maksuturva service will respond to PrestaShop that the payment has been *accepted*. The Maksuturva Gateway Payment module has also a support for delayed payments when the payment method is available (see Illustration 8). For that purpose, an order which payment has not yet been verified can be automatically verified by opening the order itself. When an order with an unverified payment is opened, the verification process will automatically start. If verification is received from Maksuturva payment status will change from unverified (see Illustration 9) to verified (Illustration 10).



ID	Reference	New client	Delivery	Customer	Total	Payment	Status	Date	PDF
15	PTEEJDDXY	0	Finland	e. eba	133,92 €	Maksuturva PMT: 1180	Pending confirmation from Maksuturva	08/07/2014 12:59:55	View
14	PKTSOYXEQ	0	Finland	e. eba	167,40 €	Maksuturva PMT: 1177	Payment accepted	08/07/2014 12:54:36	View
13	ZDAFXKJOO	0	Finland	e. eba	167,40 €	Maksuturva PMT: 1164	Payment accepted	08/07/2014 12:51:29	View

Illustration 8: Unconfirmed payment in PrestaShop Orders list



Payment status update

The payment is still awaiting for confirmation
[Refresh this page to check status](#)

ORDER: PTEEJDDXY #15

Print order View invoice No delivery slip Partial refund

STATUS 1 DOCUMENTS 1

Pending confirmation from Maksuturva 08/07/2014 12:59:56

Pending confirmation from Maksuturva UPDATE STATUS

Illustration 9: Order view with an unverified payment

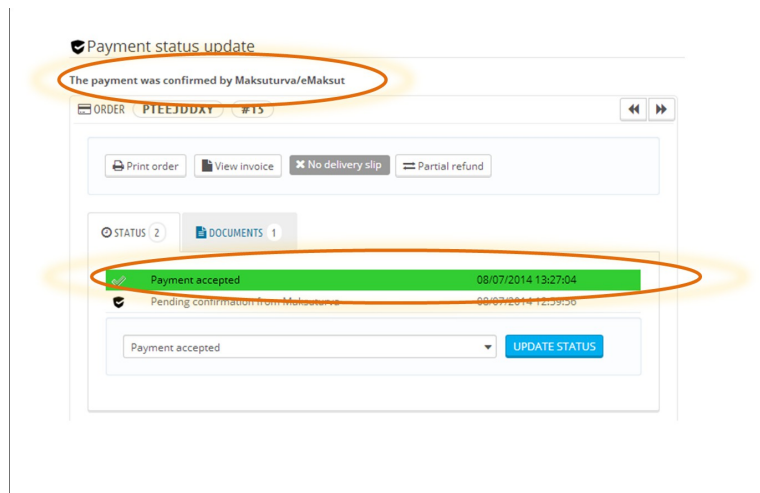


Illustration 10: Order view with a verified payment

9 Testing the installation

It is strongly encouraged that the Maksuturva payment module is thoroughly tested before taking it into use in on production web store. It is essential to verify that customer and product information, prices, taxes and discount as well as payment message verifying hash are transmitted correctly to Maksuturva.

9.1 Sandbox testing

First, one should validate payment messages by general a.k.a sandbox testing by activating Sandbox mode (see chapter 6.6 Sandbox mode). Activating the Sandbox mode, the module will use default test credentials and all confirmed orders are directed to a Sandbox test page where one can see the sent payment message data. Also all possible errors in data are presented.

The payment module calculates a checksum / hash from the payment data and webstore secret key combination. Secret key itself is never sent. When Maksuturva service receives the payment message it verifies the authenticity by doing the same calculation and comparing the two checksums.

If Sandbox test page reports errors in hash, the following checks are recommended:

- all the message data on the Sandbox test page are accurate and match the ones that have been sent. No extra spaces or symbols are added or removed and no symbol is changed to something else neither.
- Check that the selected encoding is accurate and change it if required (see chapter 6.4 Communication encoding).
- Check if product information (e.g. name, description) includes problematic special symbols (see chapter 3 Limitations and notes).
- If the product information is copied from word processing software (Office Word etc.) copied text might include markings (header, bulletpoint and emphasis markings etc.) which – even if they are not visible – cause errors in hash calculation. In these cases the product information data should be verified by copying it to more simple text editor (e.g. Notepad) which removes all possible invisible symbols and then copied back to PrestaShop replacing previous product

information content.

Sandbox test page also shows links to responses to webstore (see Illustration 11). By selecting appropriate link, one can verify the information that buyer and store administration see when payment is e.g. accepted or canceled.

[Ok paluu / OK return](#) Palaa välittäen ns request stringissä maksamisen vahvistusviestin osoitteeseen, joka tuli pmt_okreturn -kentässä.

[Virhe paluu / Error return](#) Palaa ilman lisätyjä parametreja osoitteeseen, joka tuli pmt_errorreturn -kentässä

[Peruutus paluu / Cancel return](#) Palaa ilman lisätyjä parametreja osoitteeseen, joka tuli pmt_cancelreturn -kentässä

[Viivästetty maksu \(tilisiirto\) paluu / Delayed payment \(giro\) return](#) Palaa ilman lisätyjä parametreja osoitteeseen, joka tuli pmt_delayedpayreturn -kentässä

Illustration 11: Return options in Maksuturva Sandbox test page

9.2 Testing in Maksuturva test environment

When Sandbox mode is disabled, module acts as in production mode and order confirmation directs buyer to Maksuturva payment page. If actual Maksuturva credentials are used real money transactions will occur.

If communication URL is set to refer to Maksuturva test environment it is possible to try the full online bank payment and invoice payment process without using actual money. Test environment uses different credentials than Maksuturva production environment and test environment service needs to be ordered separately from the actual Maksuturva service. More information about testing is available on Maksuturva web site www.maksuturva.fi.

10 Paying through Maksuturva payment gateway

Step 1 - Log in as a customer. Perform a purchase of items you choose and select Maksuturva as the payment option during the checkout (see Illustration 12).

3 PLEASE CHOOSE YOUR PAYMENT METHOD

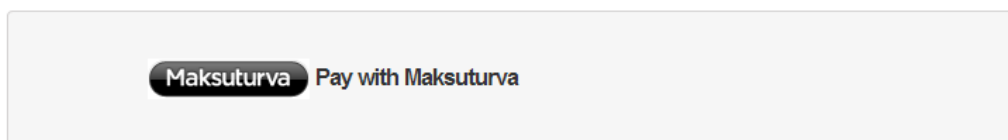
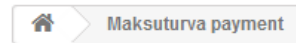


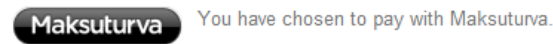
Illustration 12: Selecting Maksuturva as a payment method

Step 2 – Confirm the order



Order summary

Maksuturva payment



Here is a short summary of your order:

- The total amount of your order is 133,92 € (tax incl.)

You will be redirected to Maksuturva to perform the payment.

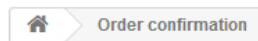
Please confirm your order by clicking 'I confirm my order'.



Illustration 13: Confirming the order

Step 3 – If using Sandbox mode verify payment by clicking “OK”. If using Maksuturva test or production environment, choose the payment method from the available options and complete the payment process.

Step 4 – Now the payment was confirmed. Buyer sees the order confirmation page (see Illustration 14) and store administration the new confirmed order (see e.g. illustrations 8 and 10 at page 12).



ORDER CONFIRMATION

Your order on SMT testikauppa is complete.



Your order will be shipped as soon as possible.

For any questions or for further information, please contact our customer support.

[← Back to orders](#)

Illustration 14: Order confirmation message to buyer after one has been redirected back to PrestaShop from Maksuturva service