



Maksuturva/eMaksut
Payment Gateway Module
Magento 1.5-1.9



Revision 125

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 Language customization.....	7
5.3 Magento setup.....	7
6 Configuration.....	8
6.1 Enable the module.....	9
6.2 Sandbox mode.....	9
6.3 Key Version.....	10
6.4 Communication encoding.....	10
6.5 Emaksut.....	10
6.6 Seller ID.....	10
6.7 Secret key.....	10
6.8 Communication URL.....	10
6.9 Allow Magento to send order confirmation emails.....	11
6.10 Changing payment id numbering format.....	11
7 Administration.....	13
7.1 Order status.....	13
7.2 delayed payments.....	13
7.3 Manually verifying payments.....	15
7.4 Payment amount changes when additional invoicing fees.....	16
8 Testing the installation.....	17
8.1 Sandbox testing.....	17
8.2 Testing in Maksuturva test environment.....	18
9 Paying through Maksuturva payment gateway.....	18

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 and character encodings.
13.11.12	108c	Fixed the handling of seller cost changes when eg. invoicing fees are added to the price. That is: <i>seller costs sent from webstore != seller costs returned from Maksuturva service.</i>
07.03.13	121	Fixes to handling of product name and description (use other one if the other is blank), hash calculation problems with multibyte characters and handling of other currencies than euro. Also now the module creates invoices to orders and marks them as paid. Error message is more informative now, in case of connection errors while verifying payments.
12.09.13	122	Bugfixes in payment verification and invoice saving. Previously invoices (even if registered as paid) did not affect the order's total amount.
24.10.13	123	<ul style="list-style-type: none"> - Products' article number (SKU) and short description are now passed on to Maksuturva service. - Prices are passed on as net prices + VAT (previously gross prices + VAT) for minimizing rounding errors. - Added support for Magento configurable products. Configurable product rows now sent as in a single row (previously in two rows: parent and child row). - Added minimal support for Magento bundled product. All bundled products are set as "tailored product" which, in general, buyers are not allowed to return. - Updated documentation with instructions for changing the used order ID type.
05.03.14	124	<ul style="list-style-type: none"> - Added support for allowing Magento to send order confirmation emails to buyer and seller after a succesful payment. See section 6.9 for details. - Payment ID numbering was changed to use long numbering format. <p>NOTE: after upgrading the module to revision 124 from earlier versions, the payment status verification does not work for orders that are made with previous module revision. See section 6.10 for details.</p>
08.10.14	125	Bugfix: 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).

2 System requirements

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

Requirement	Supported versions
Magento	<ul style="list-style-type: none">1.5.x, 1.6.x, 1.7.x, 1.8.1.x, 1.9.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:

- No reported flaws or notable limitations

4 Package contents

This guide assumes that all the package files were correctly downloaded by the user. The following folder structure is presented by the module when uncompressed:

```

app/
├── code
│   ├── local
│   │   ├── Maksuturva
│   │   │   ├── Makadmin
│   │   │   │   ├── controllers
│   │   │   │   │   └── BackendController.php
│   │   │   │   └── etc
│   │   │   │       └── config.xml
│   │   │   └── Maksuturva
│   │   │       ├── Block
│   │   │       │   ├── Adminhtml
│   │   │       │   │   └── Sales
│   │   │       │   │       └── Order
│   │   │       │   │           └── View.php
│   │   │       │   └── Form.php
│   │   │       │   └── Redirect.php
│   │   │       ├── controllers
│   │   │       │   └── IndexController.php
│   │   │       ├── etc
│   │   │       │   ├── config.xml
│   │   │       │   └── system.xml
│   │   │       ├── Model
│   │   │       │   ├── MaksuturvaGatewayAbstract.php
│   │   │       │   ├── MaksuturvaGatewayImplementation.php
│   │   │       │   ├── PaymentMethod.php
│   │   │       │   └── Source
│   │   │       │       └── Encoding.php
│   │   │       └── sql
│   │   │           ├── maksuturva_setup
│   │   │           └── mysql4-install-0.1.0.php
│   └── etc
│       ├── modules
│       │   ├── Maksuturva_Makadmin.xml
│       │   └── Maksuturva_Maksuturva.xml
│       └── locale
│           ├── en_US
│           │   └── Maksuturva_Maksuturva.csv
│           └── fi_FI
│               └── Maksuturva_Maksuturva.csv

```

Illustration 1: Tree structure of package contents

File	Description
app/code/local/Maksuturva/Makadmin/controllers/BackendController.php	Contain the action used to verify payments from the admin interface.
app/code/local/Maksuturva/Makadmin/etc/config.xml	Configure the payment verification module
app/code/local/Maksuturva/Maksuturva/Block/Adminhtml/Sales/Order/View.php	Add the 'Verify payment' button to orders on admin interface.
app/code/local/Maksuturva/Maksuturva/Block/Form.php	Print the module title on checkout page.

File	Description
app/code/local/Maksuturva/Maksuturva/Block/Redirect.php	Print the redirect page with data to be sent to Maksuturva gateway.
app/code/local/Maksuturva/Maksuturva/controllers/IndexController.php	Controller with actions to redirect the user to Maksuturva and receive responses.
app/code/local/Maksuturva/Maksuturva/etc/config.xml	Module configurations, routes and setup default values
app/code/local/Maksuturva/Maksuturva/etc/system.xml	Configuration file that contains admin interface setup fields.
app/code/local/Maksuturva/Maksuturva/Model/MaksuturvaGatewayAbstract.php	Base class used to wrap Maksuturva's features.
app/code/local/Maksuturva/Maksuturva/Model/MaksuturvaGatewayImplementation.php	Implementation of base maksuturva's class
app/code/local/Maksuturva/Maksuturva/Model/PaymentMethod.php	Main class of Maksuturva payment method.
app/code/local/Maksuturva/Maksuturva/Model/Source/Encoding.php	Source that enable selection between UTF-8 and ISO-8859-1 on the admin interface.
app/code/local/Maksuturva/Maksuturva/sql/mysql4-install-0.1.0.php	Database installation file, currently not used.
app/etc/modules/Maksuturva_Makadmin.xml	Configuration file to load the administrative portion of Maksuturva payment module (Status query)
app/etc/modules/Maksuturva_Maksuturva.xml	Configuration file to load the Maksuturva payment module
app/locale/en_US/Maksuturva_Maksuturva.xml	Translation file for US english
app/locale/fi_FI/Maksuturva_Maksuturva.xml	Translation file for Finnish

5 Installation

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

5.1 Package installation

The first step of installation involves to copy the package structure with its contents into a valid Magento installation directory. The simplest way of performing such operation is through copying the catalog folder from the maksuturva package into the installation Maksuturva directory. It is important to notice that a *merge* operation needs to be done, instead of replacing the entire directory.

Another option is to copy the package contents individually into their destination directories, creating such directories when they do not exist.

5.2 Language customization

It is possible to customize the language and messages used in the module. The customization takes effect by changing the text entries located in the files

app/locale/en_US/Maksuturva_Maksuturva.csv or
app/locale/fi_FI/Maksuturva_Maksuturva.csv

```
app/locale/  
├─ en_US  
│   └─ Maksuturva_Maksuturva.csv  
└─ fi_FI  
    └─ Maksuturva_Maksuturva.csv
```

To add support to an existing installed language, open the destination language directory (**app/locale/**), then create the local folder **es_ES**, where spanish is taken as example, copy the **Maksuturva_Maksuturva.csv** from the english locale (**app/locale/en_US/Maksuturva_Maksuturva.csv**) to the new folder. Finally, open the **Maksuturva_Maksuturva.csv** file in edit mode and perform the translations and customizations.

5.3 Magento setup

When the files are copied into the destination directory, open the administrative panel of your Magento installation and go to the System menu and then access the Configuration option. The Illustration 2 shows the path to open the configuration page.

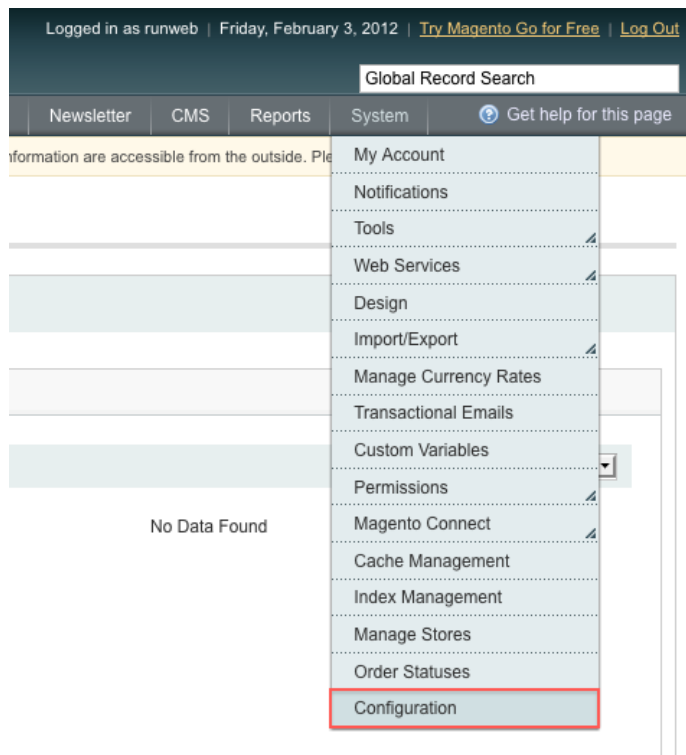


Illustration 2: Open the configuration page

On the configuration page, select the Advanced option from the bottom of the menu on the left, as shown in Illustration 3.

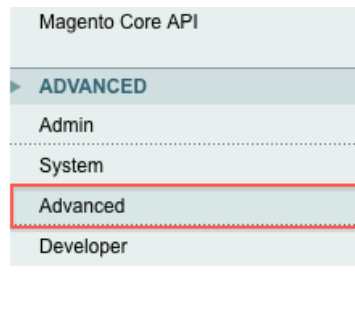


Illustration 3: Advanced menu

On the advanced page, under “Disable Modules Output” you should see two modules as shown in Illustration 4.

Advanced	Mage_Tag	Enable	Save Config
	Mage_Tax	Enable	
	Mage_Usa	Enable	
	Mage_Weee	Enable	
	Mage_Widget	Enable	
	Mage_Wishlist	Enable	
	Mage_XmlConnect	Enable	
	Maksuturva_Makadmin	Enable	
	Maksuturva_Maksuturva	Enable	
	Phoenix_Moneybookers	Enable	
	translations	Enable	

Illustration 4: Modules listing

If the modules are not displayed, click the “Save Config” button and it should update the list, displaying the modules.

6 Configuration

The configuration section for Maksuturva's module is reached by accessing the **System > Configuration** menu and then select “**Payment methods**” item from the left menu, as shown in Illustration 5. The parameters and their possible configurations are presented in Illustration 6.

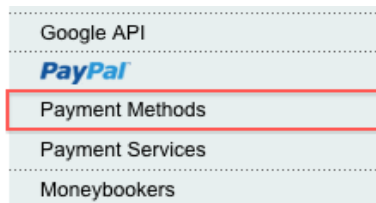


Illustration 5: Payment methods menu

Illustration 6: Maksuturva payment method configuration options

All the configuration entries are detailed below. Remember to save the changes after your actions.

6.1 Enable the module

The first configuration entry defines if the module should be activated for usage in payments for customers. When set to “Yes” it will be used. When set to “No”, it will not.

6.2 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 Magento.

When set to “No”, 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 “Yes”.

6.3 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 Magento 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 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.

6.7 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.

6.8 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*).

6.9 Allow Magento to send order confirmation emails

By default Magento does not send order confirmation emails to customers when an external payment service outside of Magento is used. Since Maksuturva payment module revision 124 Magento order confirmation emails are sent if sales emails are enabled. The feature of sending confirmation emails from Magento is independent on possible confirmation emails sent from Maksuturva- / eMaksuturva service.

In Magento sending order confirmation emails can be enabled and disabled in: *System* → *Configuration* → *Sales* → *Sales Emails* → *Order* (see *Illustration 7*). Email templates can be modified in *System* → *Transactional Emails*.

Illustration 7: Enabling and disabling Magento's order confirmation emails

6.10 Changing payment id numbering format

Maksuturva service requires unique payment ids for every payment. Order numbering can use short (e.g. 227) or long (eg. 100000227) format for Maksuturva payment id. **The short order numbering was used before but was changed to long format since Maksuturva payment gateway module revision 124 as this is more common in Magento.**

However, if one wishes module to continue using short numbering format , it can be done by altering the code in

app/code/local/Maksuturva/Maksuturva/Model/MaksuturvaGatewayImplementation.php

In rows 180 – 182 change the highlighted text with the following new values:

current numbering (long format):

```
"pmt_id"          => $order->getIncrementId(),  
"pmt_orderid"     => $order->getIncrementId(),  
"pmt_reference"   => ($order->getIncrementId() + 100),
```

previous numbering (short format):

```
"pmt_id"          => $order->getEntityId(),  
"pmt_orderid"     => $order->getEntityId(),  
"pmt_reference"   => ($order->getEntityId() + 100),
```

One should keep in mind, that after the change in the numbering format, the payment verification function (see 7.2 delayed payments) can not verify payments that are made with the previous settings.

7 Administration

7.1 Order status

When a customer has confirmed order and is transferred to Maksuturva payment service, Magento sets the order as "pending".

When the customer returns to Magento and Maksuturva confirms the payment, an invoice is created and set as paid and the order's amounts are updated. The order status is now set as "processing".

Magento updates order status as "complete" when the order is fully paid and shipped.

7.2 delayed payments

The Maksuturva Gateway Payment module allows customers to perform delayed payments, meaning that a constant verification of payment status is needed. For that purpose, there is a button "Verify Order" on the view page of each order. The Illustration 8 show how to access the order list. Then you must click the "view" link on the order you wish to verify, as shown on Illustration 8.

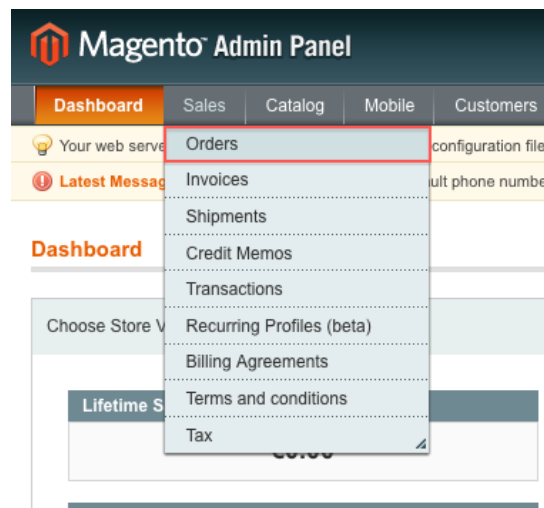


Illustration 8: Access order list

Page 1 of 1 pages | View 20 per page | Total 1 records found | [New Order RSS](#) Export to: CSV Export Reset Filter Search

Select Visible | Unselect Visible | 0 items selected Actions Submit

	Order #	Purchased From (Store)	Purchased On	Bill to Name	Ship to Name	G.T. (Base)	G.T. (Purchased)	Status	Action
Any			From: To :			From: To :	From: To :		
<input type="checkbox"/>	100000001	Main Website Main Store English	Feb 3, 2012 10:50:26 AM	Filipe Grillo	Filipe Grillo	€134.99	€134.99	Processing	View

Illustration 9: Magento order list

Magento Admin Panel Global Record Search Logged in as runweb | Friday, February 3, 2012 | [Try Magento Go for Free](#) | [Log Out](#)

Dashboard **Sales** Catalog Mobile Customers Promotions Newsletter CMS Reports System [Get help for this page](#)

Your web server is configured incorrectly. As a result, configuration files with sensitive information are accessible from the outside. Please contact your hosting provider.

Latest Message: Reminder: Change Magento's default phone numbers and callouts before site launch You have 1 notice unread message(s). [Go to notifications](#)

Order View

Order # 100000001 | Feb 3, 2012 10:50:26 AM

[Back](#) [Edit](#) [Cancel](#) [Send Email](#) [Hold](#) [Invoice](#) [Ship](#) [Reorder](#) [Verify Payment](#)

Information

- Invoices
- Credit Memos
- Shipments
- Comments History
- Transactions

Order # 100000001 (the order confirmation email is not sent)

Order Date	Feb 3, 2012 10:50:26 AM
Order Status	Processing
Purchased From	Main Website Main Store English
Placed from IP	192.168.56.1

Account Information

Customer Name	Filipe Grillo
Email	filipe.grillo@gmail.com
Customer Group	General

Illustration 10: Order view page with order details and verify payment button

When the “Verify Payment” is clicked, the system will request a status update from Maksuturva gateway. If the payment is confirmed and invoice is created and the order total amounts updated. A message is also displayed on top of the page and in in the “Comments History” box (illustrations 11 and 12).

Payment identified by Maksuturva

Order # 100000002 | Feb 3, 2012 11:04:31 AM

[Back](#) [Edit](#) [Cancel](#) [Send Email](#) [Hold](#) [Invoice](#) [Ship](#) [Reorder](#) [Verify Payment](#)

Illustration 11: Pending payment verification result

Comments History

Add Order Comments

Status

Processing ▾

Comment

☐ Notify Customer by Email ☒ Submit Comment

☐ Visible on Frontend

☐ Feb 6, 2012 2:24:33 AM | Processing

Customer Notification Not Applicable

Payment identified by Maksuturva

☐ Feb 3, 2012 11:04:37 AM | Pending Payment

Customer Notification Not Applicable

Waiting for delayed payment confirmation from Maksuturva

☐ Feb 3, 2012 11:04:32 AM | Pending Payment

Customer Not Notified

Illustration 12: Order comment history

7.3 Manually verifying payments

Besides the automatic pending status verification, another possibility is to open your Maksuturva account from within the Maksuturva module configuration panel. The link “Open KauppiasExtranet to view payments” is highlighted in Illustration 13 .

Illustration 13: Link to view payments in KauppiasExtranet

7.4 Payment amount changes when additional invoicing fees

If the buyer chooses to use a payment method with additional costs (eg. invoicing fees), Maksuturva Payment Service adds them to the total price for the buyer to pay instantly or through invoice.

In the web store the seller costs (shipping and invoicing costs) will not be updated, but store admin can see the price change in the order's comment history following the Payment confirmed -message (see Illustration 14).

Illustration 14: Order's comment history can be found at the bottom of the order view page. In this case the additional costs have been 2,00 € and the actual price the buyer has paid is 2,00 € more than what Magento says as total amount.

8 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.

8.1 Sandbox testing

First, one should validate payment messages by general a.k.a sandbox testing by activating Sandbox mode (see chapter 6.2 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 15). 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 lisättyjä parametreja osoitteeseen, joka tuli pmt_errorreturn -kentässä

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

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

Illustration 15: Return options in Maksuturva Sandbox test page

8.2 Testing in Maksuturva test environment

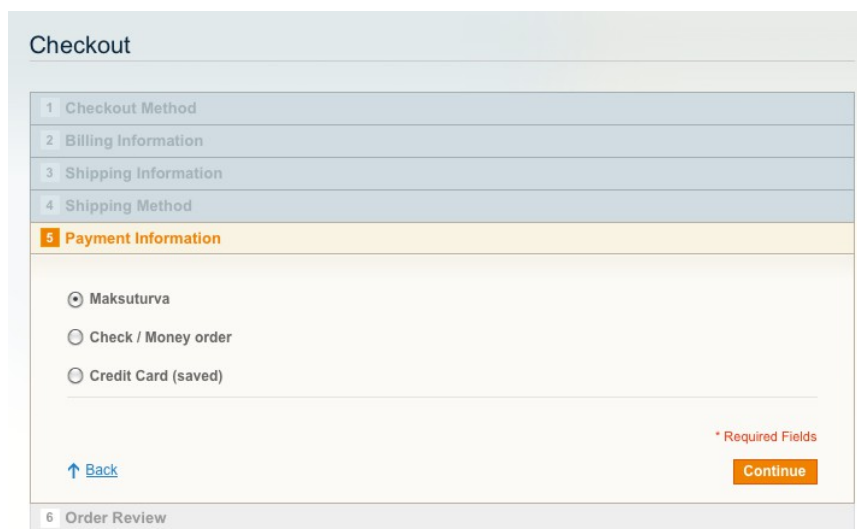
When Sandbox mode is disabled, module acts as in production mode and order confirmation directs buyer to Maksuturva payment page. If actual Maksuturva credentials are used real money transactions will occur.

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.

In order to test the Maksuturva Payment Gateway module, open your configuration section, as shown in section 6 Configuration), set the Sandbox mode to “Yes” and save your changes. Follow the steps below to test a purchase.

9 Paying through Maksuturva payment gateway

Step 1 - Log in as a customer and perform a purchase of a random item, selecting Maksuturva as the payment option during the checkout (see Illustration 16).



The screenshot shows a checkout page titled "Checkout" with a list of steps: 1 Checkout Method, 2 Billing Information, 3 Shipping Information, 4 Shipping Method, 5 Payment Information (highlighted in orange), and 6 Order Review. Under the "Payment Information" step, there are three radio button options: "Maksuturva" (selected), "Check / Money order", and "Credit Card (saved)". At the bottom left is a blue "Back" link with an upward arrow, and at the bottom right is an orange "Continue" button. A red asterisk note "* Required Fields" is also visible.

Illustration 16: Step 1 - Select maksuturva

Step 2 – Confirm the order.

Checkout

- 1 Checkout Method
- 2 Billing Information
- 3 Shipping Information
- 4 Shipping Method
- 5 Payment Information
- 6 Order Review**

Product Name	Price	Qty	Subtotal
Chair	€129.99	1	€129.99
Subtotal			€129.99
Shipping & Handling (Flat Rate - Fixed)			€5.00
Grand Total			€134.99

Forgot an Item? [Edit Your Cart](#)

[Place Order](#)

Illustration 17: Step 2 - Confirm the order

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

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

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

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

Illustration 18: Step 3 - Select simulation action

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 19) and store administration the new confirmed order.

Your order has been received

Thank you for your purchase!

Your order # is: [100000001](#).

You will receive an order confirmation email with details of your order and a link to track its progress.

Click [here to print](#) a copy of your order confirmation.

[Continue Shopping](#)

Illustration 19: Step 4 - Simulation result