



Maksuturva
Payment Gateway Plugin for
WooCommerce 2.2 -



version 2.0.5

Table of Contents

1	Changes	3
2	Service requirements	4
3	System requirements	5
4	Package contents	6
5	Installation.....	7
5.1	Upgrading an existing plugin.....	7
5.2	Installing the plugin	7
6	Configuration	9
6.1	Enable / Disable.....	9
6.2	Title	9
6.3	Customer message.....	9
6.4	Seller id.....	9
6.5	Secret Key and Secret key version	9
6.6	Gateway URL	9
6.7	Maksuturva Payment Prefix.....	10
6.8	Sandbox mode.....	10
6.9	Maksuturva encoding.....	10
7	Language customization	11
8	Testing the installation.....	12
8.1	Testing in Maksuturva test environment.....	12
9	Paying through Maksuturva Payment Gateway	13
10	Administration	15
10.1	Payment verification.....	15

1 Changes

date	version	description
03/06/2015	1.0.0	Initial release.
03/11/2016	2.0.0	<p>Changed:</p> <ul style="list-style-type: none"> - Complete re-factoring of the plugin source code to follow WordPress and WooCommerce standards and best practices. <p>Fixed:</p> <ul style="list-style-type: none"> - Issue #1: on_uninstall, do_background_checks is too generic global functions. The plugin now uses classes and the function names have been renamed. <p>Removed:</p> <ul style="list-style-type: none"> - Removed the eMaksut option from plugin administration, and always send "Y" as value for the "escrow" field. Maksuturva handles this mode internally, and the setting in the plugin has no effect. <p>Added:</p> <ul style="list-style-type: none"> - The plugin now creates a separate row for additional fees (e.g. invoicing fee) added to orders in Maksuturva service. The fee is shown as an extra row in the order both on the receipt and the administration interface.
03/17/2016	2.0.1	<p>Fixed:</p> <ul style="list-style-type: none"> - Fix JavaScript redirect to payment gateway on receipt page - Fix payment table update on plugin re-activation - Fix issue with paying an already pending order
04/28/2016	2.0.2	<p>Fixed:</p> <ul style="list-style-type: none"> - Fix order status after successful payment status query to Maksuturva <p>Changed:</p> <ul style="list-style-type: none"> - Improve order processing to allow for double-submissions from gateway <p>Added:</p> <ul style="list-style-type: none"> - Add cron script to check for payment status periodically
04/28/2016	2.0.3	<p>Fixed:</p> <ul style="list-style-type: none"> - Fix db update between plugin versions
07/04/2017	2.0.4	<p>Fixed:</p> <ul style="list-style-type: none"> - Fix order total discount issue, by forcing the order total discount value to be float. - Fix issue when no fee payment rows are added to the order. Re-calculate totals to match fees.
07/04/2017	2.0.5	<p>Fixed:</p> <ul style="list-style-type: none"> - Fix issue when payment status query log table size grows too large. <p>NOTE: The status query log table will be TRUNCATED for versions before 2.0.5, so if you want to keep the history, please make sure you back up the database table.</p>

Table 1: Changes

2 Service requirements

The plugin requires an activated Maksuturva payment service to work. You can order a Maksuturva account at <https://www.maksuturva.fi/tilaapalvelu>. For testing purposes you can order test credentials to Maksuturva's test environment at <http://test1.maksuturva.fi/tilaapalvelu>.

Without an active Maksuturva (test environment or production environment) account you can test the plugin using the Sandbox test mode. See more about testing in section 12.

3 System requirements

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

Requirement	Supported versions
WooCommerce	<ul style="list-style-type: none">• 2.2.10, 2.3.x
Wordpress	<ul style="list-style-type: none">• 4.0
PHP	<ul style="list-style-type: none">• 4.x• 5.x
PHP cURL Module	<ul style="list-style-type: none">• Any
PHP mbstring module	<ul style="list-style-type: none">• Any

Table 2: System Requirements

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

4 Package contents

This guide assumes that all the package files were correctly downloaded by the user. The folder structure is presented in Table 3.

File	Description
docs/*	Plugin documentation
includes/*	Class files included by the main plugin class to separate business logic into logical containers
languages/*	Plugin translations
templates/*	Template files used by the module both in the back-end and front-end
LICENSE	Plugin license file
maksuturva_logo.png	Maksuturva logo used by WooCommerce
README.md	Readme file
readme.txt	Plugin readme file required by WordPress
wc-maksuturva.php	Main plugin class

Table 3: Plugin package folder structure

5 Installation

Prior to any change in your environment, it is strongly recommended that you backup the entire existing WordPress and WooCommerce installation.

5.1 Upgrading an existing plugin

Upgrade the plugin by first deactivating and uninstalling the previous version of the plugin. Then continue by installing the new version by the following installation instructions.

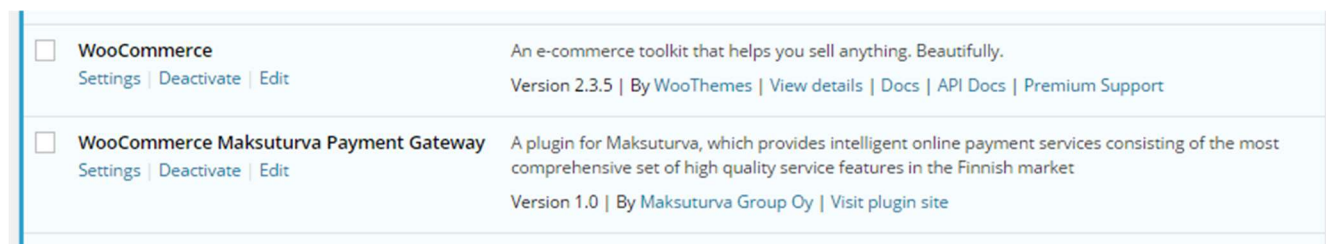


Illustration 1: Maksuturva WooCommerce Payment Gateway plugin installed in WordPress plugins page

5.2 Installing the plugin

Save the plugin folder as a zip-archive. In WordPress plugins page select *Add New* and then *Upload Plugin*. Choose the plugin zip-archive file from your computer and select *Install Now*. After installation *activate the plugin* (see *Illustration 2*).

GeneralProductsTaxCheckoutShippingAccountsEmailsAPI

[Checkout Options](#) | [BACS](#) | [Cheque](#) | [Cash on Delivery](#) | [PayPal](#) | **Maksuturva**

Maksuturva

Take payments via Maksuturva.

Enable/Disable

☒ Enable Maksuturva Payment Gateway

Title

?

Customer Message

?

Pay via Maksuturva.

Account settings

Seller id

?

Secret Key

?

Secret Key Version

?

Advanced settings

Gateway URL

?

Payment Prefix

?

Sandbox mode

☐ Sandbox mode

Maksuturva sandbox can be used to test payments. None of the payments will be real.

Maksuturva encoding

?

☒ UTF-8☐ ISO-8859-1

Save changes

Maksuturva Group Ltd Ruoholahdenkatu 23 | FIN-00180 Helsinki, Finland | info@maksuturva.fi

6 Configuration

Before you're able to use Maksuturva Payment Gateway plugin in your WooCommerce store, you need to do some configurations in the plugin's settings page (see Illustration 2). You can access the settings through the plugin in WordPress plugins list (as seen in Illustration 1) or by selecting Maksuturva in WooCommerce settings page's Checkout tab. Remember to *Save changes* afterwards.

6.1 Enable / Disable

Checking this check-box activates Maksuturva as a payment option for your customers in checkout process.

6.2 Title

Title for the payment gateway, shown on the checkout page. Defaults to Maksuturva.

6.3 Customer message

Description of the payment gateway. This message is shown below the payment gateway when chosen. Defaults to Pay via Maksuturva.

6.4 Seller id

With a Maksuturva service you'll get a unique seller id. This id is ignored when using the Sandbox mode.

6.5 Secret Key and Secret key version

Maksuturva services use your seller id and secret key combination to authenticate your web-store. Secret key is used to sign all payments coming from your store to Maksuturva so it is important to keep the secret key secret. Never send the secret key (or a screenshot picture of it) in an unencrypted email or place it somewhere where an outsider could have an access to it.

Your initial secret key's version number is *001*. If your secret key has been compromised you can order a new secret key from Maksuturva. In that case you need to update the version number to refer to the new secret key you have taken in to use. Your Seller id and Secret Key are ignored when using the Sandbox mode.

6.6 Gateway URL

Maksuturva production environment (where the customers can make an actual payments) uses an SSL-secured connection in a URL <https://www.maksuturva.fi>.

If you are testing the service using Maksuturva's test environment (where you can walk through the whole payment process without using actual money), the insecure connection uses URL <http://test1.maksuturva.fi>. Remember to change the URL to refer back to production mode before you open your store to actual customers.

Both service environments are separate and have their own service accounts so you have to order the credentials separately.

6.7 Maksuturva Payment Prefix

Maksuturva service requires unique ids for each payments. Orders with payment id that has already been used are not accepted.

As the payment id is calculated from the order numbers, if you move your Maksuturva account to a new ecommerce software installation and order numbering starts from beginning a conflict in payment

id numbers can occur. Adding a unique prefix for payment ids after re-installation of the web store you can prevent the collision of the payment id numbers. By default the prefix is empty.

6.8 Sandbox mode

When the Sandbox mode is activated the plugin uses Maksuturva's general test credentials which directs all payment attempts to Maksuturva's Sandbox test page. On the test page one can validate the order data e.g. money amounts, product descriptions etc. In the Sandbox test page one can also test how the orders will look to the customer or in the admin page when one has returned from Maksuturva with a successful payment, error or by canceling the payment.

As the WooCommerce won't know the difference between payments made in sandbox mode or in Maksuturva's test or production environment, remember to deactivate sandbox-mode after you've opened the store for live customers. If you are uncertain whether an order has an actual payment or has it been made in a test environment you can check all the payments in Maksuturva's Kauppias Extranet -services.

6.9 Maksuturva encoding

There are two possible character encodings that are accepted by Maksuturva. Even if you are using a different encoding, the plugin will attempt to convert the data from your WordPress 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 might cause the hash verification to fail in the Maksuturva service. If you get errors in the Maksuturva service page informing that the hash value is invalid, you should check if changing the encoding value fixes the problem.

7 Language customization

Payment Gateway Module can be translated to different languages and individual texts can be altered by editing .po -files in the languages directory and compiling them as corresponding .mo -files. One common tool for this in Windows- and Mac-environments is Poedit (poedit.net). Remember to add the new and altered files to the plugin zip-archive and reinstall the plugin to WordPress (See section 7).

8 Testing the installation

It is strongly encouraged that the Maksuturva Payment Gateway plugin 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 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 Gateway URL is set to refer to Maksuturva test environment (<http://test1.maksuturva.fi>) 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 at <http://docs.maksuturva.fi/>.

9 Paying through Maksuturva Payment Gateway

Perform a purchase of items and at the checkout select Maksuturva as the payment option during the checkout (see Illustration 3). You will see a confirmation page with your order details, you can still cancel your order here (see Illustration 4). Clicking on the Maksuturva button, you'll be redirected to Maksuturva service where you can proceed with the payment method of your choice. After completing the payment you'll see a confirmation of the purchase (see Illustration 5). The store admin will see the order status change from Pending to Processing.

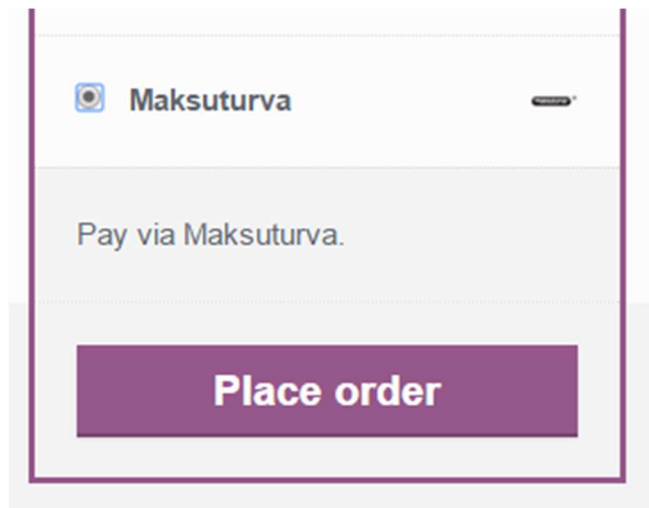


Illustration 3: Selecting Maksuturva as a

payment method

[Home](#) / [Checkout](#) / Pay for Order

Pay for Order

ORDER NUMBER:	DATE:	TOTAL:
147	March 11, 2016	28.88€

Maksuturva

Cancel order

Illustration 4:

Confirmation page

Order Received

Payment confirmed by Maksuturva.

Thank you. Your order has been received.

ORDER NUMBER:	DATE:	TOTAL:
147	March 11, 2016	28.88€

Order Details

PRODUCT	TOTAL
Happy Ninja × 1	38.50€
Subtotal:	38.50€
Discount:	-9.62€
Shipping:	Local Pickup
Total:	28.88€ (Includes 2.63€ Tax 10)

Illustration 5:

Message the customer receives in WooCommerce after completing the payment

10 Administration

When a customer chooses to place an order and is transferred to Maksuturva service the order's status is "pending". When the payment is placed and customer returns to the shop from Maksuturva, the status is changed to "processing". If an error occurs or customer cancels the payment, customer is returned to the store and the order status is changed to error or canceled.

10.1 Payment verification

The Maksuturva Payment Gateway Plugin automatically verifies payments from Maksuturva and displays the status on the order page. If the order is still pending after a certain amount of time, it's most likely because the customer has left the payment gateway or closed the browser window in the middle of the payment process. In this case it's OK for the admin to manually update the order status to canceled.

Payments are not verified while in Sandbox-mode and the verification do not work for bank payments when using Maksuturva test environment.