Installation Manual for Ogone Magento

This manual describes the installation and usage of the Ogone extension for Magento.

Release Date: Mon, 07 Aug 2023 14:01:06 +0200
Version: 3.0.434

wallee AG

General-Guisan-Strasse 47

CH-8400 Winterthur

E-Mail: info@sellxed.com

Telefon:

CH: +41 (0)44 505 13 60

DE: +49 (0)40 2999 60117

UK: +44 (0)161 850 6890

US: +1 (0)205 557 5869

####conditional####

Table of Contents

1 Introduction

This manual explains the installation, configuration and usage of the payment module for Magento and Ogone. It is important that you strictly follow the described steps. Only by doing so, a secure usage in compliance with all security regulations can be guaranteed.

Before beginning with the installation, please make sure that you are in possession of all necessary data:

- PSPID & Password. You receive these credentials from Ogone after conclusion of contract. Our support staff is happy to help you acquiring a Ogone contract if you do not yet have one.
- Magento payment module from sellxed.com/shop
- FTP access to your server
- Login credentials for your shop

Note that you must use at least PHP version 5.6 for our plugins. PHP 8 or higher is currently not supported.

1.1 Procedure of the Installation

1. Configuration of the administration interface of Ogone.
2. Upload of the module
3. Configuration of the basic settings of the payment module
4. Configuration of the **payment methods**
5. Carrying out a test purchase by means of the attached test data at the end of this document
6. If the test was successful, copy all data from your test account into the Ogone live account.

Installation Service

Our payment plugins should have per default the correct settings for most of our customers' preferences. That means once you have entered the required credentials in the plugin configuration to connect your account to your website, the plugin should be fully operational. Should you be willing to receive detailed information on a setting you do not know, you may contact our support team who will be able to assist you further.

Our support team is at your disposal during regular business hours at: http://www.sellxed.com/support. Furthermore, you have the option of ordering our installation service. We will make sure the plugin is installed correctly in your shop: http://www.sellxed.com/shop/de/integration-und-installation.html

.htaccess Directory Protection

In order to test the module, any kind of directory protection or IP blocking on your server must be deactivated. This is crucial as otherwise the payment feedback of Ogone might not get through to the shop.

1.2 System Requirements

In general, the plugin has the same system requirements as Magento. Below you can find the most important requirements of the plugin:

- PHP Version: 5.4.x or higher
- Magento: 1.7.x or higher
- OpenSSL: Current version with support for TLS 1.2 or higher.
- fsockopen: The PHP function fsockopen must be enabled. The plugin must be able to connect to external systems over the Internet.
- PHP Functions: All common PHP functions must be enabled.

2 Configuration Ogone - Backend

For the configuration, log into the Ogone backend:

- Test Environment: https://secure.ogone.com/ncol/test/backoffice/
- Live Environment: https://secure.ogone.com/ncol/prod/backoffice/

The settings carried out on the following pages are a precondition for the configuration of the main module.

The configuration of the settings is done under **Configuration > Technical Information**. In the following chapter the chapter titles correspond to the tabs in the back office of Ogone. For some parts screenshots will be added for a more precise illustration. Should a specific setting not be mentioned then it has no importance for the configuration of the payment module and can be ignored by you as such.

2.1 Tab: Global Transaction Parameters

Under the tab Global Transaction Parameters you define the basic settings for the processing of transactions.

2.1.1 Default operation code and default data capture procedure

We recommend choosing **Sale** and **"Automatic data capture by our system at the end of the day"**. All incoming payments will be directly processed and debited. If you are using transaction management and deferred capturing we strongly advise to use these settings. In case you have saved different settings in the module, then the module's settings will overwrite those of the Ogone.

2.1.2 Payment Retry

We recommend entering **10**.

2.1.3 Processing for Individual Transactions

We recommend the option **"Online but switch to offline when the online acquiring system is unavailable""**.

2.2 Tab: Global Security Parameters

Here you define the security parameters for the interaction between your shop and the gateway of Ogone.

2.2.1 Hashing Method

The sellxed payment modules support all hashing methods. In order for the module to function, it is crucial that the settings saved with Ogone correspond to those made in the backend of your shop.

In order to guarantee a smooth processing of payments, we recommend the following hashing methods:

- Each parameter, followed by the pass phrase. It may be that this parameter is not visible in your shop. In this case it is transmitted automatically.
- Hash Algorithm = SHA 512
- Character Encoding = UTF-8
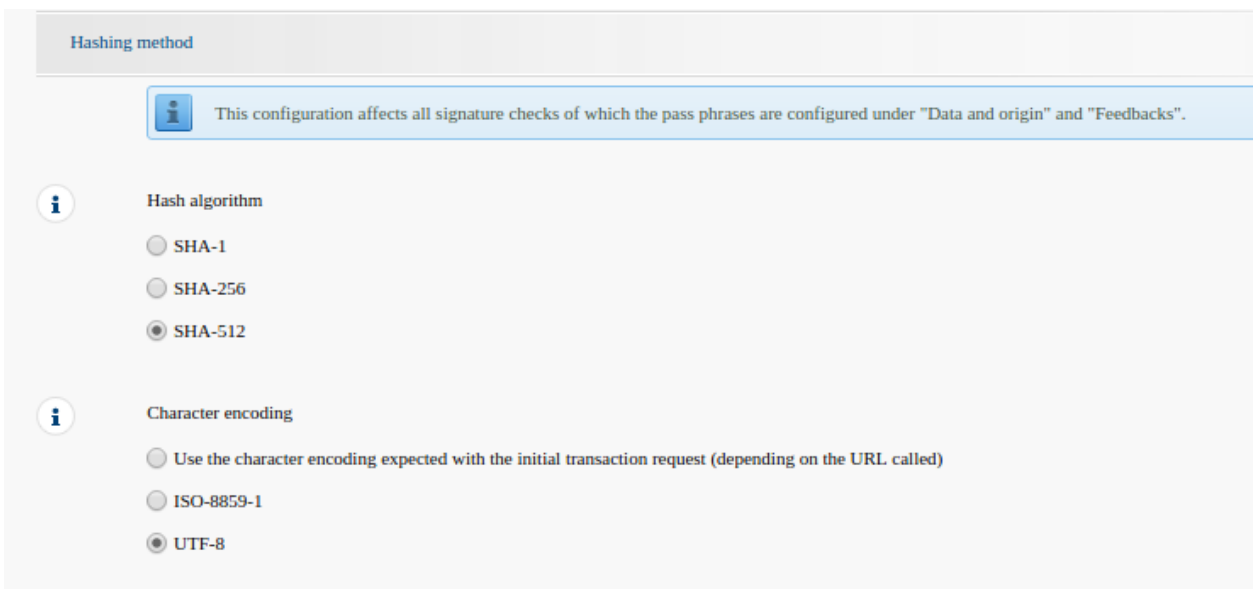
Figure 2.1: Configuration of the SHA-Hash.

These settings must be identical to the settings in the main module (cf. Configuration of the main module). Please note that the character encoding in our modules is set to UTF-8 per default and cannot be changed.

2.3   Tab: Payment Page

The checkbox under **Cancel Button** should be ticked. The field under Back Button Redirection can be left empty.

2.4   Tab: Data and Origin Verification

The tab Data and Origin Verification defines security elements concerning the transmission of data and the loading of the payment page.

2.4.1 Checks for e-commerce & Alias Gateway

The first box under "URL of the merchant page" can remain empty.

The box "SHA-IN pass phrase" is a security element of the payment verifying its authenticity. Therefore it is important that you use the same pass phrase in the back end and in the shop.

> ### Customweb SHA-String Generator
>
> Please note that only ASCII characters may be used for SHA pass phrases. Please use our SHA pass phrase generator so that you do not enter invalid characters.

2.4.2 Checks for Ogone DirectLink and Ogone Batch (Automatic)

DirectLink allows you to use the Hidden Authorisation (Alias Gateway) as well as the direct debiting of transactions within your shop. If you have activated DirectLInk in your account and wish to use this feature, enter **the same** SHA-IN pass phrase in the box "SHA-IN pass phrase" as above.

Further information on the configuration of DirectLink can be found under Configuration Hidden Authorisation.

2.5   Tab: Transaction Feedback

In the tab Transaction Feedback the response of Ogone to the shop is controlled.

2.5.1 HTTP Redirection in the Browser

The parameters in the transaction feedback determine the processing in the shop after payment has been completed. However, the four fields for "HTTP Redirection in the Browser" can remain **empty** because they will be configured through the settings of our module.

Un-tick the option "I would like Ogone to display a short text to the customer on the secure payment page if a redirection to my website is detected immediately after the payment process."

2.5.2 Direct HTTP-Server-to-Server-Request

- The "Timing of the Request" must be set to **Online but switch to a deferred request when the online requests fail**. As "Request Method" you must select **"POST"**.
- In the section **General** make sure that the Option: "I would like Ogone to display a "processing" message to the customer during payment processing." is **un-ticked** too.
- Copy here the URL that is listed in the module backend under: **System > Ogone > Setup for Magento 1 and under Stores > Ogone for Magento 2**.

URL on which the merchant wishes to receive a deferred HTTP request, should the status of a transaction change offline.

For the **Transaction Feedback** please enter the URL that is generated in the main module. More information can be found under the section <u>Transaction Feedback</u>

All orders have the status 'Pending'

Should you notice during the tests that orders are not registered in the shop and that no confirmation email is sent, this might be due to a failure of the transaction feedback. Make sure that your firewall is not blocking any request coming from Ogone.

Figure 2.1: Configuration of the feedback URL for the HTTP-Feedback.

Deferred Requests und Time Out

Ogone sends the notification requests with a time out of 20 sec. Should your shop not respond to the HTTP Request within this time, a time out error will be sent. If you set the option to **Online but switch to a deferred request when the online requests fail** the request will be retrieved four times in ten minute intervals.

Please note that, depending on your Ogone subscription, it is possible that the only option available is **Always online (immediately after the payment to allow customization of the response seen by the customer).** In this case, always select this option.

2.5.3 Dynamic e-Commerce Parameters

With this option you control which parameters are sent in the response from Ogone. In order for payments to effectively be evaluated in your shop, all necessary parameters must be sent. It is possible that you have other options available to select. Just make sure that you move every parameter from **available** to **selected** except for the following:

- PM
- CN
- SUBBRAND
- AAVADDRESS
- AAVCHECK
- AAVMAIL
- AAVNAME
- AAVPHONE
- AAVZIP

Figure 2.1: Configuration of the Dynamic E-Commerce Parameters. Your selection may look differently depending on your subscription with Ogone.

2.5.4 Dynamic Parameters

With this option you control which parameters are sent in the response from Ogone. In order for payments to effectively be evaluated in your shop, all parameters must be sent. It is possible that you have other options available to select. Just make sure that you move **every** parameters from **available** to **selected**.

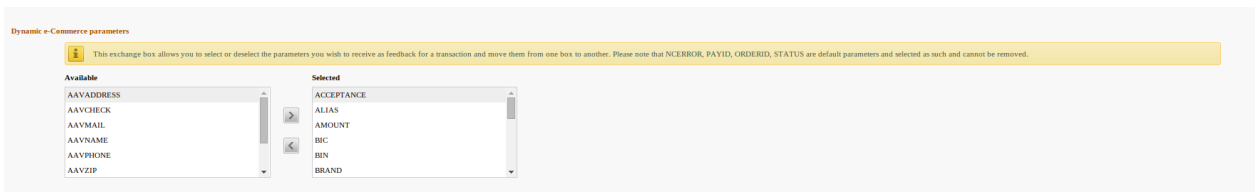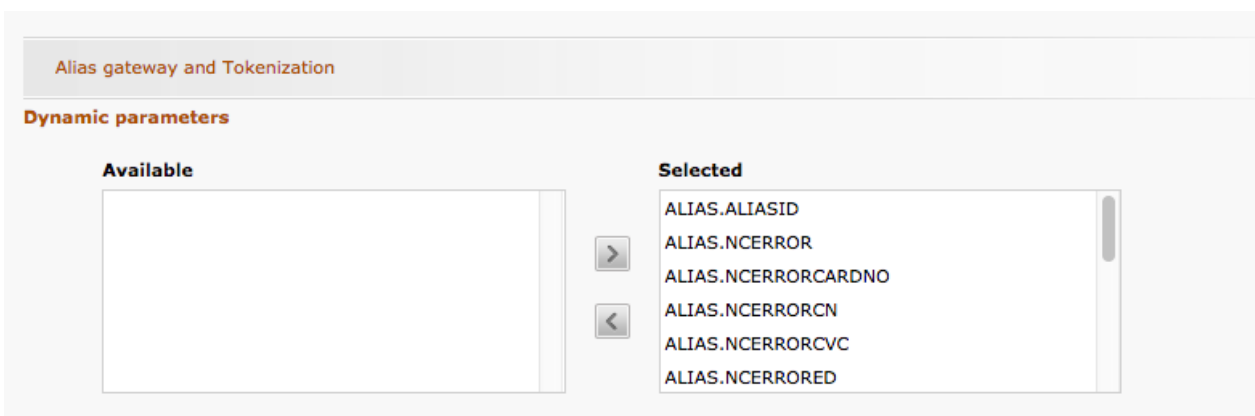**Figure 2.1:** Configuration of the Dynamic Parameters. Your selection may look differently depending on your subscription with Ogone.

2.5.5 Security for Request Parameters

The box "SHA-OUT pass phrase" is an additional safety mechanism for payments. Here too must the entered pass phrase be identical to the one in your shop. However, please make sure that your SHA-OUT pass phrase differs from the SHA-IN pass phrase. .

Customweb SHA-String Generator
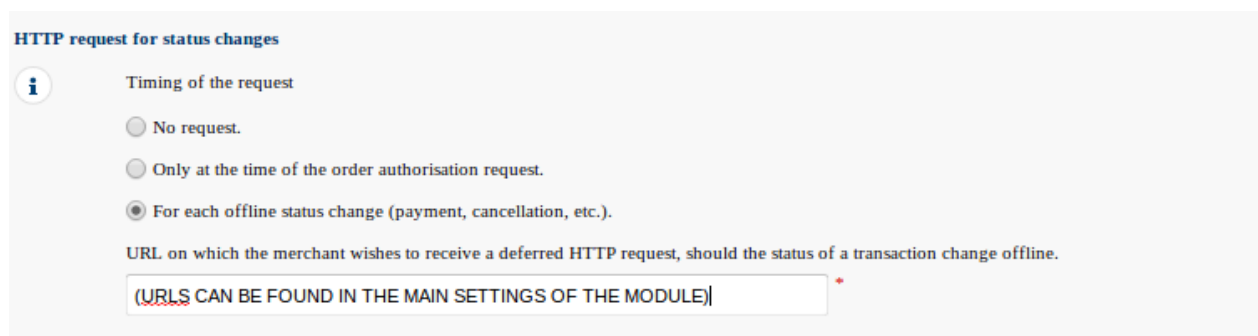
Please note that only ASCII characters may be used for SHA pass phrases. Please use our SHA pass phrase generator so that you do not enter invalid characters.

2.5.6 HTTP Request for Status Changes

You have the option to be informed about status changes by Ogone directly. This feature is especially useful when a payment has an uncertain status because the acquirer system could not be reached. The payment status will be updated automatically. Furthermore this feature allows you to capture payments inside Ogone and have synchronize them with your shop

In order to use this feature you will have to select the option **For each offline status change (payment, cancellation, etc.).**. Afterwards just copy the URL into the field that is displayed in the Setup of your Main configuration: **System > Ogone > Setup for Magento 1 and under Stores > Ogone for Magento 2.**

Enter the URL shown there into the field 'URL on which the merchant wishes to receive a deferred HTTP request, should the status of a transaction change offline.'

Figure 2.1: Enter the URL from the Setup of the Main Configuration.

2.5.7 Dynamic Parameters

Move all parameters from **available** to **selected**.

2.6   (Dynamic-)Template

You can now find the option template via **Configuration > Template**. The Dynamic Template enables you to design the payment mask for the credit card data so it matches your web shop. Should the field for the dynamic template not be visible, please contact Ogone. It may be that you have to activate additional options in your account.

For the usage of the dynamic template, please save the following settings:

- Enable JavaScript on template: **'No'**
- Our modules do not allow "usage of static template". So click 'No'
- Allow usage of dynamic template: **'Yes'**
- In case you are using the dynamic template, enter the host name, i.e. https://Your-Shop. com in the lower box. The upper one must remain empty.

Figure 2.1: Configuration of the Dynamic Template.

Dynamic Template

Please note that you require an SSL certificate for the usage of the dynamic template. Otherwise the CSS-Resources will not be reloaded in certain browsers which might lead to layout issues.

The activation of the dynamic template in the shop is explained here.

2.7   Hidden Authorisation, Ajax Authorization (FlexCheckout - Hosted Tokenization Page), External Checkout & Alias Manager

2.7.1 Preconditions

For the usage of the Hidden Authorization, Ajax Authorization (FlexCheckout), Alias Manager or External Checkout during the payment processing, certain adjustments are necessary in the backend of Ogone. Please check first if the options are activated in your account under **Account > Your Options**.

ID	Option	Activation	Monthly fee	Transaction/Unit cost	Status	
AFTP	AFTP Module	0 CHF	0 CHF	0 CHF	Active	i Deactivate
D3D	D3D (Direct Link 3D)	0 CHF	0 CHF	0 CHF	Active	i Deactivate
ELVDE	ELV (Direct Debits Germany)	0 CHF	0 CHF	0 CHF	Active	i Deactivate Configuration
FACQ	Foreign acquirers	0 CHF	0 CHF	0 CHF	Active	i Deactivate
INV	Invoice appearance	0 CHF	0 CHF	0 CHF	Active	i Deactivate
NPMA2	Foreign payment methods (fees per payment method activated) - CH	0 CHF	0 CHF	0 CHF	Active	i Deactivate
OPC	One Page Checkout	0 EUR	0 EUR	0 EUR	Active	i Deactivate
RECO1	Reconciliation Module	0 CHF	0 CHF	0 CHF	Active	i Deactivate
RECX	Alias Manager	0 CHF	0 CHF	0 CHF	Active	i Deactivate Configuration
SCH	Scheduled Payments	0 CHF	0 CHF	0 CHF	Available	i Activate
SUB	Subscription manager	0 CHF	0 CHF	0 CHF	Active	i Deactivate
USR4	User Manager up to 20 users	0 CHF	0 CHF	0 CHF	Active	i Deactivate Configuration
USR5	User Manager up to 50 users	0 CHF	0 CHF	0 CHF	Available	i Activate
USR6	User Manager up to 100 users	0 CHF	0 CHF	0 CHF	Available	i Activate
USR7	User Manager up to 200 users	0 CHF	0 CHF	0 CHF	Available	i Activate

(1) CAP1 : Includes transaction fee for 3-D Secure transactions

Figure 2.1: Make sure to activate the OPC and DPR options.

If an option is not available please contact Ogone.

Options and their Meanings:

- For the usage of the Hidden Authorization you require the option **OnePageCheckout (OPC).**
- For the usage of the Ajax Authorization (FlexCheckout) you require the option **OnePageCheckout (OPC)** and **DirectLink (DPR)**.
- If you additionally want to use the Alias Manager, you also need the option **Alias Manager (RECX)**.

In case these options are not be available, please contact Ogone directly.

2.7.2 Setting Up an API-User

If you want to use the features of the transaction management or the hidden authorization (Alias Gateway) you will need to set up an API user as described herafter:

Go to **Configuration > User Administration** and click on 'New User'. Choose a **User ID** and insert a user name of your choice and your e-mail address. Select **Administrator without user manager** as a profile. It is furthermore important that you select the option **Special user for API (no access to the administration)**. If you do not see this option, you will need to let Ogone activate the option DirectLink for your account.

> ### Password of the API-User
>
> The User ID and the API-user password may only contain letters and numbers as wel as special characters. The following special characters are not allowed: & (Ampersand), " (double quotes), ' (single quotes)

Note down the access data. They will have to be inserted in the respective place in the configuration of the main module.

2.7.3 Usage of External Checkouts

For the usage of external checkouts there are additional options required. External checkouts, such as MasterPass are payment methods, which offer the possibility to transfer addresses automatically.

The usage of the external checkouts asks for the option Direct Link and for the activation of payment method in your account.

For the performance of the payment method, you have to make sure that under Technical Information > Transactionfeedback in the section Direct Link all parameters selected are shifted

Figure 2.1: Response Parameter for Direct Link.

3 Module Installation and Update in the Magento Shop

3.1 Installation

At this time you should already be in possession of the module. Should this not be the case, you can download the necessary files in your customer account in the <u>sellxed shop</u> (Menu "My Downloads Downloads"). In order to install the module in your shop, please carry out the following steps:

1. Download the plugin. The download can be found in your sellxed.com account under "My Downloads".
2. Unzip the archive you have just downloaded.
3. In the unzipped folder navigate to the folder "files"
4. For some shops there are different versions of the plugin provided. If this is the case open the folder which corresponds to your shop version.
5. Using your preferred FTP client upload **entire content** of this folder into the root directory of your shop. For some shops there is a specific folder containing the plugins. If that is the case upload the plugin into this folder. Make sure that the folders aren't replaced but merely merged.
6. If you haven't yet done so, log back into your shop.

In the following you will find the installation manual for Magento 1 as well as Magento 2

3.1.1 Magento 1

Compiler Cache / Caching

Before you start the installation or after doing an update of the module, please make sure, that the compiler cache is disabled. Also make sure that you cleared the cache otherwise it could be that the settings page is not visible.

1. Create a back-up of your shop.
2. Extract the content of the ZIP file.
3. Copy the **entire content** of the folder "files" into the main directory of the server of your shop respectively to the folder where the plugins are copied normally. In order to do so, use your usual ftp program. Make sure that the folders aren't replaced but merely merged.

3.1.2 Magento 2

Im Folgenden werden die Schritte für die Installation und das Update der Extensions erklärt.

3.1.2.1 Installations Requirements

Bitte beachten Sie, folgende Voraussetzungen sind für den Betrieb von Magento 2 zwingend zu erfüllen:

1. System requirements von Magento (http://devdocs.magento.com/guides/v2.0/install-gde/system-requirements.html)
2. Der Cron Job muss korrekt eingerichtet sein, sonst können keine Extensions installiert werden. Weitere informationen fidnen Sie hier: http://devdocs.magento.com/guides/v2.0/config-guide/cli/config-cli-subcommands-cron.html

3.1.2.2 Installation

For the installation of the plugin in your Magento 2 store please follow these steps:

1. **Readiness Check**

   Before uploading the plugin it is necessary to do the Readiness Check for Magento 2

   1. 1. Save [this script](#) as cw-setup.php on your copmuter.
   2. 2. Upload this cw-setup.php Script into the root directory of your Magnto 2 installation.
   3. 3. Access your server via SSH and run this command: php cw-setup.php
   4. 4. The script will present you with additional commands which you need to execute.

   About the cw-setup.php script

   The provided script generates a set of commands which should be executed to clean up Magento 2. The script itself doesn't execute anything. It only generates the necessary commands.

2. **Upload**

   Upload the content of the folder **files_2** into the root directory of your server. Execute the commands provided by the cw-setup.php script again.

3.2 Updates and Upgrades

You have direct and unlimited access to updates and upgrades during the duration of your support contract. In order to receive constant information about available updates we ask you to subscribe to our RSS feed that we publish for your module.

More information regarding the subscription of this RSS feed can be found under: http://www.sellxed.com/en/updates_upgrades.

We only recommend an update if something doesn't work in your shop, if you want to use new feature or if there is a necessary security update.

3.2.1 Update Checklist

We ask you to strictly comply with the checklist below when doing an update:

1. Always do a backup for your database and your files in your shop
2. Use always a test system to test the update process.
3. Wait until all the files are copied to the shop, clear the cache if there is one in your shop and then visit the configuration page of the main module so that the update process will be initialized.

> ### Do not do updates directly in the live environment
>
> Please test the update procedure first in your test shop. Our support team is able and willing to help you if you experience problems with the update process. However, if you decide to perform the update directly in your live shop there is the possibility of a downtime of the shop of more than two days depending on the availability of our support if you do not want to book our complementary support.

Depending on the version it could be that the database has to be migrated. We recommend you therefore, to perform the updates in times when the shop is not visited too frequently by your customers.

3.2.2 Update Instructions

Please always read the update instruction. Those instructions can be found in the changelog. If there are no special remarks, you can proceed by just overwriting the files in your system.

3.2.2.1 Magento 1

1. Download the newest version of the plugin. The download is available in your sellxed.com account under "My Downloads".
2. Extract the ZIP file.
3. Upload the folder **files_1**into the root directory of your shop.
4. Clear the Magento cache

3.2.2.2 Magento 2

1. Download the newest version of the plugin. The download is available in your sellxed.com account under "My Downloads".
2. Extract the ZIP file.
3. Upload the folder **files_2**into the root directory of your shop.
4. Clear the Magento cache

3.3 Database Migration

This section describes how you can run the database migration script

This only needs to be done if your Magento installation shows error messages regarding missing tables or columns in the database.

3.3.1 Magento 1

Clear the Magento cache

If this does not resolve the issue, remove the entry ogonecw_setup from the core_resource table and clear the cache again.

3.3.2 Magento 2

From the command line run the following command:

php bin/magento setup:upgrade

4 Module Configuration in the Magento Shop

The configuration consists of two steps. The first step is the configuration of the main module with all the basic settings (cf. Configuration of the Main Module). During the second step you can then carry out individual configurations for each payment method . This allows for full flexibility and perfect adaptation to your processes.

> ### Create backups!
>
> Please create a backup of the main directory of your shop. In case of problems you will then always be able to return your shop to its original state.
>
> We furthermore recommend testing the integration on a test system. Complications may arise with third party modules installed by you. In case of questions, our support is gladly at your disposal.

4.1 Configuration of the Main Module

4.1.1 Magento 1

The settings for the main module can be found under **System > Configuration > Ogone**. Enter the various options, some of which you have already specified in the configuration with Ogone. Information on the functionalities as well as the impact of the individual settings can be found directly by the option.

The field "Time-Out for pending Payments" defines after how many minutes the order status should switch from pending to cancelled (0 means that nothing should be changed) if no successful feedback concerning the payment is received from Ogone. This might occur e.g. if the customer has closed his or her browser. Further information on this topic can be found under Order Status.

4.1.2 Magento 2

You will find the settings of the main module via **Stores > Configuration > Sales > Ogone > General** .

Fill in the fields according to the configuration of Ogone or with the data which you received directly from Ogone. You can find more exact explanations regarding each option in the help text by the input boxes.

Figure 4.1: Magento 2: Configuration of the basic module. Please note, that this is an example and the settings can differ.

4.2 Defining the URL for the Transaction Feedback

To ensure correct payment processing in your shop, at least one Feedback URL has to be provided in the Ogone configuration. The URL to be entered can be found in your Magento Shop under: **System > Ogone > Setup for Magento 1 and under Stores > Ogone for Magento 2 > Setup**

4.3 Configuration of the Payment Method

4.3.1 Magento 1

After having successfully installed the main module, you will find the individual payment modules under System > Configuration > Payment Methods. You can save individual settings for each payment method and thereby optimally adapt the payment methods to your existing processes. The most central options are described in more detail further below.

4.3.1.1 Direct Capturing of Invoices

The "Invoice Creation" specifies if invoices should be captured directly or not. Captured invoices can no longer be modified. If you wish to be able to modify the ordered article for debiting, please set this option to "Deferred".

4.3.2 Magento 2

After having successfully configured the main module, you will find the settings for each payment method in your shop via Stores > Sales > Payment Methods. For each payment method you can change the settings individually to optimize the payment processing in your shop.

4.3.2.1 Accounting

For each payment method you need to choose whether you want to register directly or delayed. A delayed booking only reserves the amount. Therefore it is important that you follow the instructions for such transactions as listed below. Reservations are usually only valid for a few days so we recommend to register directly. The delayed booking isn't available for every payment method.

4.3.2.2 Authorization Method

For some payment methods you can choose between different authorization methods. The authorization method determines how the payment form is presented in your shop. Please note that for some authorization methods with credit cards there are extended PCI requirements. (https://www.sellxed.com/en/blog/pci-300-changes).

4.4 Direct Capturing of Transactions

The option "Capture" allows you to specify if you wish to debit payments directly or if you first wish to authorise them and then debit the payment at a later point.

Depending on your acquiring contract, a reservation is only guaranteed for a specific period of time. Should you fail to debit the payment within that period, the authorisation may therefore no longer be guaranteed. Further information on this process can be found below.

> ### Different settings between Ogone and the module
>
> It may be that settings saved in the payment modules overwrite settings saved in Ogone.

4.5 Uncertain Status

You can specifically label orders for which the money is not guaranteed to be received. This allows you to manually control the order before shipment.

4.5.1 Setting the order state

For each payment method you may select in which state the order should be set to depending on the booking state. This is the initial state of the order.

4.6 Optional: Validation

Note: It can be that this option is not visible in your module. In this case just ignore this section.

With the option 'Validation' you can define the moment when the payment method should be made visible to the customer during the checkout process. This setting is relevant for modules where the usage depends on the customer's compliance with specific preconditions. For example, if a solvency check has to be carried out or if the payment method is only available in certain countries. In order for the credit check or address validation to also work with European characters, the charset of the "Blowfish mode" must be set to "UTF-8" for certain PSP settings.

You have the choice between these options:

- **Validation before the selection of the payment method:** A validation verification is carried out before the customer selects the payment method. If the customer does not fulfill the requirements, the payment method is not displayed
- **Validation after selection of the payment method:** The verification of the compliance occurs after the selection of the payment method and before the confirmation of the order
- **During the authorisation:** The validation verification is carried out by Ogone during the authorisation process. The payment method is displayed in any case

4.6.1 Usage of the Integrated Multishop Functionality of Magento

The payment module supports the multishop feature of Magento. No further modifications are necessary. The module automatically recognises the shop the order belongs to. In order for the multishop functionality to work, it is, however, necessary that the individual sub-shops within Magento have been configured correctly.

4.6.2 Usage of several independent Magento shops

In case you wish to connect several independent shops to one contract with one contract with Ogone, you require a sellxed payment module for each shop as well as the corresponding multi-shop module which you will find directly by the payment module.

1. Extract the multi-shop module and open the file "router.php"
2. Enter a shop ID of your choice in the box "Shop ID" within the main configuration of the module (e.g. shop_a and shop_b)
3. Enter the following line into the corresponding place within the file router.php of the multi-shop module (for each shop shop_a and shop_b):
   $redirect['shop_a'] = 'https://www.your-shop.com/index.php/OgoneCw/process/process';
4. Enter the URL below (instead of the URL mentioned above) under the point "Feedbacks" in the backend of Ogone: **http[s]://yourshop.com/router.php**
5. Copy the file router.php into the main directory of your shop

5 Settings / Configuration of Payment Methods

5.1 General Information About the Payment Methods

The plugin contains the most common payment methods. In case a desired payment method is not included per default, please contact us directly.

In order to be able to use a payment method, it must be activated in your account with Ogone as well as in your shop. Information about the configuration of the payment methods can be found further above.

Below you can find important information for specific payment methods that deviate from the standard process.

5.2 Information on Payment Status

For each payment method you can define an initial payment status (status for authorized payments etc.). You hereby define the payment status for each state depending on the processing type of the order (captured, authorized, etc.). It's the initial status which the order assumes. Depending on the mutation carried out by you, the status can change.

> ### Important info regarding Order Status
>
> Never set the status to **Pending Ogone** or any similar pending status which is implemented by the module.

5.2.1 Order status "pending" / imminent payment (or similar)

Orders with the status 'pending Ogone' are pending orders. Orders are set to that status if a customer is redirected in order to pay but hasn't returned successfully or the feedback hasn't reached your shop yet (Customer closed window on the payment page and didn't complete payment). Depending on the payment method these orders will automatically be transformed into cancelled orders and the inventory will be cleared (so long as the Cronjob is activated). How long this takes depends on the characteristics of the payment method and cannot be configured.

If you have a lot of pending orders it usually means that the notifications from your webserver to Ogone are being blocked. In this case check the settings of your firewall and ask the Hoster to activate the IPs and User Agents of Ogone.

5.2.2 Order status "cancelled"

Orders with the status "cancelled" have either been set to that status automatically due to a timeout, as described above, or have been cancelled directly by the customer.

5.3 Usage Hidden Authorisation / Alias Gateway

You have the possibility of processing credit cards without redirection to the payment page of Ogone.

In order to be able to use this feature, you must first carry out the steps listed under Hidden Authorisation / Alias Gateway and create an API-User.

You can then activate the authorisation method 'Hidden Authorisation' for the payment methods of your choice. This procedure requires you as a merchant to fulfill the PCI-SAQ-A-EP. At no time is any credit card data saved on your server. Contact your Acquirer for more information about the SAQ and the requirements.

5.4 Credit Card

For the integration you have the choice between the pre-selection in the shop; i.e. the customer chooses the card type already in the shop; or a generic integration in which case the customer only sees the option 'credit card' as a payment method and then chooses the card type on the payment page Ogone.
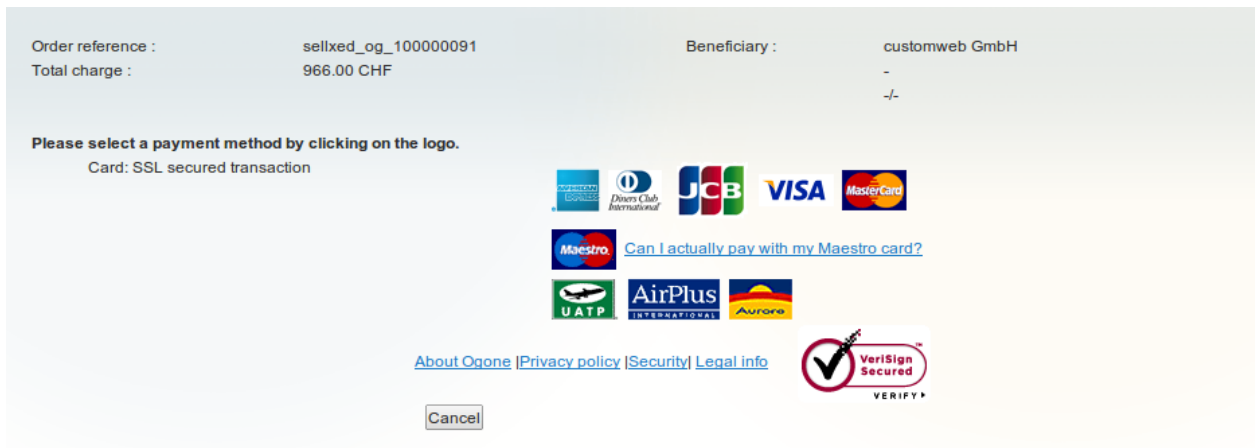
Figure 5.1: No pre-selection of the card type - the type will then be chosen on the payment page.

Pre-Selection of the Payment Method

We recommend choosing the option pre-selection. This allows you to configure individual settings for each payment method and add payment fees to the various methods.

5.5 PayPal

You can process PayPal directly via Ogone. In order to do so, however, some adjustments to your PayPal account are required. Please carry out the following steps:

1. Activate the payment module in your shop
2. Make sure that the payment method is activated in your Ogone account
3. The API adjustments must be carried out in your PayPal account. The various steps are explained in the "PayPal Express Checkout Integration Guide". This document can be found in your PSP account under: Support > Integration and User Guides > Technical Guides > Ogone PayPal Express Checkout Integration Guide .

5.6 Bank Transfer

In order to use the payment method Bank Transfer, make sure that you configured the settings for the HTTP status changes as described above. Payments done via the payment method bank transfer will first be marked as pending. As soon as the client pays the invoice, Ogone sends us the notification for the status change and we will update the state of the payment method.

5.7 MasterPass Integration: What to keep in mind

If you integrate MasterPass as a payment method in your shop, you have two integration options: External and Standard Checkout. When using External Checkout, the customer does not have to provide address data during checkout. The customer only needs a MasterPass account and the

data will be transmitted by MasterPass. In order to use this option as a merchant, you have to activate this payment method with Ogone.

Please keep in mind, that using additional functions such as address transmission through MasterPass, may require the activation of the respective options through Ogone aktiviert.

In order to use MasterPass in compliance with the MasterCard guidelines, please read the following information carefully before using this payment method.

5.7.1 Integration Possibilities for MasterPass

MasterPass can be integrated as a normal payment method with redirection to Ogone or as an external checkout.

5.7.1.1 MasterPass as a Payment Method

If you like to integrate MasterPass as a normal payment method, activate the payment method as usual (seeConfiguration of the payment method). In this scenario the customer will first be redirected to MasterPass and then to the wallet of the issuer.

5.7.1.2 MasterPass as External Checkout

If you integrate MasterPass as an external Chekout, you can minimize the steps towards the payment radically. The adress transfer and the creation of the account is processed via MasterPass. If you activate the external checkout, there appears a button in the basket with which you can jump after the login directly to the last step of the order confirmation. The creation of the account happens in the background.

For the activation of the external checkout go to **System > Ogone for Magento 1 and under Stores > Ogone for Magento 2** . There you'll find all settings for the external checkout:
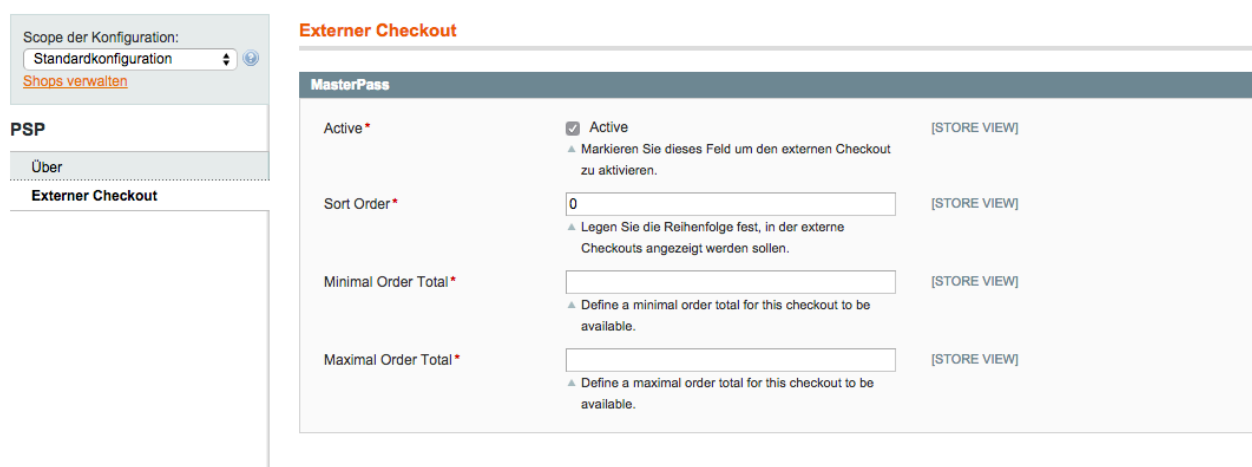

Figure 5.1: Activation of the external checkout.

MasterPass as external checkout is not for all cases available. If this point in the manual doesn't appear, please make sure to contact us.

5.7.2 Checkout: MasterPass Learn More Button

When using MasterPass as a payment method for standard checkout, it is mandatory to link to the MasterPass Learn More Page. In order to do this, navigate to Payment Method Configuration. Paste the link for the Learn More Page into the field 'Description' for the respective languages. The field 'Learn More' will now be displayed when choosing a payment method.

We have already collected the most important links for you. You only have to copy and paste the links into the description field:

- German:Mehr Erfahren
- French:apprendre plus
- English: Learn More
- Spanish:aprender más
- Italian:saperne di più
- Dutch:Leer Meer

Figure 5.1: Example for entering a description (Magento). These settings are to be found in the payment method configuration for MasterPass.
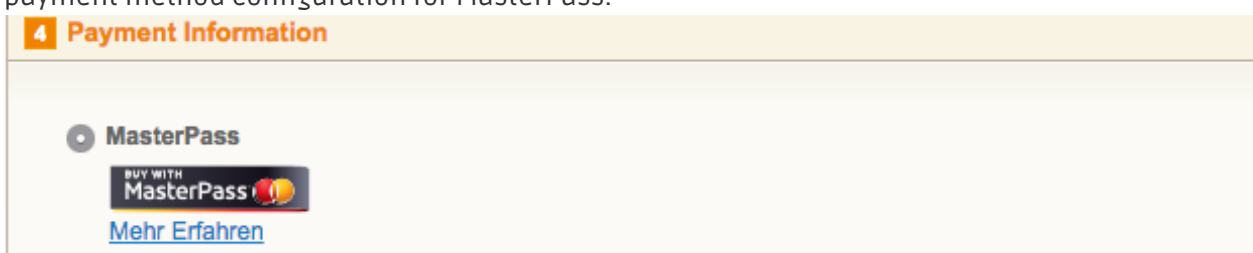
Figure 5.1: The Learn More Page link should appear in the frontend.

Links for additional languages can be found in this document: <u>https://developer.mastercard.com/...</u>

5.7.3 Additional information

General and additional information on using the MasterPass trademark and logo in your webshop are available in the official MasterPass Mechant Branding Requirement Document. which you can access here: <u>https://developer.mastercard.com/...</u>

5.8 Usage of the Ajax Authorization (FlexCheckout)

You have the possibility of processing credit cards without redirection to the payment page of Ogone.

In order to be able to use this feature, you must first carry out the steps listed under <u>Hidden Authorisation, FlexCheckout (Hosted Tokenization Page), External Checkout und Alias Manager</u> and create an API-User.

You can then activate the authorisation method 'Ajax Authorization (FlexCheckout)' for the payment methods of your choice. At no time is any credit card data saved on your server.

6 The Module in Action

Below you will find an overview of the most important features in the daily usage of the Ogone module.

6.1 Capturing / Cancelling of Orders

6.1.1 Magento 1

Please note:

The transaction management between your shop and Ogone is not synchronised. If you capture payments with Ogone, the status in the shop will not be updated and a second capturing in the shop is not possible.

6.1.1.1 Case 1: Invoice Capturing Directly After the Order

In the following, the process is explained for the case that you have chosen to directly create the invoice after having placed the order and only authorise payments - not capture them directly - (further information on the configuration can be found under Creating the Invoice).

In order to capture the order, open the invoice and click "capture". If you click "Cancel", the authorisation will be cancelled.

6.1.1.2 Case 2: Deferred Invoice Capturing

In case you have chosen the deferred invoice creation, you first have to create the invoice by clicking on the button "Invoice". Select "Capture online" below the total amount and then click on "capture invoice". The invoice has now been captured.

6.1.1.3 Cancel of orders/transactions

In order to cancel a transaction you first have to open the invoice and click on the cancel button. After that open the order and click on cancel here again. This will send the cancel of the order to Ogone. Please note that you need to cancel the invoice and the order.

6.1.2 Magento 2

6.1.2.1 Capture Bug in Magento (v. 2.0.0)

In version 2.0.0 of Magento there is a bug. It could be that invoices cannot be captured. Please make the following adjustments in the file **/vendor/magento/module-sales/Controller /Adminhtml/Order/Invoice/Capture.php**:

1. Delete the lines in the picture bellow which begin with a - .

2. Add the lines which has been marked with a + .

Here you can find the codes to be inserted in the form of a text:

+)->addComment(__('The credit memo has been created automatically.'));

```
Method: execute()

Diff:

        try {
-            $invoiceManagement = $this->_objectManager->get('Magento\Sales\Api\InvoiceManagementInterface');
-            $invoiceManagement->setCapture($invoice->getEntityId());
+            $invoice->capture();
             $invoice->getOrder()->setIsInProcess(true);
```

Figure 6.1: Opening the Invoice

6.1.2.2 Performing Captures

> **Please note**
>
> The transaction management between Ogone and your shop isn't synchronized. When you capture the payments at Ogone , the status is not adjusted and a second capture isn't possible in your shop.

In the configuration of the payment method you can decide whether the invoice should be captured after having been created or if only the payment is authorized. You also decide if you already want to create an invoice. If you create an invoice you can't make any adjustments for the capture.

In order to capture invoices please take the following steps:

1. Open the order and create an invoice.
2. If you have set the Invoice Settlement to deferred you can still make adjustments to the invoice.
3. Click on Capture Online.

Figure 6.1: Capture options

6.1.2.3 Cancelling / Voiding Orders

In order to issue a cancellation of the payment you must use the "Void" button. If you wish to use the "Cancel" button you must cancel the payment in the Ogone backend portal.

6.2   Place Orders in the backend of Magento

6.2.1   Magneto 1

With the Magento payment module you can place orders and capture the payment directly in the shop; so called Mail Order / Telephone Order (MOTO).

1. Go to the order overview and click on "Create New Order"
2. The usual window for the creation of orders will be opened. Select the customer in question, the item and the address
3. Select the preferred payment method and click on "Submit Order".
4. Depending on the authorisation method that you have saved for the selected payment method, you will either be redirected to the Payment Page of Ogone or the mask for the credit card will appear such as in the image below. Enter the credit card data of the customer.

Figure 6.1: MoTo Transactions in Magento

6.2.2 Magento 2

With the Magento payment module you can trigger orders directly from your shop, so called Mail Order / Telephone Order (MOTO).

1. Go to the overview of the order (Sales > Orders) and click on "Create New Order"at the top right.
2. The familiar window to create an order will open. Choose the desired customer, article and address.
3. In the payment method box, choose the desired payment method and click on "Finalize order".
4. After that you will be redirected either to the Payment Page of Ogone or you will be asked to enter your payment information in the shop, depending on your settings.

If an alias of the customer exists, you can alternatively use that for the order.

6.3 Refunds

6.3.1 Magento 1

You can refund already captured transactions and automatically transmit them to Ogone. In order to do so, open the invoice of the captured order. By clicking on "Credit Memo" the window for refunds is opened. Enter the amount you wish to refund. By clicking on "Refund" the refund is transmitted to Ogone.

Below you find a step-by-step guide including screenshots:

Figure 6.1: 1. open the invoice

Figure 6.1: 2. click on credit memo - Important - open the invoice first!.

Figure 6.1: 3. Adjust the refunding amount (see more info below Refunds Based on Quantity)

Figure 6.1: 4. To process the refund click on **Refund.**

Figure 6.1: Wait for the confirmation of the refund.

6.3.1.1 Refunds Based on Quantity

You can make refunds based on product quantity. In order to do so, open the credit memo window and enter the amount of products you wish to refund to your customer's card

You can carry out as many individual refunds as you wish until you have reached the quantity of the original order. The status of the order then automatically switches to **complete**.

Figure 6.1: Overview of all Refunds

6.3.1.2 Refunding of Any Amount

As each position of the order needs to be transmitted to Ogone we can't support the function **Adjustment Refund** - meaning the refund of any amount. Only whole items can be refunded by adjusting the refund amount.

Adjusting the amount of several refunds

It is possible to carry out several refunds on an order. Please note that you can refund a maximum of 100% of the original amount. If you have already carried out a refund you need to make sure not to refund more than the initial amount by adjusting the amount and the Adjustment Fee.

6.3.2 Magento 2

6.3.2.1 Resolving a Current Magento Bug (v. 2.0.0)

In the 2.x version of Magento there is a bug. Please make the following adjustments to the file **/vendor/magento/module-sales/Model/Order/Payment.php** :

1. Delete the lines in the picture bellow which begin with a - .
2. Add the lines which has been marked with a + .

Method: registerRefundNotification($amount)

Diff:

```
        $creditmemo->setPaymentRefundDisallowed(
            true
        )->setAutomaticallyCreated(
            true
-       )->register()->addComment(
+       )->addComment(
            __('The credit memo has been created automatically.')
        );
```

Figure 6.1: Adjustments so you are able to enter a credit memo.

Here you can find the codes to be inserted in the form of a text:

+)->addComment(__('The credit memo has been created automatically.'));

6.3.2.2 Performing Credits

For registered transactions you can create credits and transmit them automatically to Ogone . In order to do so open an invoice of an already registered order. By clicking on "Credit Memo" a dialog will open for the refunds. Now you can adjust the number of products which you would like to credit in the credit memo dialogue.

Here you can find each step again including screenshots:

Figure 6.1: 1. Opening the Invoice

New Memo for #000000003

← Back   Reset

Order & Account Information

Order # 000000003 (The order confirmation email was sent)

Order Date	Dec 14, 2015, 7:54:41 AM
Order Status	Processing
Purchased From	Main Website Main Website Store Default Store View
Placed from IP	212.90.201.186

Account Information

Customer Name	Guest
Email	info@customweb.com
Customer Group	NOT LOGGED IN

Address Information

Billing Address Edit

Test Customweb
Teststrasse 2
Zürich, Zug, 8032
Switzerland
T: 555466684

Delivery address Edit

Test Customweb
Teststrasse 2
Zürich, Zug, 8032
Switzerland
T: 555466684

Figure 6.1: 2. Click on Credit Memo - Important the invoice needs to be opened first.

Refund Totals	
Subtotal	156,00 €
Refund Shipping	5
Adjustment Refund	0
Adjustment Fee	0
Grand Total	**161,00 €**

Append Comments ☐
Email Copy of Credit Memo ☐

✔ Refund Offline   ✔ Refund

Figure 6.1: 3. Choose which products you would like to credit and click on Refund Online.

Figure 6.1: Overview of all the credits made

> ## Adjusting the amounts for several refunds
>
> It is possible to issue several credits on one order. Please note that you can't credit more than 100% of the original amount. If you have already issued a credit you will need to make sure you don't credit more than the original amount, by adjusting the amount and the Adjustment Fee.

6.4   Useful Transaction Information on the Order

6.4.1   Magento 1

In each order, processed via our module, you can find an overview of the most important information about the transaction as well as a transaction history.

Figure 6.1: Transaction Information.

6.4.2 Magento 2

In each order, which was processed by our module, you will find an overview of the most important information on the transaction and transaction history.

Payment & Shipping Method

Payment information

VISA
Visa

Transaction ID	14
Authorisation Amount	107
Currency	CHF
Payment period	Visa
Payment ID	h1ElhGAKOSCItASfE0IIbOShnndA
Test Transaction ❓	Yes
Transaction authorised	Yes
Transaction uncertain ❓	No
Transaction paid	Yes
Card Type	Saferpay Test Card
Effective Payment Method Name ❓	SaferpayTestCard
Authorisation Method	PaymentPage
3D Secure	Successful

Dec 14, 2015   11:08:14 AM | Authorization
The amount of 107.00 is authorized. (Customer sucessfully returned from the Saferpay payment page.)

The order was placed using CHF.

Shipping & Handling Information

Flat Rate - Fixed CHF10.00

Figure 6.1: Transaction Information.

6.5   Usage of the Alias Managers / Token Solution

6.5.1   Magento 1

With the Alias Manager, your customers can securely save their credit cards with Ogone for later purchases. You can enable this by activating the option "Alias Manager" in the configuration of the Payment Method. The customer can then choose from their saved credit cards without having to enter all the details again.

Figure 6.1: Alias Manager Transaction

Attention

To use the Alias Manager tit may be necessary to activate certain options in your Ogone account. Please contact Ogone directly to do so.

6.5.1.1 Deleting Customer Aliases

You can easily delete a customer's alias by opening the customer in the backend of Magento. Switch to the tab Ogone Alias and delete the saved alias directly here.
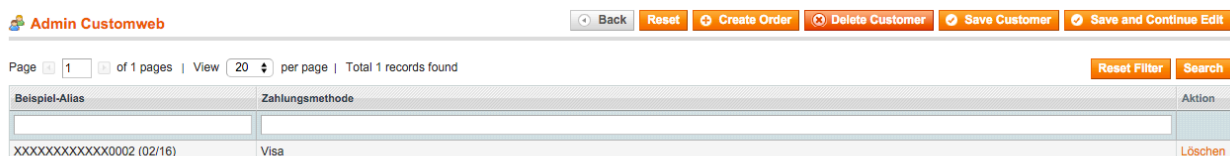
Figure 6.1: Delete previously saved aliases of your customers.

6.5.2 Magento 2

Due to the integration of the Alias Manager the customers credit card data can be saved for later purchases. The customer can choose between the cards that have already been saved and won't have to enter all of credit card data again.

6.5.2.1 Activating the Alias Manager / Token

In order for you to be able to use this you will need activate the option "Alias Manager" in the payment methods.

Figure 6.1: Alias Manager Transaction

6.5.2.2 Customers Deleting the Token

In Magento 2 you can allow your customers to manage the aliases in their customer account at "My Ogone Aliases / Token"". This feature can also be enabled in the settings of the payment method

Shop

Customer Alias Management	Yes ▾	[STORE VIEW]

You can allow the customers to manage/delete their aliases in their account.

Await Notification	Yes ▾	[WEBSITE]

In some cases it may take a few seconds until the transaction is successfully authorized. When this option is enabled, the notification is awaited.

External Checkout: Guest Checkout	Show Account Selection ▾	[WEBSITE]

When using an external checkout, the customer can either be asked to choose an option to authenticate (as guest, register or login) or he can always be checked out as guest. For the second option to work, guest checkout has to be enabled in Magento.

Figure 6.1: Activating the Alias Manager for Customers.

6.6 Dynamic Template

We provide a standard skin for the usage of the dynamic template for all sellxed modules. You can activate the option by setting the option "Dynamic Template" to "Use Shop Template" in the main module. Please make sure that all settings specified under Template have been carried out in the backend of Ogone.

The Dynamic Template allows you to load the Payment Page of Ogone in the layout of your shop. The customer, therefore, does not notice that the customer has been forwarded to the website of Ogone. The template dynamically adjusts to your shop template. This, furthermore, allows you to easily integrate Mobile Skins (cf. Screenshots).

The Dynamic Template requires an SSL Certificate

In order for the Dynamic Template to work correctly with all browsers, you require an SSL certificate in your shop.
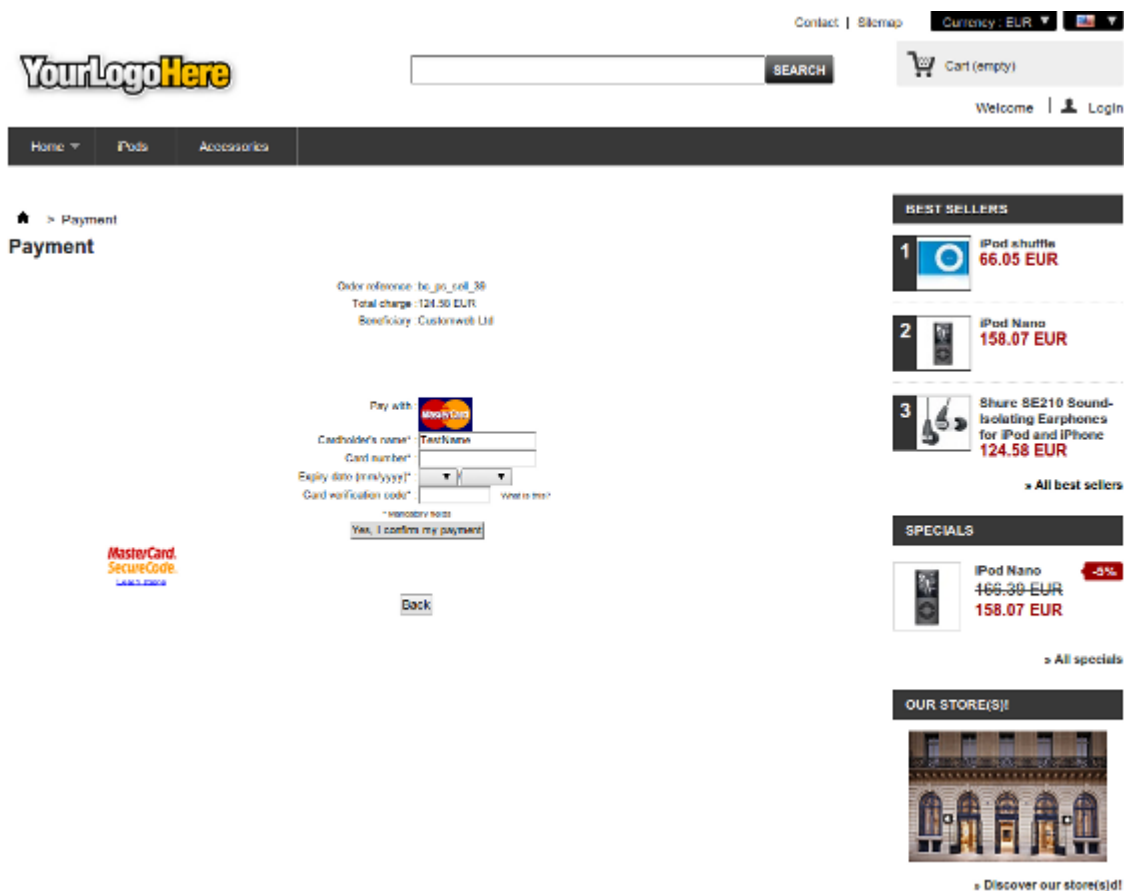
Figure 6.1: Example: The skin adapts to the layout of your shop.
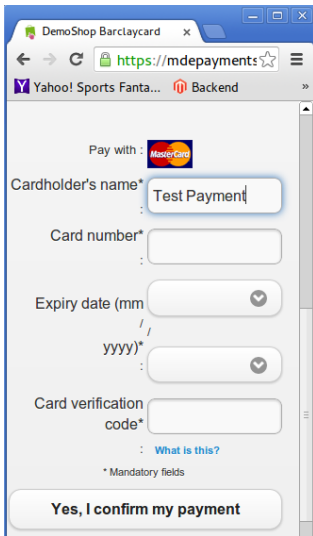
Figure 6.1: Example: Dynamic Template with mobile skin

6.7 Tax Calculation Settings

6.7.1 Magento 1

To ensure that the taxes are calculated correctly and transmitted correctly to a third party invoice provider inside Magento please make sure that you set the calculation settings in the menu under System > Configuration > Tax as showed in the screenshot below.
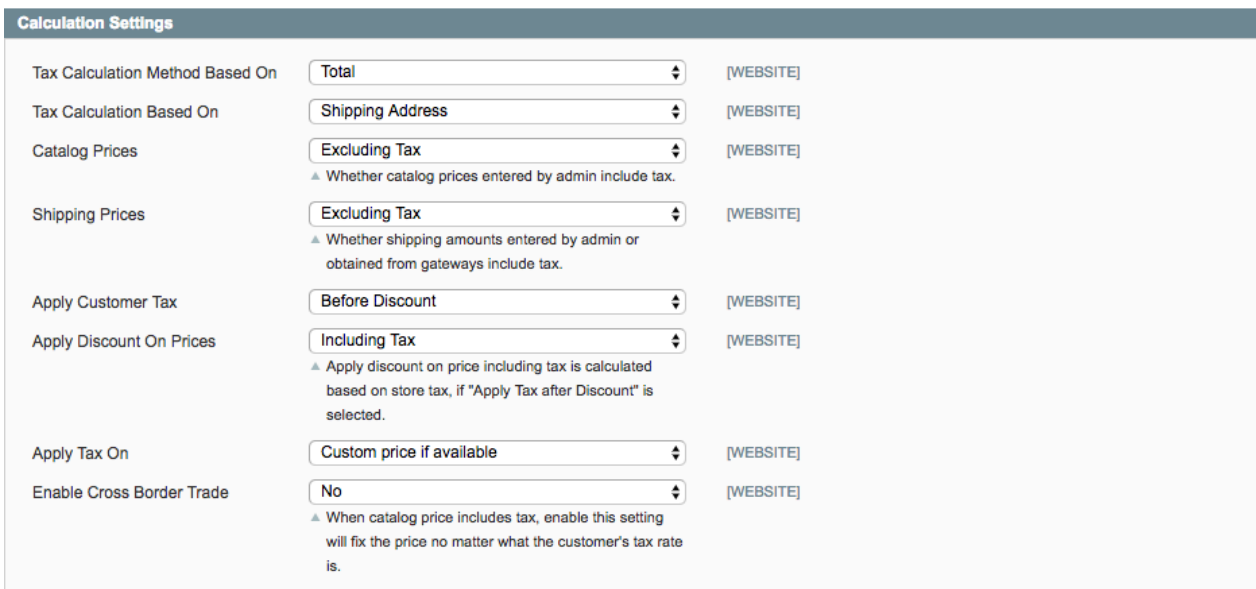
Figure 6.1: Recommended tax calculation settings for Magento.

Please also make sure that inside the settings of your promotion in the tab "Actions" the discount is not applied to the shipping amount as Magento is not correctly transmitting the tax rate there.

6.8 OneStepCheckout

6.8.1 Magento 1

The checkout of Magento can be modified and reduced to one step. The so called OnePageCheckout. Our modules support the most important OnePageCheckout extensions. If you are thinking of using a OneStepCheckout extension in combination with our Ogone Magento, please make sure that you are using one of the following:

- OneStepCheckout by www.onestepcheckout.com
- OneStepCheckout by Ahead Works
- LightCheckout by GoMage
- Firecheckout by Templates Master

6.8.2 Magento 2

> There is no official support for any OnePageCheckout module for Magento 2.

6.9 Setup a Cron Job to Activate the Timed Operations

6.9.1 Magento 1

To activate the timed operations of the plugin (e.g. update service, deleting pending orders, etc.) make sure that you set up the Magento Cron engine. Especially the update function allows you to automatically retrieve additional information or changes of your order directly via the API of Ogone. Please note it could be that in order to use the update feature it may be necessary that Ogone activates additional options in your account.

The module uses the standard cron engine of Magento. More information regarding the set up can be found here.

Here we suggest you use a Cron Engine like for example EasyCron. This allows you to open the file (URL) with an external service.

6.9.2 Magento 2

The module uses the standard cron engine of Magento 2. More information regarding the set up can be found here.

6.10 Magento 1 API Manual

Via API, our module provides plenty of information on a transaction. The following section describes which calls are possible and what information is provided.

6.10.1 General Information on Using the API

General information on using the Magento SOAP API can be found here: http://www.magentocommerce.com/api/soap/introduction.html

6.10.2 API Calls

This overview shows the API Versions and which API Calls they support.

6.10.2.1 API V1

- $client->call($session, 'ogonecw_transaction.list'); //List of all transactions
- $client->call($session, 'ogonecw_transaction.info', {transactionsId}); // Single Transaction

6.10.2.2 API V2

- $client->ogonecwTransactionList($session); //List of all transactions
- $client->ogonecwTransactionInfo($session, {transactionsId}); //Single Transaction

6.10.2.3 XML-RPC

- $client->call('call', array($session, 'ogonecw_transaction.list')); //List of all transactions
- $client->call('call', array($session, 'ogonecw_transaction.info', {transactionsId})); //Single Transaction

6.10.2.4 REST

- $this->callApi('ogonecw/transactions'); //List of all transactions
- $this->callApi('ogonecw/transactions/{transactionsId}'); //Single Transaction

7 Testing

Before switching from test to live mode it is important that you test the module extensively.

> ### Testing
>
> Do not forget to switch the operating mode from test to live after having successfully tested the module.

7.1 Test Data

In the following section you can find the test data for the various payment methods:

Credit / Debit Card

Card number	3750 0000 000 0007	American Express
Expiry Date	12/2020	
CVC	1234	
Card number	5399 9999 9999 9999	Mastercard
Expiry Date	12/2020	
CVC	123	
Card number	4111 1111 1111 1111	VISA (Without 3-D Secure Check)
Expiry Date	12/2020	
CVC	123	
Card number	4000 0000 0000 0002	VISA (With 3-D Secure Check)
Expiry Date	12/2020	
CVC	123	

American Express

Card number	3750 0000 000 0007	American Express
Expiry Date	12/2020	
CVC	1234	

MasterCard

Card number	5399 9999 9999 9999	Mastercard
Expiry Date	12/2020	
CVC	123	

Visa

Card number	4000 0000 0000 0002	VISA (With 3-D Secure Check)
Expiry Date	12/2020	
CVC	123	
Card number	4111 1111 1111 1111	VISA (Without 3-D Secure Check)
Expiry Date	12/2020	
CVC	123	

8   Errors and their Solutions

You can find detailed information under http://www.sellxed.com/en/faq. Should you not be able to solve your problem with the provided information, please contact us directly under: http://www.sellxed.com/en/support

8.1   Module is not shown

If the module is not shown even though the files have been downloaded onto the server correctly, please check the following:

- The local.xml can be found in the /app/etc directory
- Make sure **disable_local_modules** is set to ‚false‘

8.2   The Referrer URL appears in my Analytics Tool

When a customer and the notification are redirected via Header Redirection, the Ogone Referrer URL might appear in your Analytics Tool thus hiding the original traffic source. However, most Analytic Tools are able to minimize this problem.

In case you are using Google Analytics as reporting tool, this step by step guide may help you to exclude the URLs: under bullet point 4.

9 Compatibility with Third-Party Plugins

The plugins listed below are compatible with our payment modules and allow you to handle certain tasks in a easier way.

9.1 Magento 1

9.1.1 Fees and discounts within Magento

To configure a Ogone payment gateway based fee and discount, you will need the following 3rd-Party plugin.

- [Fooman Surcharge](#)

9.2 Birthday and gender in Magento 1

For certain payment service providers it is necessary to check the birthday an the gender of a customer. Magento 1 does not check this by default. this check can be enabled under "Go to System > Configuration | Customers > Customer Configuration > Name and Address Options".

9.3 Birthday and gender in Magento 2

For certain payment service providers it is necessary to check the birthday an the gender of a customer. Magento 2 does not check this by default. this check can be enabled under "Stores > Configuration | Customers > Customer Configuration > Name and Address Options".

10 Error Logging

The module will log different unexpected errors or information depending on the configured level. If there is any issue with the module, this log can help identify the cause.

10.1 Log Levels

You can configure the log level in the Ogone settings. The store log setting does not affect these log messages.

- Error: Logs unexpected errors only. (Default)
- Info: Logs extended information.
- Debug: Logs information helpful for debugging.

10.2 Log Location

The log file is stored in the Magento log directory. (Default Path: {shopRootDirectory}/var/log)

11 Advanced Information

This section of the manual is for advanced usage of the module. The content is for advanced users with special requirements. Everything in this section is optional and not required for the daily usage of the module.

11.1 Transaction Object

This section describes how to extract information from a transaction, if you need it for further processing. E.g. you require more information of the transaction for further processing an order in your ERP system.

The code snippets in this section assume your script resides in the root folder of the shop with the default shop folder structure.

11.1.1 Magento 1

Initialize the shops core functions.

```
define('MAGENTO_ROOT', getcwd());
$mageFilename = MAGENTO_ROOT . '/app/Mage.php';
require_once $mageFilename;
Mage::setIsDeveloperMode(true);
ini_set('display_errors', 1);
umask(0);
Mage::app();
```

Now you can load the transaction and then extract the transactionObject.
Load the transaction object by Id:

```
$dbtransaction = Mage::getModel('ogonecw/transaction')->load
($transactionId);
$transactionObject = $dbtransaction->getTransactionObject();
```

Load the transaction object by Order Number:

```
$dbtransaction = Mage::getModel('ogonecw/transaction')->load
($orderId,'transaction_external_id');
$transactionObject = $dbtransaction->getTransactionObject();
```

11.1.2 Magento 2

Initialize the shops core functions.

```
use Magento\Framework\App\Bootstrap;
require __DIR__ . '/app/bootstrap.php';
$bootstrap = Bootstrap::create(BP, $_SERVER);
$obj = $bootstrap->getObjectManager();
```

```
$state = $obj->get('Magento\Framework\App\State');
$state->setAreaCode('frontend');
```

Get the transaction factory.

```
$factory = $obj->get
('Customweb\OgoneCw\Model\Authorization\TransactionFactory')->create();
```

Now you can load the transaction and then extract the transactionObject.
Load the transaction object by Id:

```
$dbTransaction = $factory->load($transactionId, 'entity_id');
$transactionObject = $dbTransaction->getTransactionObject();
```

Load the transaction object by Order Number:

```
$dbTransaction = $factory->load($orderId, 'order_id');
$transactionObject = $dbTransaction->getTransactionObject();
```

Load the transaction object by Payment Id:

```
$dbTransaction = $factory->load($paymentId, 'payment_id');
$transactionObject = $dbTransaction->getTransactionObject();
```