Installation Manual for PAYONE PrestaShop

This manual describes the installation and usage of the PAYONE extension for PrestaShop.

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 describes how to install configure and use the payment module for PrestaShop and PAYONE.

Before you get started, make sure to have all necessary data at hand.

- You received a user name and a password from PAYONE. Log in to the PAYONE Merchant Interface. (https://pmi.pay1.de/)
- PrestaShop payment module from sellxed.com/shop
- Login data for your server and shop

2 PAYONE Channels

According to the functions of the module which you want to use, you have to request the different channels at PAYONE. With the numbers of channels come various costs on the part of the PAYONE. Here you'll find an overview of the different functions and their corresponding channels.

- **Channel Frontend:** Payment page
- **Channel Client-API:** Hidden, Alias Manager, Ajax, Moto
- **Server-API:** Refund, Capture, Cancel

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

2.1 Installation Process

This document contains all information necessary to install the module. Make sure to follow the described steps strictly in order to ensure the safe use of the module in compliance with all security regulations.

1. Configuration of the PAYONE test administration surface. The test platform can be found under https://pmi.pay1.de/
2. Configuration of the main settings
3. Configuration of the payment methods
4. Conducting a test order by means of the Test Data enclosed at the end of this document

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 PAYONE might not get through to the shop.

2.2 System Requirements

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

- PHP Version: 5.4.x or higher
- PrestaShop Version: 1.6.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.

3 PAYONE - Backend Configuration

First, log on to the PAYONE Merchant Interface (https://pmi.pay1.de/)

3.1 Payment Portal Setup

Under **Configuration > Payment Portals** you can set up the payment portal for your shop. Among other things you define the TransactionStatus URL. This is the URL that the parameters for further processing are sent to after payment has taken place.

3.1.1 Shop Portal Administration: Tab General

General settings such as portal name, Shop URL and the sender e-mail can be defined in the tab 'General'. These settings can be defined independently.

Make sure that the Server API, Front-end and Client API are activated in the Channel settings as shown in the screenshot.

Figure 3.1: Tab 'General'

3.1.2 Shop Portal Administration: Tab Extended

In this tab you define the TransactionStatus URL. The URL you need to enter can be found directly on the general information site of the module in your shop. More information can be found in the section Transaction-Feedback).The remaining fields can be left empty. They will be filled by the module.
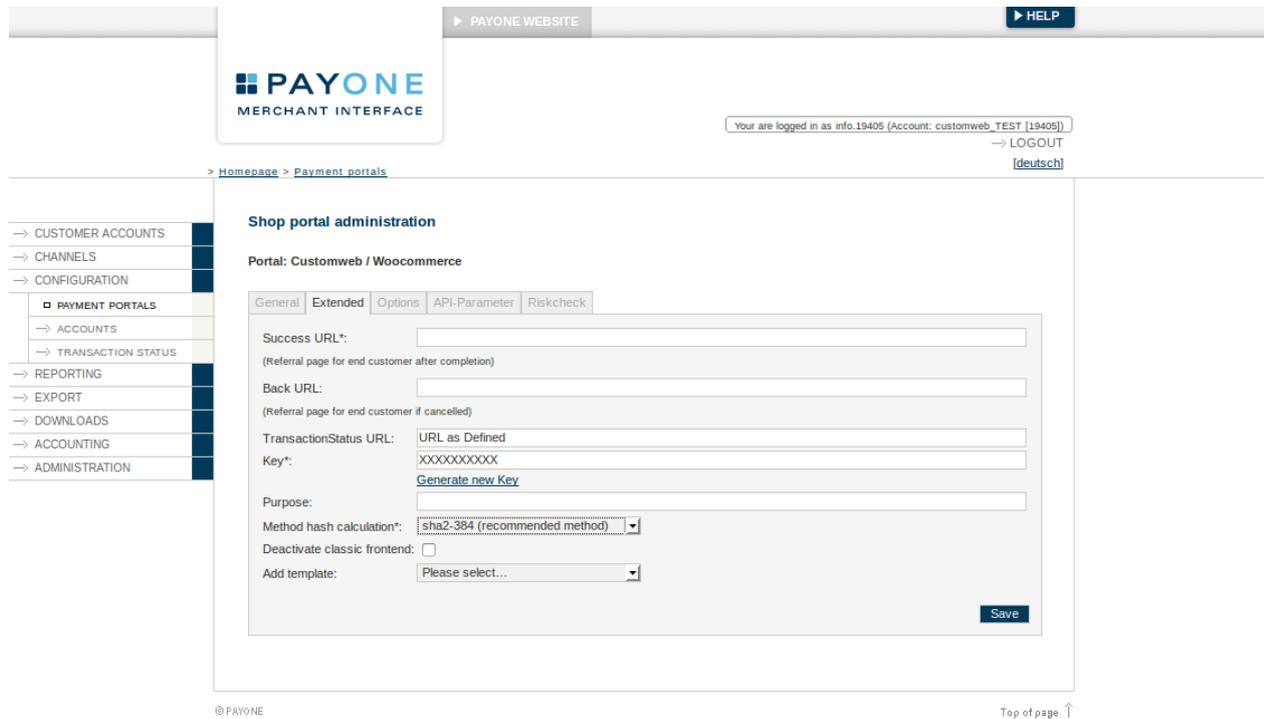
Figure 3.1: Extended configuration of the payment portal

3.1.3 Shop Portal Administration: API-Parameter

This tab contains all relevant information to enter in the Main Module .
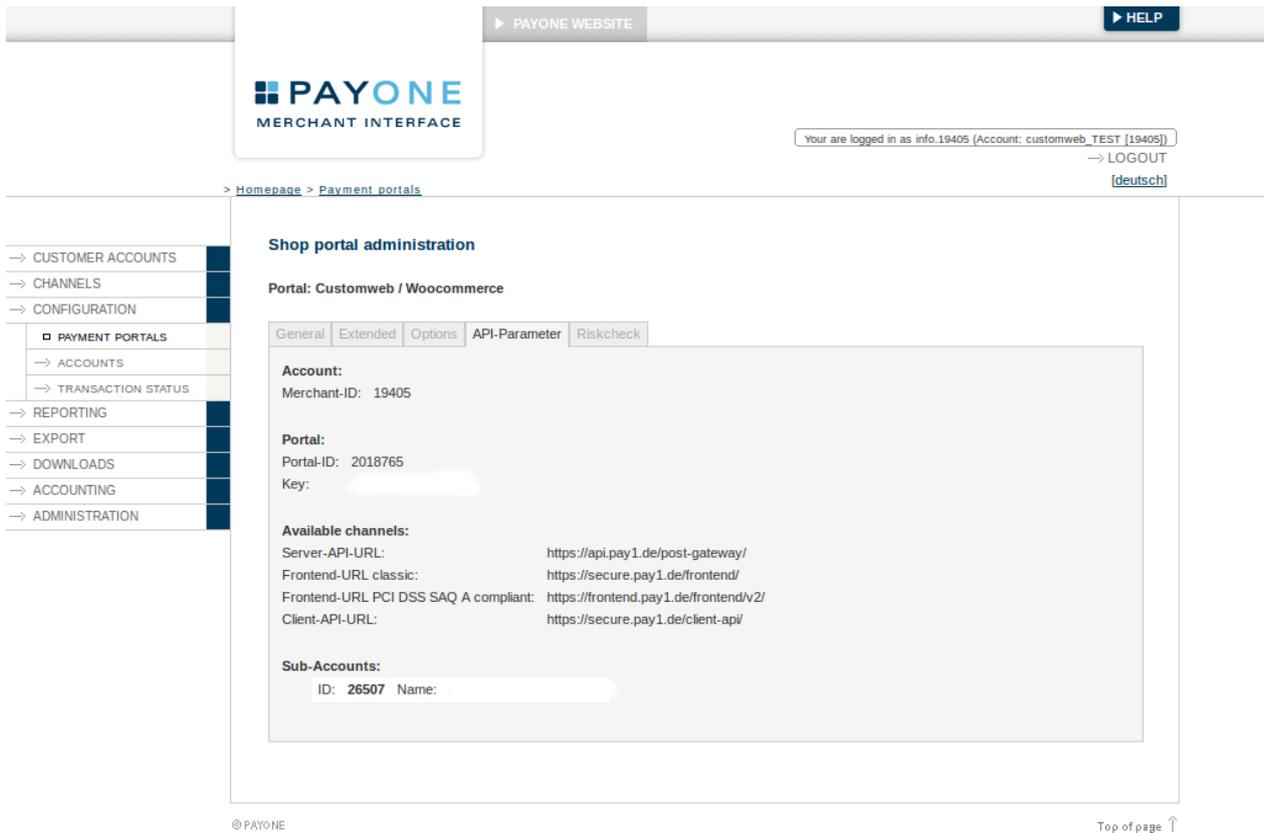
Figure 3.1: API-Parameter Configuration.

3.2 Authorization Methods

PCI 3.1 - SAQ A-EP

Please note that with the changes in PCI new rules apply regarding the applicable self assessment questionnaire. If you host the forms to entering of the credit cards directly in your webshop (Hidden Authorization) or Ajax authroization new rules apply. If you want to continue to work with SAQ-A, you have to use Payment Page or the Widget Authorization.

In the configuration of the payment method you have the possibility to switch between various authorization methods. These regard primarily the depiction of the credit card forms in the shop. You'll find further information about that in chapter authorization methods.

3.3 Account Activation

Further information on the payment method configuration and the implementations of the settings can be found here. After the activation and configuration of the payment methods you can conduct a test by means of the Test Data provided in this manual. As soon as the tests As soon as the tests have been successful you can contact PAYONE to demand the activation of your account. You will be able to process payments immediately.

4   Module Installation and Update in the PrestaShop Shop

4.1   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 sellxed shop (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.

4.2   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.

4.2.1 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.

4.2.2 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.

5 Module Configuration in the PrestaShop 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.

5.1 Configuration of the Main Module

You will find the settings for the module under **Modules**, in the group **Checkout , PAYONE Integration**. Install the module and open the configuration mask by clicking **Configure**. Enter the individual options such as described above in the configuration of the administration interface of PAYONE. You will find information on the individual options directly in the additional texts in the module

Do not forget to configure the main shop view first!

If you are using a multishop setup within PrestaShop, you will need to configure the payment plugin in the main shop view first. If you do not configure the settings of the plugin in the main shop view at all, the payment will not work properly. We strongly advise you to configure your main store first before moving on to the configuration of your subshops.

5.2 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 PAYONE configuration. The URL to be entered can be found in your PrestaShop Shop under: **Modules > PAYONE > Setup > Setup**

5.3 Configuration of Payment Methods (for PrestaShop 1.6)

After you have installed the main module you can install the corresponding payment methods. To do so you have to install the desired payment methods under **Modules** in the group of **Payments and Gateways**. You can save individual settings for each payment method and thereby optimally

adapt the payment to your processes. The most central are explained in more detail in this manual.

5.4 Configuration of Payment Methods (for PrestaShop 1.7)

After you have installed the main module you can install the corresponding payment methods. To do so you have to install the desired payment methods under **Modules > Module Catelog**. You can save individual settings for each payment method and thereby optimally adapt the payment to your processes. The most central are explained in more detail in this manual.

5.5 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 PAYONE and the module
>
> It may be that settings saved in the payment modules overwrite settings saved in PAYONE.

5.6 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.

5.6.1 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.

5.7 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 PAYONE during the authorisation process. The payment method is displayed in any case

5.7.1 Usage of the Integrated Multishop Functionality of PrestaShop

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

6 Settings / Configuration of Payment Methods

6.1 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 PAYONE 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.

6.2 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 PAYONE** or any similar pending status which is implemented by the module.

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

Orders with the status 'pending PAYONE' 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 PAYONE are being blocked. In this case check the settings of your firewall and ask the Hoster to activate the IPs and User Agents of PAYONE.

6.2.2 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.

7 The Module in Action

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

7.1 Capturing of Orders

> **Please note**
>
> The transaction management between your shop and PAYONE is not synchronized. If you capture payments with PAYONE, the status in the shop will not be updated and a second capturing in the shop is not possible.

In order to be able to capture orders you must make sure that you have set the option 'Capturing' to 'deferred' in the payment method configuration.

Transactions can be captured in different ways. The different options are explained below.

7.1.1 1. Via Orders > PAYONE Transactions

You can view all transactions in the transaction table under Orders > PAYONE transactions"

Figure 7.1: 1. View Transactions via Orders > PAYONE transactions.

Open the order and then click on the small magnifying glass in den transaction information overview. By clicking **Capture** button, you get into the following context.
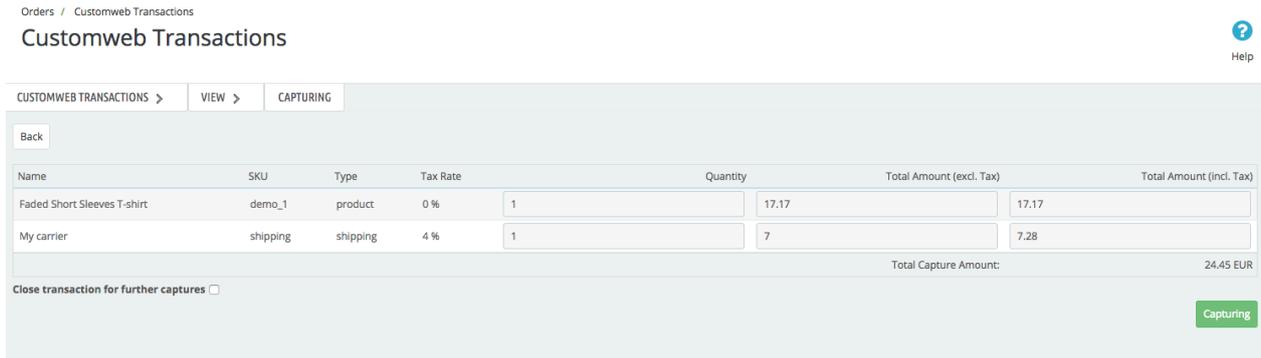
Figure 7.1: Invoice capturing in PrestaShop

Enter the amount and quantity that you wish to capture from the customer's authorization. By clicking **Capture** a direct capturing of the order occurs with PAYONE.

7.1.2 2. Capture the transaction through the order and the PAYONE transaction tab

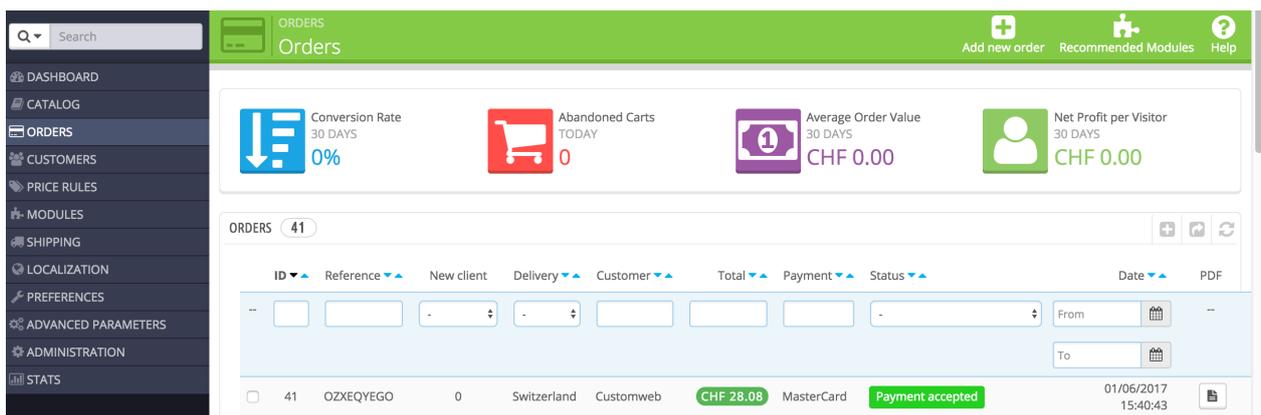Open the order and then click on the small magnifying glass in the PAYONE transactions overview.

Figure 7.1: Open order and view the transaction tab.

A window will then drop down. Enter the amount of your choice that you wish to capture from the customer's authorization. By clicking **Capture** a direct capturing of the order occurs with PAYONE.

Figure 7.1: 2. Transaction details in the transaction tab of a specific order.

By clicking **Capturing** button, you get into the following context.

Figure 7.1: Deferred/Manual capture of transactions within PrestaShop

Enter the amount or quantity choice that you wish to capture from the customer's authorization. By clicking **Capture** a direct capturing of the order occurs with PAYONE.

7.2 Useful Transaction Information on the Order

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 7.1: Transaction information within PrestaShop.

7.3 Using Invoice Details of a Processor

In the following context you can view or embed the "payment details" of for example an "Open Invoice" transaction:

7.3.1 PrestaShop Order Confirmation (E-Mail)

The "payment information" will be visible in the default "order confirmation e-mail" of PrestaShop.

7.3.2 PrestaShop Invoice (PDF)

The "payment information" will be visible in the default PrestaShop-Invoice.

7.3.3 PrestaShop-Backend (Transaction details)

You can view the transaction details in PrestaShop under **Orders > PAYONE Transactions**.

Figure 7.1: Transaction details within PrestaShop.

7.3.4 PrestaShop Success-Page

Figure 7.1: Payment information on the PrestaShop "Success-Page".

7.3.5 Payment Information in Email Template

If payment information is required for e.g. prepayment the merchant must manually change the files:

- mails/[language_code]/order_conf.html
- mails/[language_code]/order_conf.txt

The variables are defined as

```
{code}          payonecwpayment_information          payonecw_payment_information
payonecw_payment_information_txt {/code}
```

7.4 Place Orders in the backend of PrestaShop

With the PrestaShop 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. A window will open up. Search for the customer for whom you wish to place an order or manually create a new customer
3. Search for the product in question and add it to your cart
4. Before clicking **Create Order**, select the payment method
5. Depending on the authorization method that you have saved for the selected payment method, you will either be redirected to the Payment Page of PAYONE or the mask for the credit card will appear such as in the image below. Enter the customer's credit card information and click **Debit the customer**.
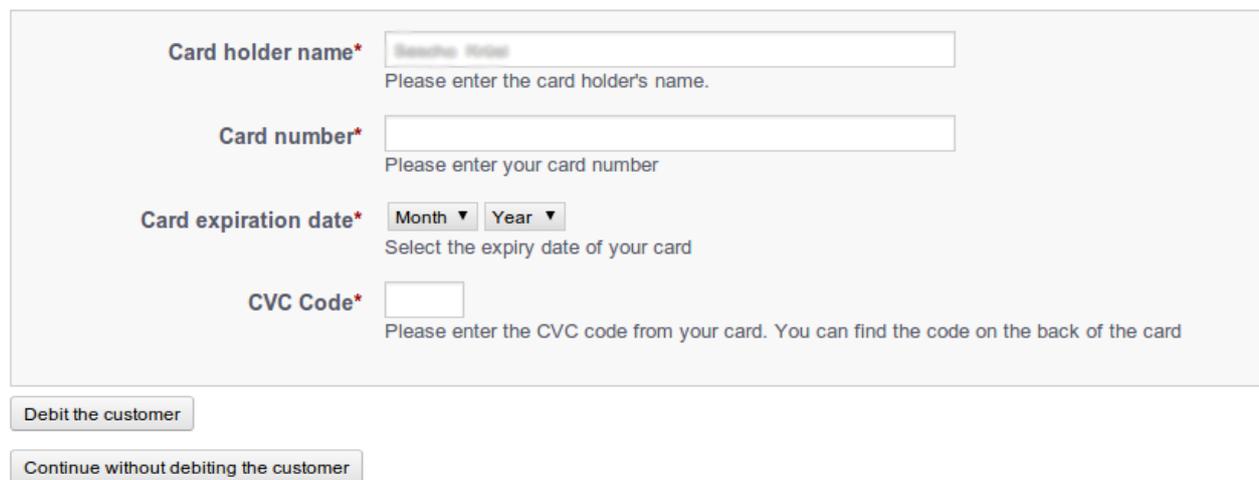
Figure 7.1: MoTo Transactions in PrestaShop

7.5 Refunds

You can refund already captured transactions and automatically transmit them to PAYONE. In order to to so, open the order. You have two alternatives for refunding the money. Either you use the refund tool or the partial refund process of PrestaShop. Both processes are explained in this passage.

7.5.1 1. Transactiontable via Orders > PAYONE transactions

You can view all transactions in the transaction table under "Orders > PAYONE transactions"

Figure 7.1: 1. transaction table via rrders > PAYONE transactions.

Open the order and then click on the small magnifying glass in den transaction information overview. By clicking **Refund** button, you get into the following context.

Figure 7.1: Refund of transactions within PrestaShop.

Just like for the capturing, you open the order and click on the magnifying glass next to the order of your choice. Enter any refund amount of your choice in the field provided for this purpose. By clicking **refund** the refund request will directly be sent to PAYONE and the specified amount will be credited on the customer's credit card.

Please note that you can't refund more than 100% of the original amount.

7.5.2 2. Refund the transaction through the order and the PAYONE transaction tab

Open the order and then click on the small magnifying glass in the PAYONE transactions overview.
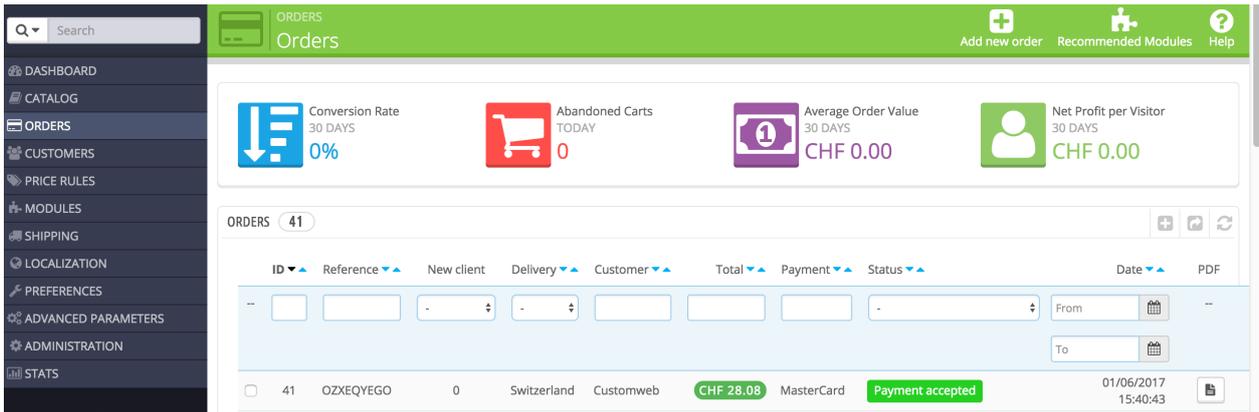
Figure 7.1: Open order and view the transaction tab.

By clicking **Refund** button, you get into the following context.
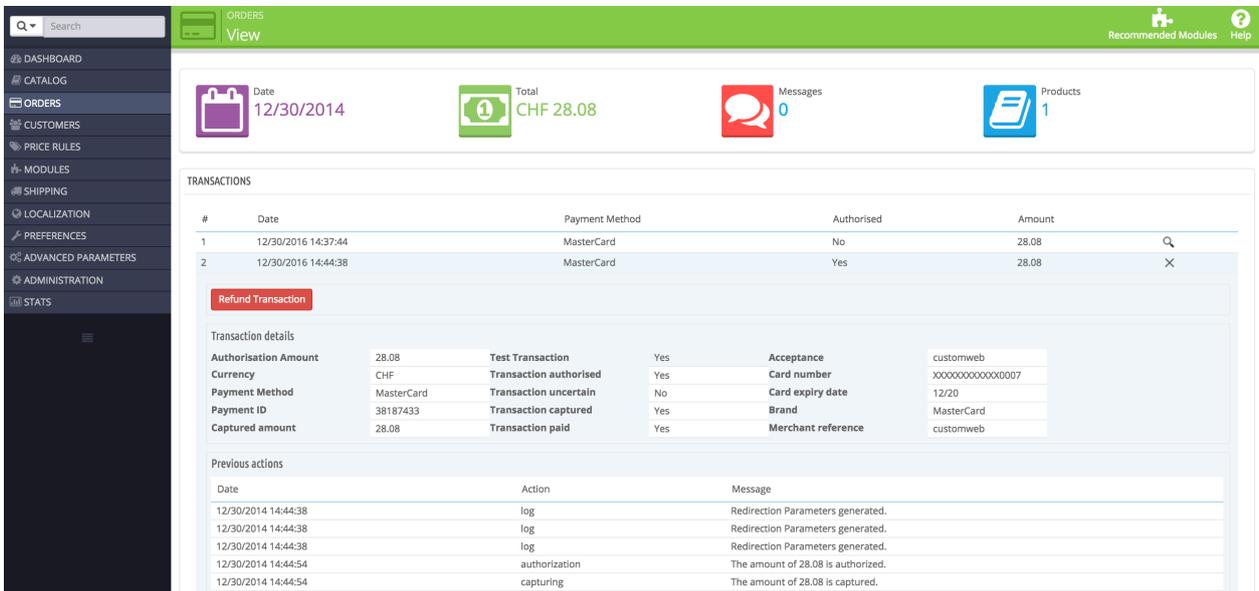
Figure 7.1: 2. transaction details in the transaction tab of a specific order.

By clicking **Refund** button, you get into the following context.
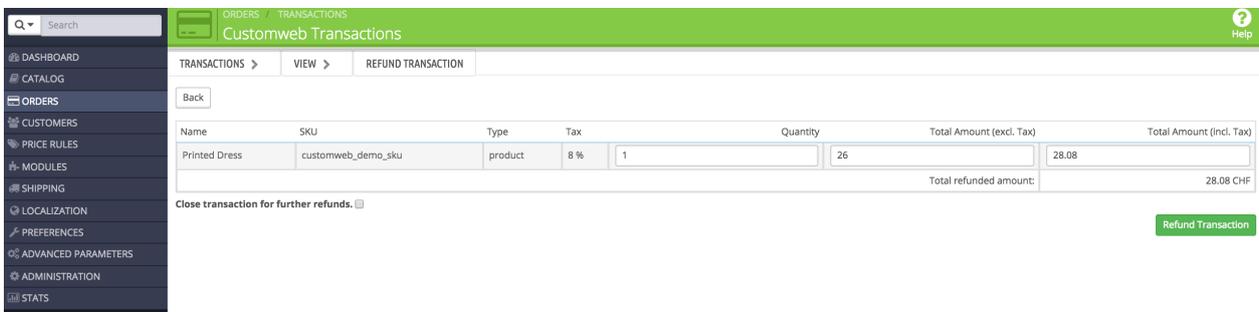
Figure 7.1: Refund of the transaction within PrestaShop.

Enter any refund amount of your choice in the field provided for this purpose. By clicking **refund** the refund request will directly be sent to PAYONE and the specified amount will be credited on the customer's credit card.

Please note that you can't refund more than 100% of the original amount.

7.5.3 3. Refund with the PrestaShop internal/default functionality.

You can carry out refunds with the Partial Refund Process of PrestaShop based on the product quantity. After having opened the order, click on **Partial Refund** in the top right corner. The browser will scroll down and you can specify the amount of the partial refund. If you wish, you can also directly add the item(s) to the stock. In order to do so, click on the corresponding check-box. By clicking **Partial Refund**, a new window opens up.
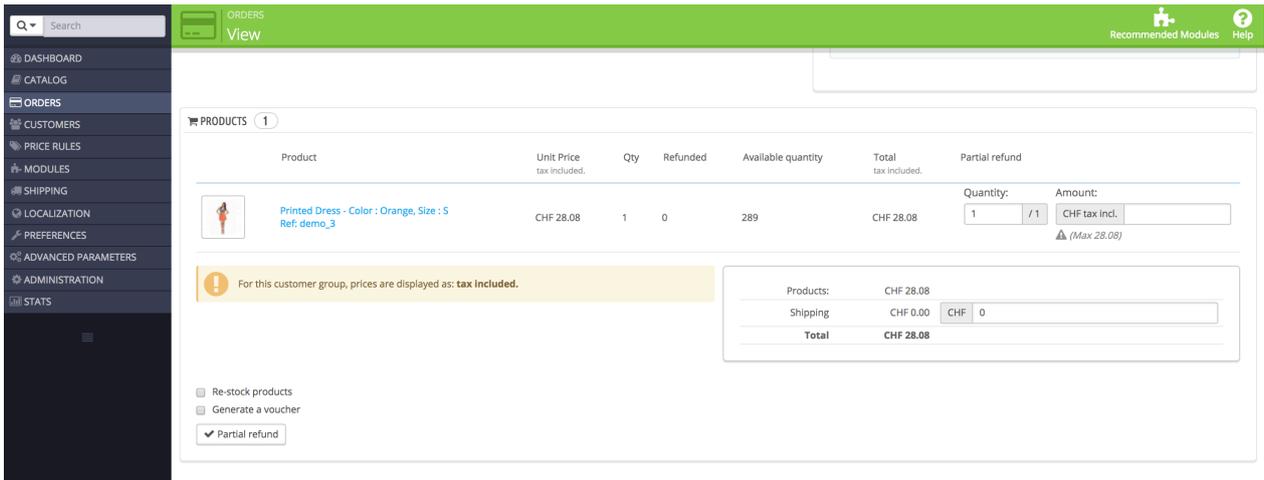
Figure 7.1: Refund of transactions within PrestaShop.

In the following window, click 'Yes' in order to transmit the refund to PAYONE and the specified amount will be credited to the customer's credit card.
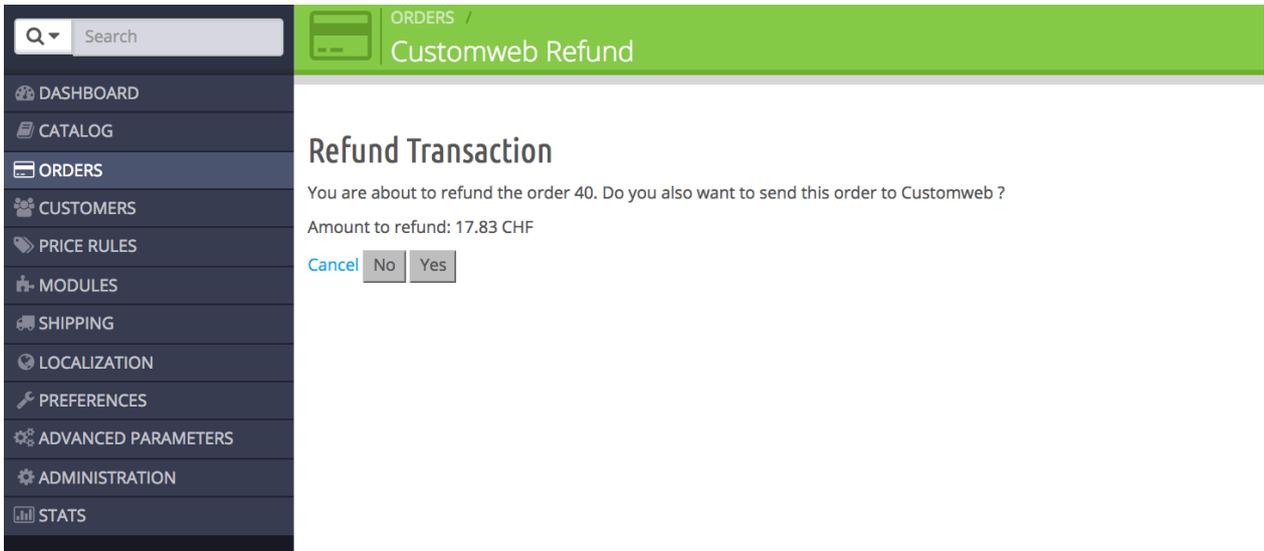
Figure 7.1: Confirm the transaction refund in PrestaShop.

Please note that you can't refund more than 100% of the original amount.

Executing a refund will not change the status of the order.

7.6 Usage of the Alias Managers / Token Solution

With the Alias Manager, your customers can securely save their credit cards with PAYONE 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 his or her saved credit cards without having to re-enter all the details.

Figure 7.1: Alias Manager Usage Within PrestaShop.

Please pay attention!

The usage of the Alias Managers requires the activation of the correct option with PAYONE. To do so, please contact the support directly.

7.7 One Page Checkout

The checkout of PrestaShop can be modified and reduced to one step. Our module support the standard OnePageCheckout of PrestaShop. No guarantee can be made in regards to the compatibility with third party modules.

You can activate the standard checkout by going to **Preferences, Orders**. Choose the option 'one-page checkout' in the drop-down menu for **Order-process type**.

7.8 Setting up Cron Job

In order to activate the time-controlled functions of the plugin (such as update service, deleting pending orders, etc.) set up a time-controlled request in your hosting provider backend using the URL which you will now find via Modules > PAYONE in the tab "Extended Information". Invoking it regularly will trigger the actions.

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

8 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.

8.1 Test Data

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

SOFORT

Sofort	
IBAN	DE85123456782599100003
BIC	TESTTEST
Bank account	2599100003
Bank code	12345678

giropay

Sofort	
IBAN	DE46940594210000012345
BIC	TESTDETT421
ID / PIN	sepatest1 / 12345
NR / TAN	MAS_Test / 123456

Direct Debits

Hidden Mandate	Accept
IBAN	DE00123456782599100004
BIC	TESTTEST

Hidden Mandate	Denied - Invalid BIC
IBAN	DE00123456782599100004
BIC	TESTTESX

Credit / Debit Card

VISA	Verified by Visa	Visa 3D
Card number	4012 0010 3714 1112	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	
3-D Secure Password	12345	

VISA	CH	Visa CH
Card number	4111 1111 1111 1111	
Expiry Date:	12/2020	

Name	Hans Muster	
CVC:	123	

VISA	DE	Visa DE
Card number	4111 1310 1011 1111	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	

VISA	AT	Visa AT
Card number	4111 1210 1111 1111	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	

MasterCard	MasterCard SecureCode	Mastercard 3D
Card number	5453 0100 0008 0200	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	
3-D Secure Password	12345	

Mastercard		Mastercard
Card number	5500 0000 0000 0004	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	

American Express		American Express
Card number	340 0000 0000 0009	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	1234	

JCB	JCB J/Secure	JCB 3D
Card number	3528 4501 3100 3315	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	
3-D Secure Password	12345	

JCB		JCB
Card number	3088 0000 0000 0009	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	

Maestro		Maestro
Card number	5000 0000 0000 0009	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	

Carte Bleue   Carte Bleue
Card number   4973 0100 0000 0004
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

Diners Club   Diners Club
Card number   30 0000 0000 0004
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

Discover   Discover Card
Card number   6011 1111 1111 1117
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

MasterCard

Mastercard   Mastercard
Card number   5500 0000 0000 0004
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

MasterCard   MasterCard SecureCode   Mastercard 3D
Card number   5453 0100 0008 0200
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123
3-D Secure Password 12345

Visa

VISA   CH   Visa CH
Card number   4111 1111 1111 1111
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

VISA   DE   Visa DE
Card number   4111 1310 1011 1111
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

VISA   AT   Visa AT
Card number   4111 1210 1111 1111
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

VISA   Verified by Visa   Visa 3D
Card number   4012 0010 3714 1112
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123
3-D Secure Password   12345

American Express

American Express   American Express
Card number   340 0000 0000 0009
Expiry Date: 12/2020
Name   Hans Muster
CVC: 1234

Diners Club

Diners Club   Diners Club
Card number   30 0000 0000 0004
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

Discover Card

Discover   Discover Card
Card number   6011 1111 1111 1117
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

Carte Bleue

Carte Bleue   Carte Bleue
Card number   4973 0100 0000 0004
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

JCB

JCB   JCB
Card number   3088 0000 0000 0009
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123

JCB   JCB J/Secure   JCB 3D
Card number   3528 4501 3100 3315
Expiry Date: 12/2020
Name   Hans Muster
CVC: 123
3-D Secure Password   12345

Maestro

Maestro		Maestro
Card number	5000 0000 0000 0009	
Expiry Date:	12/2020	
Name	Hans Muster	
CVC:	123	

Maestro Maestro

9 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

9.1 Modul Performance

Depending on your server specs it could be that the module results to performance issues in your store. In this case we suggest to do the following:

- Deactivate the PrestaShop Cache. We often saw already better results when the cache was disabled.
- Deactivate the PrestaShop displayBackOfficeHeader hook. For this go to > Modules > Hooks > and disable the hook PAYONE Integration under the **displayBackOfficeHeader** Menu.

In case you want to undo the changes above you will need to re-install the base module again.

Figure 9.1: Disable the PrestaShop Hook.

9.2 The Referrer URL appears in my Analytics Tool

When a customer and the notification are redirected via Header Redirection, the PAYONE 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.

10 Compatibility with Third-Party Plugins

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

10.1 Fee's and discount's within PrestaShop

To configure a PAYONE payment gateways based fee and discount you will need the following 3rd-Party plugin.

- Payment Fees

10.2 Birthday and gender in PrestaShop

For certain payment service providers it is necessary to check the birthday and the gender of a customer. PrestaShop does not check this by default. Information on how to enable those checks can be found here:

- Birthday
- Gender

11 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.

11.1 Log Levels

You can configure the log level in the PAYONE settings.

> If you're using Prestashop Version 1.5, make sure you FTP access to the server before activating the logging, as you may need to fix the write permissions on the log file.

- Off: Nothing is logged. (Default)
- Error: Logs unexpected errors only.
- Info: Logs extended information.
- Debug: Logs information helpful for debugging.

11.2 Log Location

In PrestaShop 1.5 the FileLogger is used. The log file is stored in the log folder of your installation. Please make sure the path exists and it is writable by the webserver. (Default Path: {shopRootDirectory}/log/)

In PrestaShop 1.6 and newer the default PrestashopLogger is used and are therefore stored in the database. The log message are visible in the PrestaShop backend under the menu item **Advanced Parameters > Logs**.

12 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.

12.1 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.

Have Prestashop initialized in your script, this is required for the database connection.

```
require(dirname(__FILE__).'/config/config.inc.php');
```

Include the module main file and modules Transaction class.

```
require_once _PS_ROOT_DIR_.'/modules/payonecw/payonecw.php';
require_once _PS_ROOT_DIR_.'/modules/payonecw/lib/PayoneCw/Entity
/Transaction.php';
```

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

```
$transactionById = PayoneCw_Entity_Transaction::loadById($transactionId);
$transactionObject = $transactionById->getTransactionObject();
```

Load transactions by Order ID:

```
$transactionsByOrderId = PayoneCw_Entity_Transaction::
getTransactionsByOrderId($orderId);
foreach($transactionsByOrderId as $transaction){
    $transactionObject = $transaction->getTransactionObject();
    //Do something with each object
}
```