

# Installation Manual for Unser PrestaShop

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

**Release Date:** Tue, 07 Dec 2021 09:12:59 +0100  
**Version:** 1.0.85

wallee AG  
General-Guisan-Strasse 47  
CH-8400 Winterthur

© copyright by wallee AG Tue, 07 Dec 2021 09:12:59 +0100  
####conditional####

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Procedure of the Installation .....	5
1.2	System Requirements .....	5
<b>2</b>	<b>Configuration .....</b>	<b>7</b>
2.1	Live Environment .....	7
2.2	Test Environment .....	7
<b>3</b>	<b>Module Installation and Update in the PrestaShop Shop .....</b>	<b>8</b>
3.1	Installation .....	8
3.2	Updates and Upgrades .....	8
3.2.1	Update Checklist .....	8
3.2.2	Update Instructions .....	9
<b>4</b>	<b>Module Configuration in the PrestaShop Shop .....</b>	<b>10</b>
4.1	Configuration of the Main Module .....	10
4.3	Configuration of Payment Methods (for PrestaShop 1.7) .....	10
4.4	Direct Capturing of Transactions .....	11
4.5	Uncertain Status .....	11
4.5.1	Setting the order state .....	11
4.6	Optional: Validation .....	11
4.6.1	Usage of the Integrated Multishop Functionality of PrestaShop .....	11
<b>5</b>	<b>Settings / Configuration of Payment Methods .....</b>	<b>13</b>
5.1	General Information About the Payment Methods .....	13
5.2	Information on Payment Status .....	13
5.2.1	Order status "pending" / imminent payment (or similar) .....	13
5.2.2	Order status "cancelled" .....	13
5.3	Invoice .....	14
5.4	Prepayment .....	14
5.5	Instalment .....	14
5.6	Secured Invoice .....	14
<b>6</b>	<b>The Module in Action .....</b>	<b>16</b>
6.1	Recommended Practice .....	16
6.2	Capturing of Orders .....	16
6.2.1	1. Via Orders > Unser Transactions .....	16

6.2.2	2. Capture the transaction through the order and the Unzer transaction tab ...	17
6.3	Useful Transaction Information on the Order .....	18
6.4	Using Invoice Details of a Processor .....	19
6.4.1	PrestaShop Order Confirmation (E-Mail) .....	19
6.4.2	PrestaShop Invoice (PDF) .....	19
6.4.3	PrestaShop-Backend (Transaction details) .....	19
6.4.4	PrestaShop Success-Page .....	20
6.4.5	Payment Information in Email Template .....	21
6.5	Refunds .....	21
6.5.1	1. Transactiontable via Orders > Unzer transactions .....	21
6.5.2	2. Refund the transaction through the order and the Unzer transaction tab ....	22
6.5.3	3. Refund with the PrestaShop internal/default functionality. ....	23
6.6	One Page Checkout .....	24
6.7	Setting up Cron Job .....	25
6.8	Partial Cancel .....	25
<b>7</b>	<b>Testing .....</b>	<b>26</b>
7.1	Test Data .....	26
<b>8</b>	<b>Errors and their Solutions .....</b>	<b>28</b>
8.1	Modul Performance .....	28
8.2	The Referrer URL appears in my Analytics Tool .....	29
<b>9</b>	<b>Compatibility with Third-Party Plugins .....</b>	<b>30</b>
9.1	Fee's and discount's within PrestaShop .....	30
9.2	Birthday and gender in PrestaShop .....	30
<b>10</b>	<b>Error Logging .....</b>	<b>31</b>
10.1	Log Levels .....	31
10.2	Log Location .....	31
<b>11</b>	<b>Advanced Information .....</b>	<b>32</b>
11.1	Transaction Object .....	32

# 1 Introduction

This manual explains the installation, configuration and usage of the payment module for PrestaShop and Unser.

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

- User name and password for the login to the backend of Unser
- PrestaShop payment module as ZIP file.
- Access data to your server and shop

Partial cancellations can only be processed in a limited manner via the shop backend. You may select the option "Close order" while capturing an order, then all items not part of the capture will be cancelled. Pure partial cancellations must be processed via the Insights of Unser.

The checkout must be served via https for the payment methods to be available.

## 1.1 Procedure of the Installation

In this document you will find all important information for the installation of the module. It is important that you strictly follow the checklist. Only this will guaranteed a secure usage of the plugin in correspondence with all security regulations.

1. Configuration of the test environment by means of the integration data from Unser. These can be found on the test platform under <https://sbx-insights.unzer.com/>
2. Configuration of the basic settings of the payment module
3. Configuration of the payment methods
4. Carrying out of a test purchase by means of the attached [test data](#) at the end of this document
5. If the test was successful, you can configure the live data in your shop. Log into the live environment with the obtained access data under: <https://insights.unzer.com/>

### .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 Unser might not get through to the shop.

## 1.2 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.

## 2 Configuration

### 2.1 Live Environment

You find the necessary credentials in your Unzer Account (<https://insights.unzer.com/>). You need the following:

- API Private Key
- API Public Key

Log into your Unzer account and go to "Settings > Configuration". There you'll find the necessary credentials.

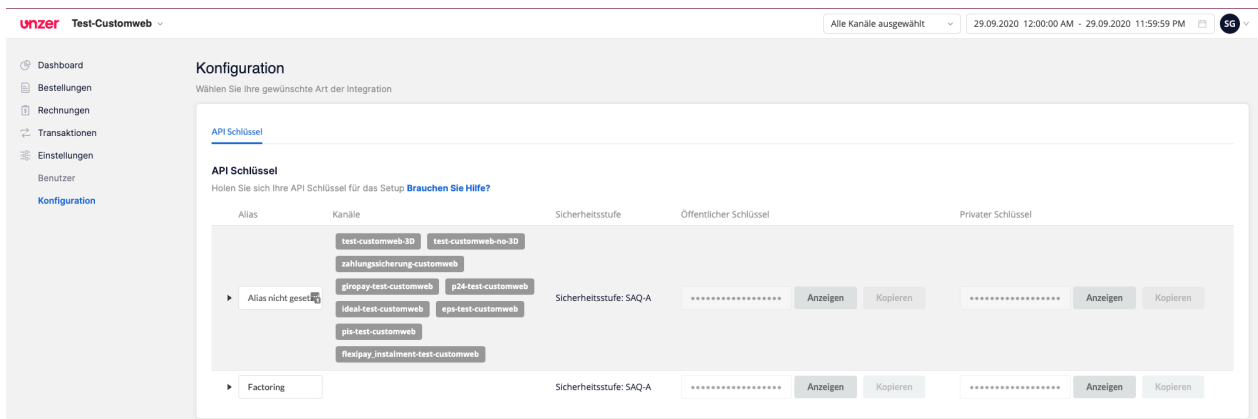


Figure 2.1: API Keys

Now enter those two keys into the main configuration of the plugin in your shop.

### 2.2 Test Environment

To login to the test environment you can use the following URL: <https://sbx-insights.unzer.com/>

## 3 Module Installation and Update in the PrestaShop Shop

### 3.1 Installation

At this time you should already be in possession of the module. In order to install the module in your shop, please carry out the following steps:

1. Unzip the archive you have just downloaded.
2. In the unzipped folder navigate to the folder "files"
3. 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.
4. 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.
5. If you haven't yet done so, log back into your shop.

### 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.

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.

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.

## 4 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.

### 4.1 Configuration of the Main Module

You will find the settings for the module under **Modules**, in the group **Checkout , Unser 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 Unser. 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.

### 4.2 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.

### 4.3 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 Catalog**. 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.

## 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 Unser and the module

It may be that settings saved in the payment modules overwrite settings saved in Unser.

## 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 Unser during the authorisation process. The payment method is displayed in any case

#### **4.6.1 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.

## 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 Unzer 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 Unzer** 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 Unzer' 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 Unzer are being blocked. In this case check the settings of your firewall and ask the Hoster to activate the IPs and User Agents of Unzer.

#### 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 Invoice

The fundamental concept of invoice is that the goods are shipped before the buyer pays the outstanding amount. That implies that the buyer has a limited time to carry out the payment to the merchant's bank account.

In case the buyer returns some of the items, you should create a refund. This does also work if the buyer has not yet paid the invoice. In this case, the outstanding amount will be reduced accordingly.

## 5.4 Prepayment

In case of prepayment, the buyer has to pay the invoice before the merchant ships the goods.

To realize this process, the shop needs to be informed about incoming payments. For this purpose the order is marked as uncertain until Unser notifies the shop about the successful payment via a background request. This leads to the uncertain state being lifted and is the trigger to go ahead and start the shipment process.

In case the customer decides not to order all items, you can create a refund. This is possible even before the buyer has paid the amount. In this case the outstanding amount is reduced so that the lower payment by the buyer will still trigger the shipment process by lifting the uncertain state.

## 5.5 Instalment

Instalment as a payment method allows the buyer to pay the outstanding amount in multiple slices.

Similar to invoice, the instalment payment method relies on the shipment to determine the due dates of the different slices. You can use the deferred capture operation to let Unser know when the shipment is ready. In case you like to reduce the amount, you can do this as part of the capture. The plugin will carry out corresponding cancel requests and send the delivery notification to Unser.

Although it is possible to trigger the delivery notification, called finalize, in Unser's Insights, we strongly recommend you to do it in the shop's backend.

In case you need to reduce the amount of the slices at a later point in time, you can do refunds.

## 5.6 Secured Invoice

The fundamental concept of invoice is that the goods are shipped before the buyer pays the outstanding amount. That implies that the buyer has a limited time to carry out the payment to the merchant's bank account.

This process has implications on to the shipment. In case the goods are not in stock, it is important not to capture the amount directly because with the capture the shipment notification is sent out and the due date is defined relatively to this point in time. In other words, you should configure the payment method to just authorize the payment and capture the

amount once the goods are ready to be shipped. This process is also recommended for other payment methods, if the goods are out of stock. In Unser Insights, the execution of the shipment notification is called finalize. The finalize operation is triggered with the capture. If only part of the amount is captured, the corresponding reduction will be applied automatically.

In case the buyer returns some of the items, you should create a refund. This does also work if the buyer has not yet paid the invoice. In this case, the outstanding amount will be reduced accordingly.

## 6 The Module in Action

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

### 6.1 Recommended Practice

Generally the most of the actions can be carried out from within the shop backend or from within the Unzer Insights. Not all actions that are triggered in Insights can be transferred correctly to the shop backend.

As such it is recommended to use the shop backend to carry out those actions (like refunds, cancels, captures etc.). This way both systems stay in sync.

### 6.2 Capturing of Orders

#### Please note

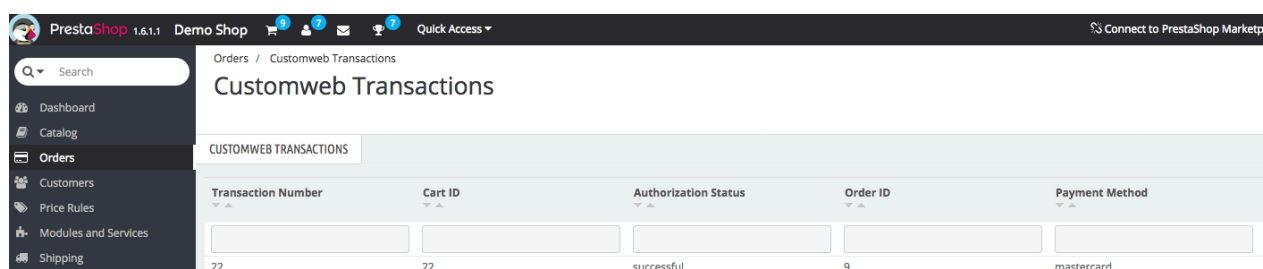
The transaction management between your shop and Unzer is not synchronized. If you capture payments with Unzer, the status in the shop will not be updated and a second capturing in the shop is not possible.

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.

#### 6.2.1 1. Via Orders > Unzer Transactions

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



Orders / Customweb Transactions				
CUSTOMWEB TRANSACTIONS				
Transaction Number	Cart ID	Authorization Status	Order ID	Payment Method
22	22	successful	9	mastercard

**Figure 6.1:** 1. View Transactions via Orders > Unzer 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.



Orders / Customweb Transactions

## Customweb Transactions

Help

CUSTOMWEB TRANSACTIONS > VIEW > CAPTURING

Back

Name	SKU	Type	Tax Rate	Quantity	Total Amount (excl. Tax)	Total Amount (incl. Tax)
Faded Short Sleeves T-shirt	demo_1	product	0 %	1	17.17	17.17
My carrier	shipping	shipping	4 %	1	7	7.28
Total Capture Amount:						24.45 EUR

Close transaction for further captures ☐

Capturing

**Figure 6.1:** 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 Unzer.

## 6.2.2 2. Capture the transaction through the order and the Unzer transaction tab

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

ORDERS Orders

Add new order Recommended Modules Help

Conversion Rate 30 DAYS 0%

Abandoned Carts TODAY 0

Average Order Value 30 DAYS CHF 0.00

Net Profit per Visitor 30 DAYS CHF 0.00

ORDERS 41

ID Reference New client Delivery Customer Total Payment Status Date PDF

41 OZXEYEGO 0 Switzerland Customweb CHF 28.08 MasterCard Payment accepted 01/06/2017 15:40:43

**Figure 6.1:** 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 Unzer.

**TRANSACTIONS**

#	Date	Payment Method	Authorised	Amount
150	01/06/2017 15:32:09	MasterCard	No	28.08
151	01/06/2017 15:32:15	MasterCard	No	28.08
152	01/06/2017 15:32:29	MasterCard	No	28.08
153	01/06/2017 15:32:48	MasterCard	Yes	28.08

**Transaction details**

Authorisation Amount	28.08	Transaction authorised	Yes	Card expiry date	12/18
Currency	CHF	Transaction uncertain	No	Brand	MasterCard
Payment Method	MasterCard	Transaction paid	Yes	Merchant reference	customweb
Payment ID	3014807004	Acceptance	customweb		
Test Transaction	Yes	Card number	XXXXXXXXXXXX9999		

**Previous actions**

Date	Action	Message
01/06/2017 15:32:48	log	Redirection Parameters generated.
01/06/2017 15:32:48	log	Redirection Parameters generated.
01/06/2017 15:32:48	log	Redirection Parameters generated.
01/06/2017 15:40:43	authorization	The amount of 28.08 is authorized.

**Figure 6.1:** 2. Transaction details in the transaction tab of a specific order.

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

Orders / Customweb Transactions

### Customweb Transactions

CUSTOMWEB TRANSACTIONS > VIEW > CAPTURING

Back

Name	SKU	Type	Tax Rate	Quantity	Total Amount (excl. Tax)	Total Amount (incl. Tax)
Faded Short Sleeves T-shirt	demo_1	product	0 %	1	17.17	17.17
My carrier	shipping	shipping	4 %	1	7	7.28

Total Capture Amount: 24.45 EUR

Close transaction for further captures ☐

**Capturing**

**Figure 6.1:** 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 Unzer.

## 6.3 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.

#	Date	Payment method	Is authorised	Amount	
46	2013-12-02 15:44:17	Master Card (Dynamic Template)	yes	124.58	✕
Transaction details					
Authorisation amount	124.58	Transaction authorised	Yes		
Currency	EUR	Transaction uncertain ⓘ	No		
Payment method	Master Card (Dynamic Template)	Transaction captured	Yes		
Payment ID	25815219	Transaction paid	Yes		
Captured amount	124.58	Alias	XXXXXXXXXXXX0007 (03/14)		
Alias Token	2403B927-26A3-4DAE-9EC3-CD000EAB349E				
Acceptance	test123				
Card number	XXXXXXXXXXXX0007				
Card expiry date	03/14				
Merchant reference	bc_ps_sell_46				

Figure 6.1: Transaction information within PrestaShop.

## 6.4 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:

### 6.4.1 PrestaShop Order Confirmation (E-Mail)

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

### 6.4.2 PrestaShop Invoice (PDF)

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

### 6.4.3 PrestaShop-Backend (Transaction details)

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

Orders / Customweb Transactions

Edit

GETTING STARTED WITH PRESTASHOP

Customize your shop Add products Configure payments Choose your shipping options

Take a tour: get started with PrestaShop

Hey Customweb, welcome on your own online shop. Follow the guide and take the first steps with your online shop!

No thanks! Let's start!

CUSTOMWEB TRANSACTIONS

Transaction ID 3

Transaction Number 3

Authorisation Status successful

Order ID 6 View

Created On 2017-03-07 17:08:27

Updated On 2017-03-07 17:08:50

Customer ID 2

Payment ID 3575510

Authorisation Amount 55.07

Currency CHF

Payment Method Open Invoice

Payment ID 3575510

Test Transaction Yes

Transaction authorised Yes

Transaction uncertain No

Transaction paid Yes

Payment Information Bank: Customweb Test Bank  
Account Holder: Customweb GmbH  
IBAN: DE2502400000TEST000000000004  
BIC: 9000  
Reference Number: BP3575510/2749

TRANSACTION HISTORY

Date Action Message

2017-03-07 17:08:50 authorization The amount of 55.07 is authorized.

CUSTOMER DATE

Customer ID 2

Billing Address Céline Fah  
Tödierrasse 17b  
CH-8004 Zürich  
Phone: 0041 44 111 22 33

Delivery address Céline Fah  
Tödierrasse 17b  
CH-8004 Zürich

Figure 6.1: Transaction details within PrestaShop.

## 6.4.4 PrestaShop Success-Page

SALE 70% OFF ALL PRODUCTS GET SAVINGS NOW

Call us now: 0123-456-789 Contact us English Sign out André Müller

Your Logo a new experience

Search

Cart (empty)

WOMEN DRESSES T-SHIRTS

Order confirmation

ORDER CONFIRMATION

01. Summary 02. Sign in 03. Address 04. Shipping 05. Payment

Order reference NHRCXPRKD

Amount 55.07 CHF

State Payment accepted

Date 03/07/2017

Payment Information Bank: Customweb Test Bank  
Account Holder: Customweb GmbH  
IBAN: DE2502400000TEST000000000004  
BIC: 9000  
Reference Number: BP3575510/2749

< View your order history

Figure 6.1: Payment information on the PrestaShop "Success-Page".

### 6.4.5 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

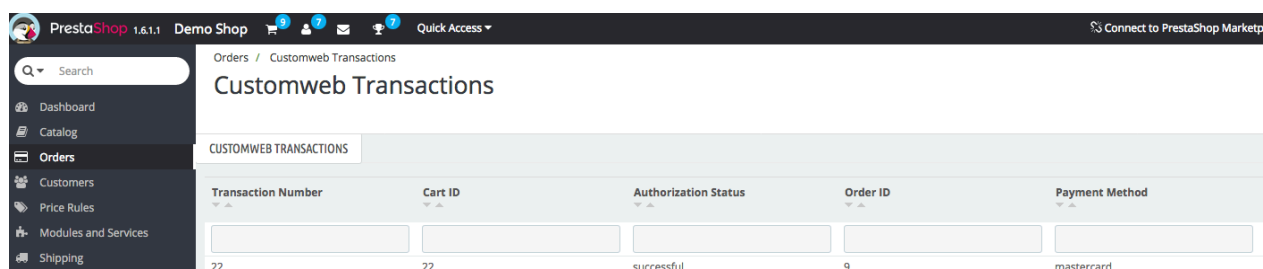
```
{code}          unzercwpayment_information          unzercw_payment_information
unzercw_payment_information.txt {/code}
```

## 6.5 Refunds

You can refund already captured transactions and automatically transmit them to Unzer. In order to do 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.

### 6.5.1 1. Transactiontable via Orders > Unzer transactions

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



Transaction Number	Cart ID	Authorization Status	Order ID	Payment Method
22	22	successful	9	mastercard

**Figure 6.1:** 1. transaction table via orders > Unzer transactions.

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

**Figure 6.1:** 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 Unser 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.

## 6.5.2 2. Refund the transaction through the order and the Unser transaction tab

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

**Figure 6.1:** Open order and view the transaction tab.

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

**Figure 6.1:** 2. transaction details in the transaction tab of a specific order.

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

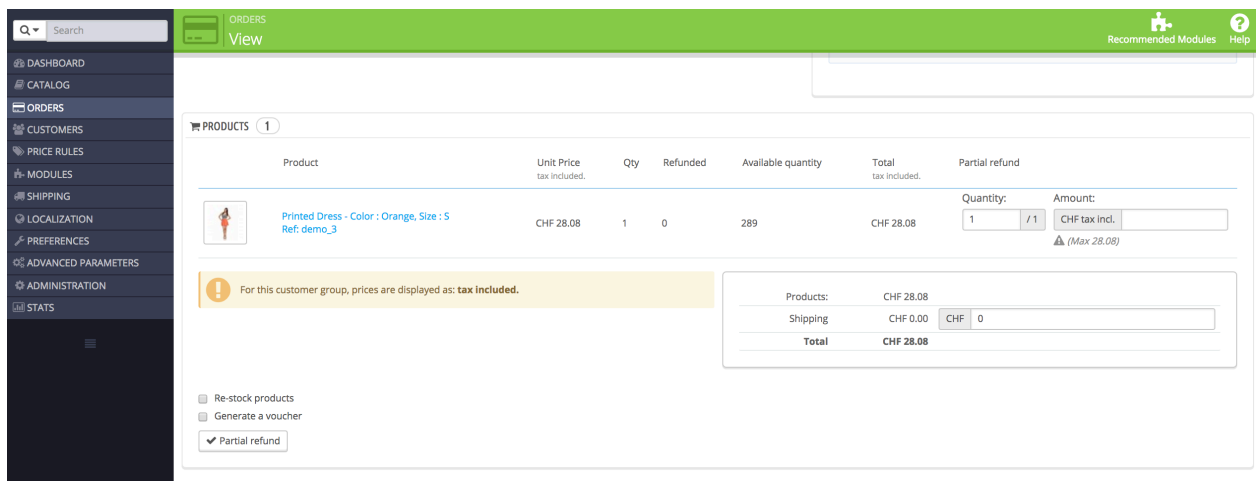
**Figure 6.1:** Refund of the transaction within PrestaShop.

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 Unser 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.

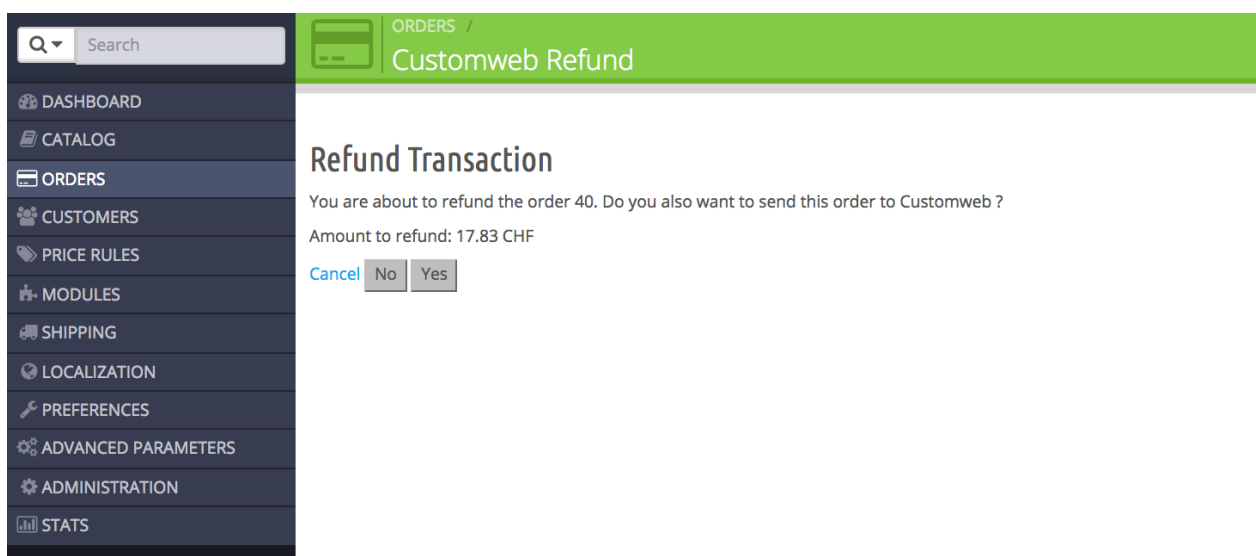
### 6.5.3 3. Refund with the PrestaShop internal/default functionality.

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 6.1:** Refund of transactions within PrestaShop.

In the following window, click 'Yes' in order to transmit the refund to Unser and the specified amount will be credited to the customer's credit card.



**Figure 6.1:** 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.

## 6.6 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**.



## 6.7 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 > Unser 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.

## 6.8 Partial Cancel

If you set the checkbox "close transaction for further captures" then the rest amount which has not been captured will be cancelled.

If you want to initiate a partial cancellation on an authorize transaction and allow further captures to be made, you must initiate the partial cancel from the Unser backend.

The transaction allows only refunds when the transaction has been closed for further captures.

## 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	4711 1000 0000 0000	Visa Success
Expiry Date	12/2030	
CVC	123	
3D PW	secret!33	
Card number	5453 0100 0005 9543	Mastercard Success
Expiry Date	12/2030	
CVC	123	
3D PW	secret3	

#### WeChat Pay

Username	keychain
Password	123

#### Alipay

Username	keychain
Password	123

#### EPS

Bank / Issuer	Stuzza Bank	EPS Simulation
Username	1003993	
Password	rX/'PvZzIW?&	

#### iDEAL

Account Bankname	ING_TEST
------------------	----------

#### giropay

Bank number	12345679
Account number	0000000300
IBAN	DE46940594210000012345
BIC	TESTDETT421
User	chiptanscatest2
User PIN	12345

User TAN	123456
TAN mechanism	optical Chip-TAN

**SOFORT**

Bank number	00000
Account number	123456
User PIN	123456
User TAN	12345

## 8 Errors and their Solutions

### 8.1 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 Unter Integration under the **displayBackOfficeHeader** Menu.

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

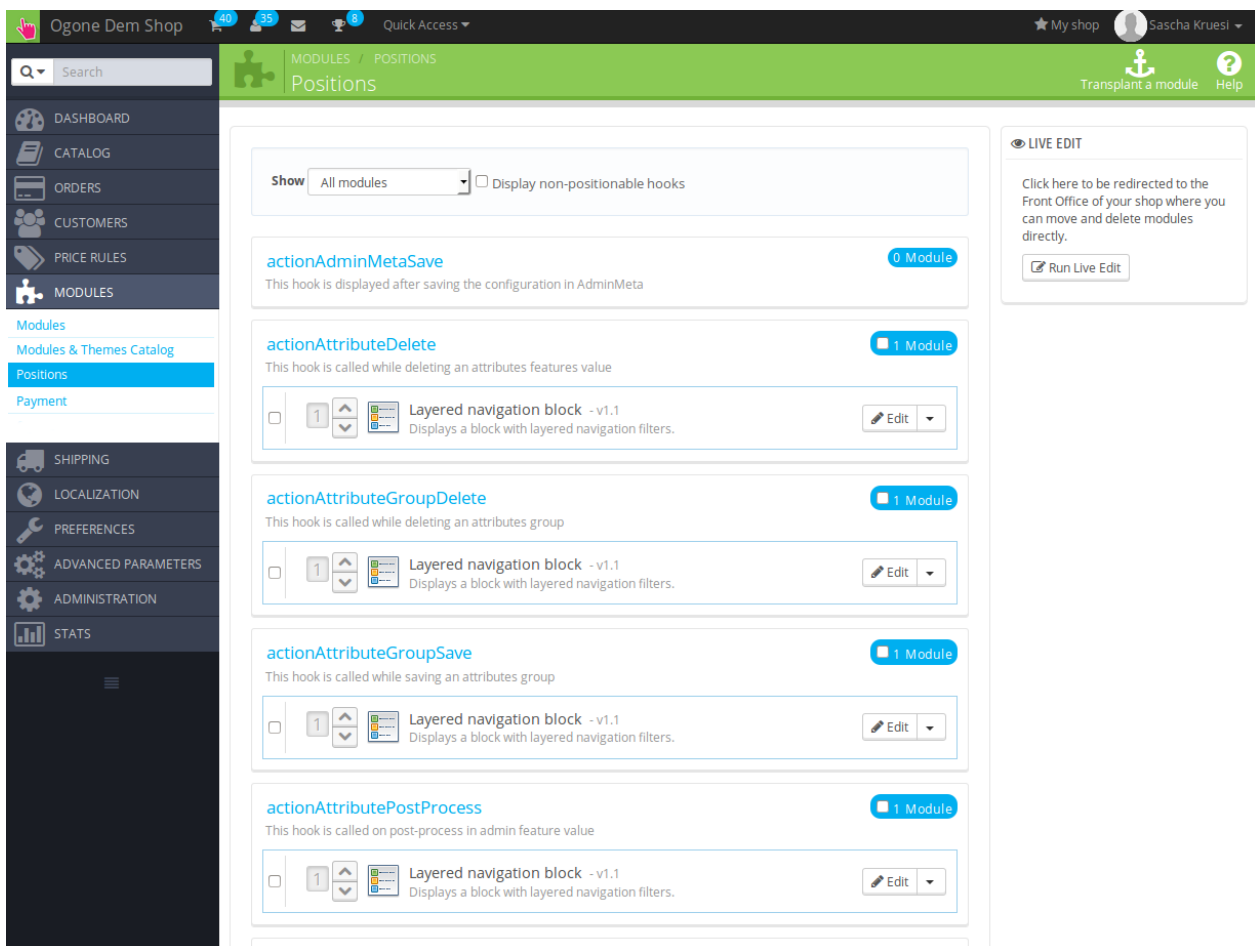


Figure 8.1: Disable the PrestaShop Hook.

## 8.2 The Referrer URL appears in my Analytics Tool

When a customer and the notification are redirected via Header Redirection, the Unzer 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 an easier way.

### 9.1 Fee's and discount's within PrestaShop

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

- [Payment Fees](#)

### 9.2 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](#)

## 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 Unzer 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.

### 10.2 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**.

## 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.

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/unzerCW/unzerCW.php';
require_once _PS_ROOT_DIR_.'/modules/unzerCW/lib/UnzerCw/Entity
/Transaction.php';
```

Now you can load the transaction and then extract the transactionObject.

Load the transaction by Id:

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

Load transactions by Order ID:

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