

# Installation Manual for Unzer e-Commerce

This manual describes the installation and usage of the Unzer extension for e-Commerce.

**Release Date:** Mon, 07 Aug 2023 14:01:06 +0200  
**Version:** 1.0.102

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

© copyright by wallee AG Mon, 07 Aug 2023 14:01:06 +0200  
####conditional####

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>5</b>
1.1	Procedure of the Installation .....	5
<b>2</b>	<b>Configuration .....</b>	<b>6</b>
2.1	Live Environment .....	6
2.2	Test Environment .....	6
<b>3</b>	<b>Module Installation and Update in the e-Commerce Shop .....</b>	<b>7</b>
3.1	Installation .....	7
3.2	Updates and Upgrades .....	7
3.2.1	Update Checklist .....	7
3.2.2	Update Instructions .....	8
<b>4</b>	<b>Module Configuration in the e-Commerce Shop .....</b>	<b>9</b>
4.1	Basic Module Configuration .....	9
4.2	Payment Method Configuration .....	9
4.3	Direct Capturing of Transactions .....	10
4.4	Uncertain Status .....	10
4.4.1	Setting the order state .....	10
4.5	Optional: Validation .....	10
<b>5</b>	<b>Settings / Configuration of Payment Methods .....</b>	<b>11</b>
5.1	General Information About the Payment Methods .....	11
5.2	Information on Payment Status .....	11
5.2.1	Order status "pending" / imminent payment (or similar) .....	11
5.2.2	Order status "cancelled" .....	11
5.3	Invoice .....	12
5.4	Prepayment .....	12
5.5	Instalment .....	12
5.6	Secured Invoice .....	12
<b>6</b>	<b>The Module in Action .....</b>	<b>14</b>
6.1	Recommended Practice .....	14
6.1.1	Manual booking on "Unzer Invoice Secured" .....	14
6.2	Useful Information on Transactions .....	14
6.3	Capture / Cancel Orders .....	14
6.3.1	Capturing and Cancelling Orders .....	16

6.4	Order Refunds .....	16
6.5	Transaction History .....	17
6.6	Using Invoice Details of a Processor .....	18
6.6.1	e-Commerce Order Confirmation (E-Mail) .....	18
6.6.2	e-Commerce Invoice (PDF) .....	18
6.6.3	e-Commerce-Backend (Transaction details) .....	18
6.6.4	e-Commerce Success-Page .....	19
6.7	Cron Job Setup .....	20
6.8	Partial Cancel .....	21
<b>7</b>	<b>Testing .....</b>	<b>22</b>
7.1	Test Data .....	22
<b>8</b>	<b>Errors and their Solutions .....</b>	<b>24</b>
8.1	The Referrer URL appears in my Analytics Tool .....	24
<b>9</b>	<b>Error Logging .....</b>	<b>25</b>
9.1	Log Levels .....	25
9.2	Log Location .....	25
<b>10</b>	<b>Advanced Information .....</b>	<b>26</b>
10.1	Transaction Object .....	26

# 1 Introduction

This manual explains the installation, configuration and usage of the payment module for e-Commerce and Unzer.

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 Unzer
- e-Commerce 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 Unzer.

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

**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

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 Unzer. 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 Unzer might not get through to the shop.

## 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.

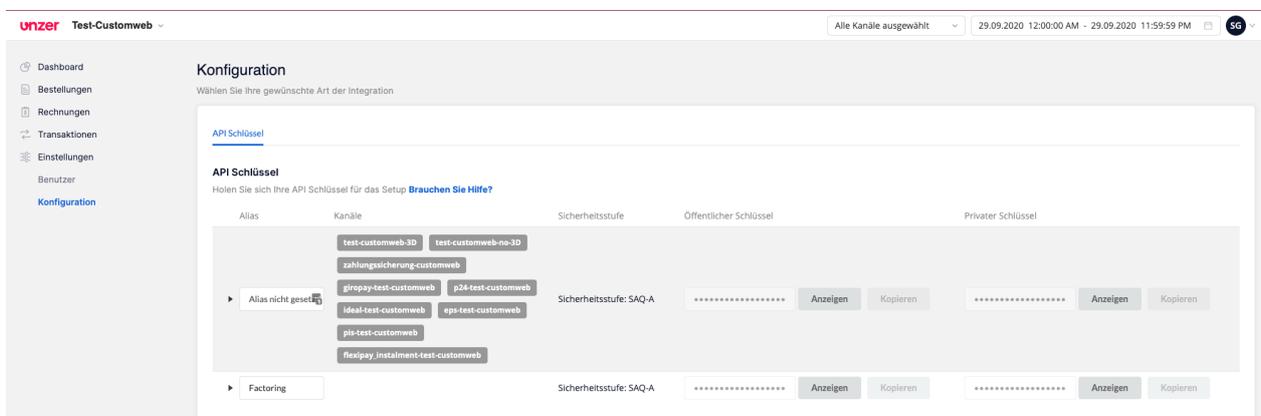


Figure 2.1: API Keys

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 e-Commerce 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 e-Commerce 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 Basic Module Configuration

Activate the plugin by selecting the module E-commerce Unzer in the **Plugin** menu.

The settings for the basic module can be found at **Unzer > Unzer**. Fill out the fields, you should have already entered the necessary data in the Unzer backend or you received it as part of your correspondence with Unzer. Each option is explained in more detail in a short info text in the shop.

### 4.2 Payment Method Configuration

After the basic module has been configured successfully, you can proceed to the payment settings. The payment methods you want to accept in your shop can be adjusted at **Settings > Store > Payments**. Every payment method is listed individually. Activate the payment methods you would like to provide in your shop by ticking the corresponding boxes on the right side. Individual changes can be made for each payment method to adjust it to your processes.

## 4.3 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 Unzer and the module

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

## 4.4 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.

### 4.4.1 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.

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

## 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 Unzer 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 Unzer 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 Unzer.

Although it is possible to trigger the delivery notification, called finalize, in Unzer'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.

#### 6.1.1 Manual booking on "Unzer Invoice Secured"

In the case of the "Unzer Invoice Secured" payment method, please mind to manually capture the transaction. This would change the transaction process in such a way that you would not be able to cancel such a transaction afterwards.

### 6.2 Useful Information on Transactions

For each order processed with our module you can access an overview of the most important transaction information. Among other things, this allows you to identify each corresponding order to each transaction displayed in the backend.

You can find the orders at Dashboard > Store Sales.

#	Transaction Number	Date	Payment method	Is authorised	Amount	Customer note	Add
8	26_8	2013-10-04 11:29:35	Saferpay Test Card	Yes	200.00		Less details
<b>Authorisation amount</b>				200.00			
<b>Currency</b>				EUR			
<b>Payment method</b>				Saferpay Test Card			
<b>Payment ID</b>				dzhzlbUOMIGAYWG618AIWKp5MA			
<b>Captured amount</b>				200			
<b>Refunded amount</b>				200			
<b>Transaction authorised</b>				Yes			
<b>Transaction uncertain</b>				No			
<b>Transaction captured</b>				Yes			
<b>Transaction paid</b>				Yes			
<b>Alias</b>				xxxx xxxx xxxx 0111			
<b>Card number</b>				xxxx xxxx xxxx 0111			
<b>Card expiry date</b>				12/2015			
<b>Card Type</b>				Saferpay Test Card			
<b>3D Secure</b>				Successful			
<b>Captures</b>							

Figure 6.1: Transaction information in WordPress E-Commerce.

## 6.3 Capture / Cancel Orders

### 6.3.1 Capturing and Cancelling Orders

In order to capture orders, open the order first (Dashboard > Store Sales) and then open the transaction information. Enter the amount you want to capture into the input field. By clicking on Capture the card is charged directly.

An order can be **cancelled** by clicking "Cancel Transaction" and the transaction as well as the reserved amount is released on the customer's card immediately.

#### Capturing Orders in the Unser Backend

The transaction management between Unser and your shop is not synchronised. When capturing payments in the Unser, the status is not adjusted in the shop and a second capture is not possible.

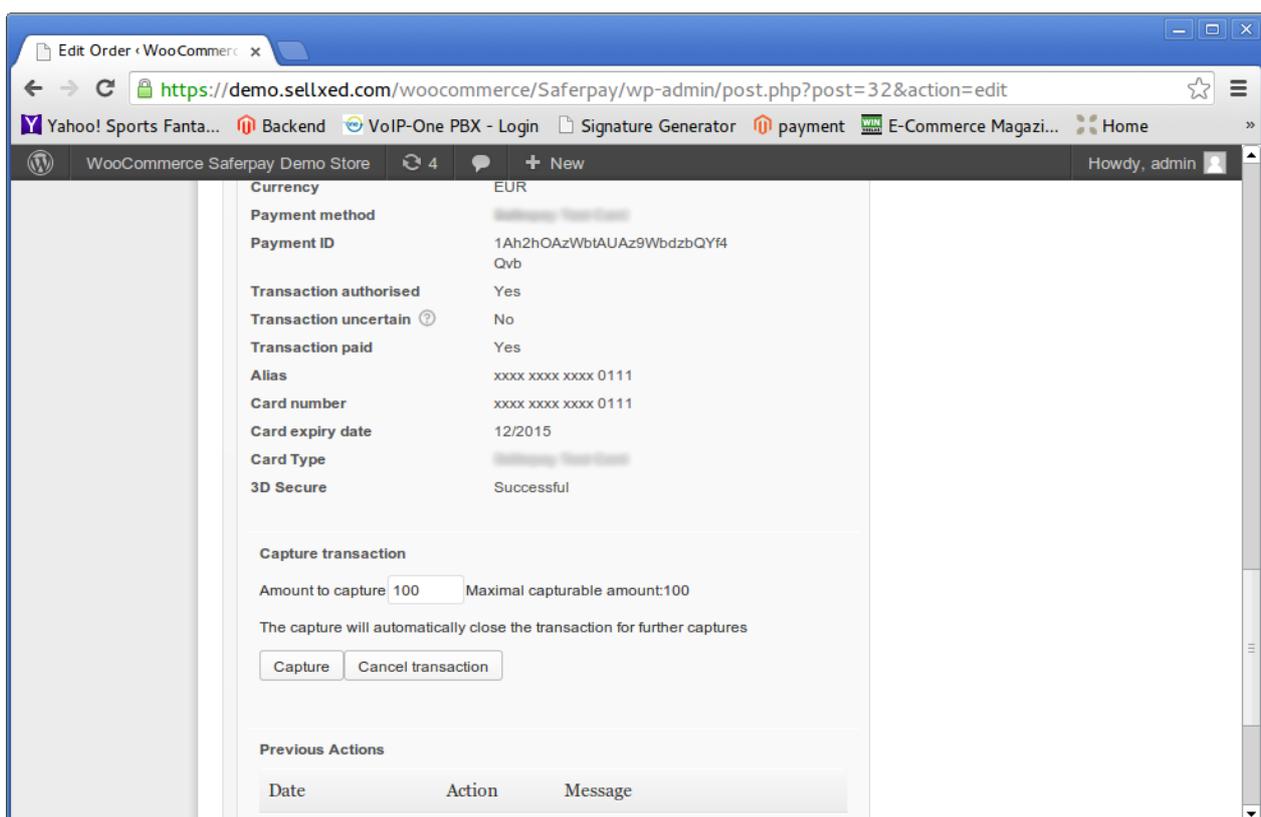


Figure 6.1: Cancelling and capturing orders

#### Partial Capturing

Please inquire, whether multiple captures are supported within your Unser contract. Otherwise, partially captured transactions could be blocked from future captures.

## 6.4 Order Refunds

In order to refund an order, open the transaction information (see above).

You can refund individual items or an arbitrary amount by adjusting the total amount or the number of items.

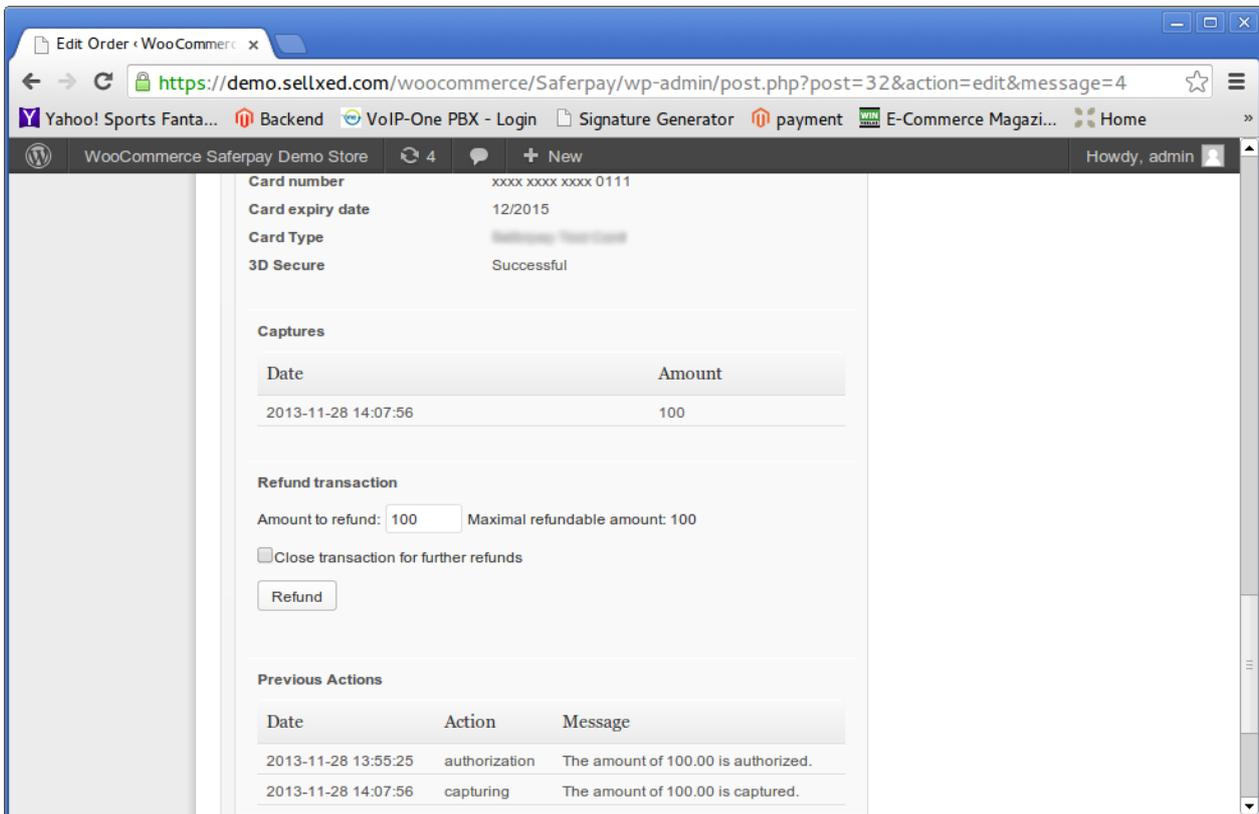


Figure 6.1: Refunds in WooCommerce for Unser.

### Maximum Refund

Our module restricts the refund amount to 100% of the originally authorised amount.

## 6.5 Transaction History

Every action within the transaction management is processed through the module. Information on each transaction can be viewed in the corresponding order.

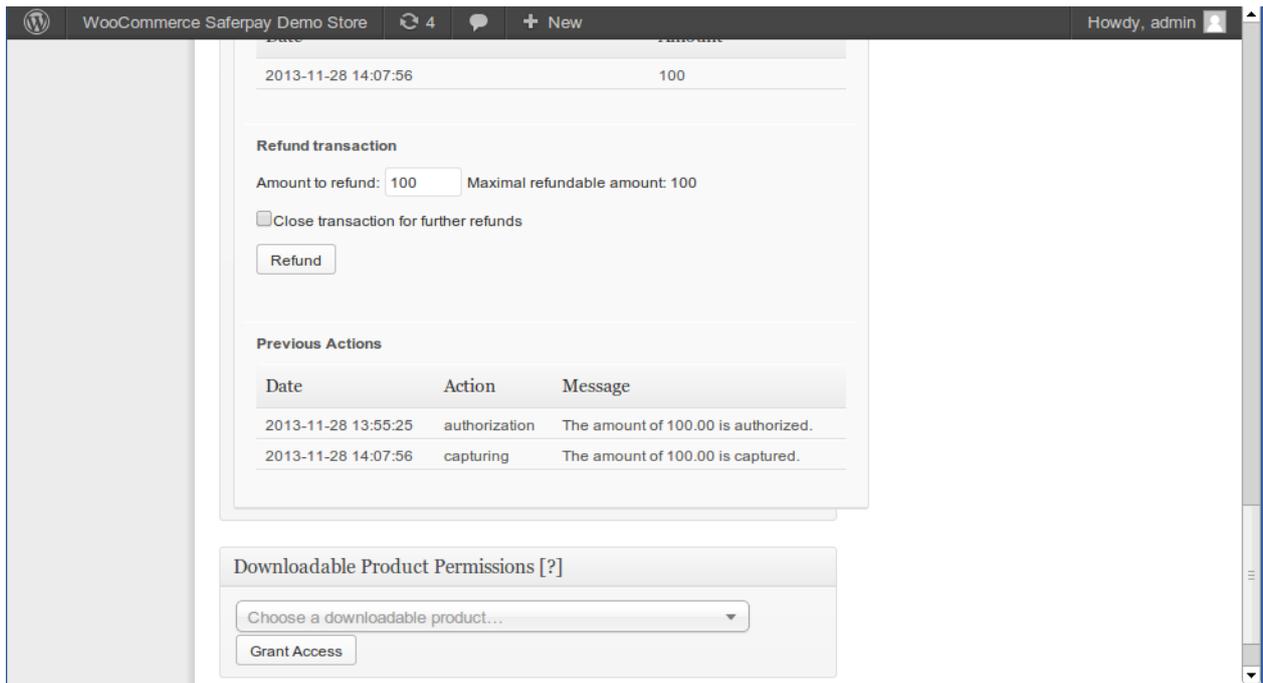


Figure 6.1: Transaction overview in the transaction history.

## 6.6 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:

### 6.6.1 e-Commerce Order Confirmation (E-Mail)

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

### 6.6.2 e-Commerce Invoice (PDF)

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

### 6.6.3 e-Commerce-Backend (Transaction details)

**Customweb Transactions**

#	Transaction Number	Date	Payment Method	Authorised	Amount
3	12	2017-03-07 15:12:01	Open Invoice	Yes	10.00

<b>Authorisation Amount</b> 10 <b>Payment Method</b> Open Invoice <b>Test Transaction</b> <span style="font-size: 0.8em;">?</span> Yes  <b>Transaction paid</b> Yes	<b>Currency</b> EUR <b>Payment ID</b> 3575423 <b>Transaction authorised</b> Yes <b>Transaction uncertain</b> <span style="font-size: 0.8em;">?</span> No
---	---

**Payment Information**  
 Bank: Customweb Test Bank  
 Account Holder: Customweb GmbH  
 IBAN: DE2501200000TEST000000000003  
 BIC: TESTBIC0003  
 Reference Number: BP3575423/2749

**Previous actions**

Date	Action	Message
2017-03-07 15:12:22	authorization	The amount of 10.00 is authorized.

**Figure 6.1:** Transaction details within e-Commerce.

### 6.6.4 e-Commerce Success-Page

Thank you. Your order has been received.

- Order Number: **12**
- Date: **March 7, 2017**
- Total: **€10.00**
- Payment Method: **Open Invoice**

## Order Details

Product	Total
<u>Test</u> × 1	€10.00 (ex. tax)
<b>Subtotal:</b>	€10.00 (ex. tax)
<b>Payment Method:</b>	Open Invoice
<b>Total:</b>	€10.00

## Customer Details

<b>Email:</b>	info@customweb.com
<b>Telephone:</b>	12345678

## Billing Address

André Müller  
 Schöneeggstrasse 2  
 8004 Zürich

## Payment Information

Bank: Customweb Test Bank  
 Account Holder: Customweb GmbH  
 IBAN: DE2501200000TEST000000000003  
 BIC: TESTBIC0003  
 Reference Number: BP3575423/2749

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

## 6.7 Cron Job Setup

In order to activate the time-controlled plugin functions (e.g. update service, deleting pending orders etc.) you have to set up cron jobs in e-Commerce. The update function is especially

relevant - it allows you to fetch subsequent transaction changes at Unzer via the API. Please keep in mind that additional options might have to be activated.

The module uses the default e-Commerce cron. Information on the setup can be found [here](#).

## 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 Unzer 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	chiptanscatetest2
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 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 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.

### 9.1 Log Levels

You can configure the log level in the Unzer settings.

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

### 9.2 Log Location

The log file is stored in the logs folder of this plugin **wp-content/plugins/plugins/ecommerce\_unzercw/logs**. Please make sure the path exists and it is writable by the webserver.

## 10 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.

### 10.1 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 Wordpress initialized in your script, this is required for the database connection.

```
require_once( dirname(__FILE__) . '/wp-load.php' );
```

Include the modules Util class.

```
require_once ABSPATH.'wp-content/plugins/ecommerce_unzerCW/classes/UnzerCw/Util.php';
```

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

Load the transaction by Id:

```
$transactionById = UnzerCw_Util::getTransactionById($transactionId);
$transactionObject = $transactionById->getTransactionObject();
```

Load the transaction by the external Id:

```
$transactionByExternal = UnzerCw_Util::getTransactionByExternalId($externalId);
$transactionObject = $transactionByExternal->getTransactionObject();
```

Load transactions by the order id:

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