Installation Manual for Unzer OpenCart

This manual describes the installation and usage of the Unzer extension for OpenCart.

Release Date: Thu, 04 Mar 2021 14:16:25 +0100
Version: 1.0.54

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

####conditional####

Table of Contents

1 Introduction

This manual explains the installation, configuration and usage of the payment module for OpenCart 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 log-in to the back-end of Unzer
- OpenCart 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 hIP of Unzer.

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 information important for the installation of the module. It is important that you strictly follow the check-list. Only by doing so, can a secure usage in correspondence with all security regulations be guaranteed.

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; otherwise the payment feedback of Unzer might not get through to the shop.

1.2 System Requirements

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

- PHP Version: 5.4.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.

In case you are using OpenCart version 3.0.3.5 or 3.0.3.6, you must patch your store to fix the twig extension error in order for modified templates to be loaded. The following extension can be used to resolve the issue: https://www.opencart.com/index.php?route=marketplace/extension/info&extension_id=40469.

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.

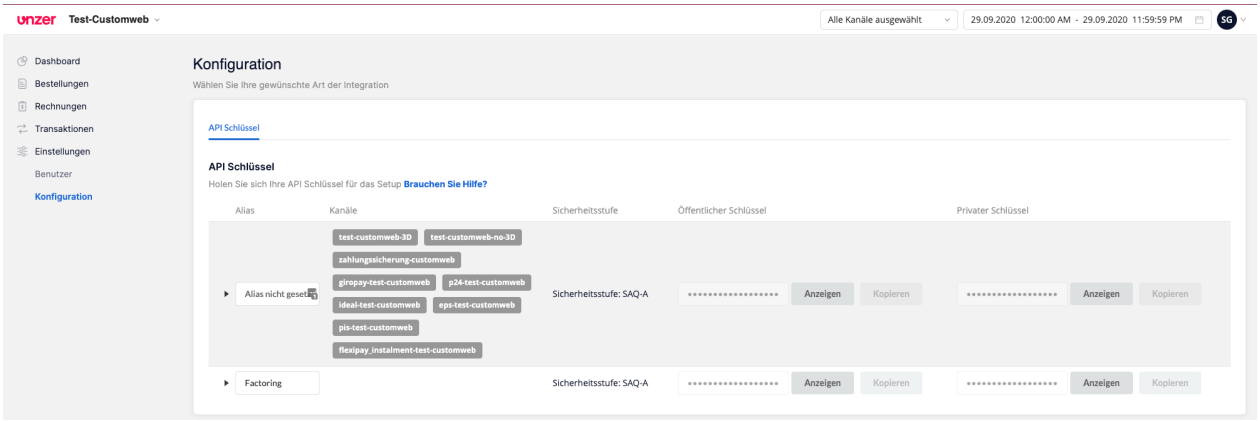


Figure 2.1: API Keys

Now enter those two key 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 OpenCart 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 shops 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 thats 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 back up 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. They can be found on the plugin page in the shop under the section change log on the lower end of the page. If there are no special remarks, you can proceed by just overwriting the files in your system

4 Module Configuration in the OpenCart 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 back-ups!

Please create a back-up 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 OpenCart 1.5.x / 2.0 / 3.0 Installation - Additions

All our modules are compatible with OpenCart 2.0. If you use OpenCart 2.0 upload all the files in the folder files_2.x to the root folder of your shopping cart. In case you are using OpenCart 1.5.x as basis you should use the folder files_1.x

In order to guarantee smooth operations and the usage of all features please make sure that you follow the instructions below.

5.1 Configuration of the Main Module

You will find the settings for the main module under **"Extension > Modules > Unzer Base Module"** . Install the module by clicking **Install**.

By clicking **Edit** you can configure the main module. Enter all data in the corresponding fields. The required data has either already been saved in the back-end of Unzer has been sent to you by Unzer. Each option is, furthermore, explained in short info texts in the shop.

5.2 Opencart 2.0 / 3.0 - Refresh Modifications

In case you are using OpenCart 2.0 you will have to refresh the modifications. Go to **Module > Modifications**. and click on the refresh button in the top right corner. Make sure that the folder system > modification is writable.

5.3 OpenCart 1.0 Specifications

In case you are using OpenCart 1.0 you should make sure that you use VQMOD in order to be able to use the whole set of functionalities. More Information about the project and the installation can be found under: https://github.com/vqmod/vqmod/wiki/Installing-vQmod-on-OpenCart

5.4 Configuration of the Payment Module

After having successfully configured the main module, you can find the settings for the individual payment methods in your shop under **Extensions > Payments** . Each payment method is listed individually. Install the payment methods you wish to offer to your customers. You can carry out individual settings for each payment method and thereby optimally adapt the payment methods to your existing processes. The most central options are described in more detail further below.

By clicking on **Install** the payment method is activated in your shop. Click **Edit** in order to modify the configuration of the payment method.

5.5 Shop ID

In the case of a multi-store set up (cf. chapter: Multi-Store), the parameter Shop ID identifies the shop's request and, therefore, redirects the feedback of Unzer based on the shop ID.

5.6 Order Prefix

With the option order prefix you can define your own order scheme for the transmission to Unzer. This option helps you identify to which shop a transaction the back-end of Unzer is related to. The tag "{id}" will automatically be replaced be the consecutive order number.

5.7 Selecting the Authorisation Method

You can choose between two options of authorizing credit card payments. By selecting one of the two under the option 'Authorisation Methods', you define how you want to process the individual payment methods.

> ### Payment Processing
>
> Please pay attention to the fact that you require the activation of further options with Unzer in order to use some of the authorisation methods.
>
> It may be that certain authorization methods are not available for every payment method.
>
> Alongside the introduction of PCI 3 there are additional certification requirements in case you are using the authorization method Hidden or AJAX. Please contact Unzer for further information or switch to another authorization method.

5.8 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 the debit the payment at a later point.

Depending on your acquiring contract, a reservation is only guaranteed for a specific amount 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.

5.9 Uncertain Status 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.

5.9.1 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.

5.10 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. 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 he or she 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

5.11 Multi-Shop Set-Up

The payment module is designed for the usage of one Unzer contract for several shops. It must be differentiated, however, between the case of an integrated multi-shop functionality and independent shops:

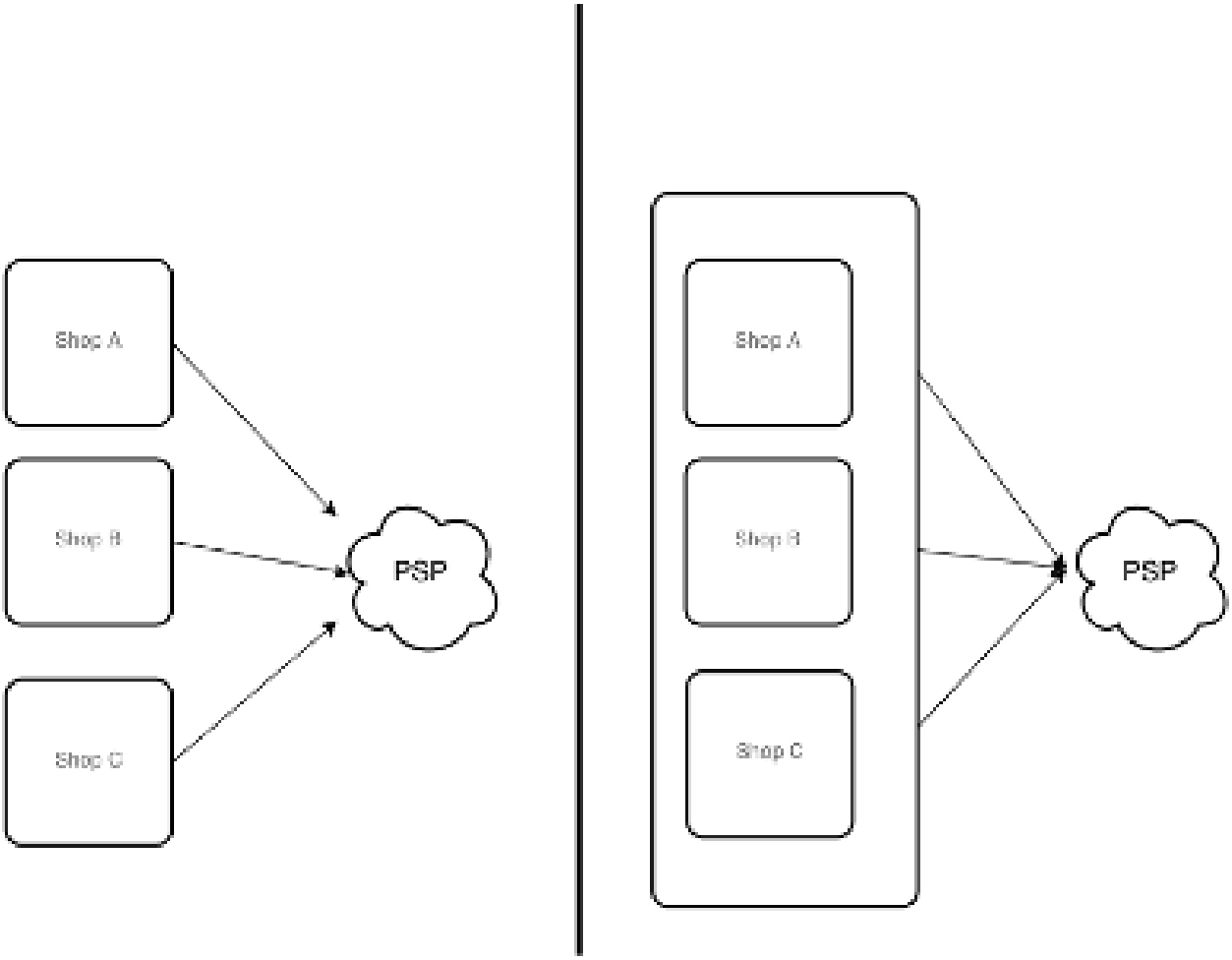


Figure 5.1: Multi-shop set-up

Licensing

In the case of independent shops, you require a payment module license for each shop. For scale prices, please contact us directly.

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 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.

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 Unzer** 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 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.

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.

6.3 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.

6.4 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.

6.5 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.

6.6 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 Unzer's 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.

7 The Module in Action

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

7.1 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 are reflected directly within the shop system.

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.

7.2 Useful Transaction Information on the Order

You can find an overview of the most important transaction information in all orders that have been processed by our module. Among others, this information allows for the definite attribution of the orders to their corresponding transaction, seen in the back-end of Unzer.

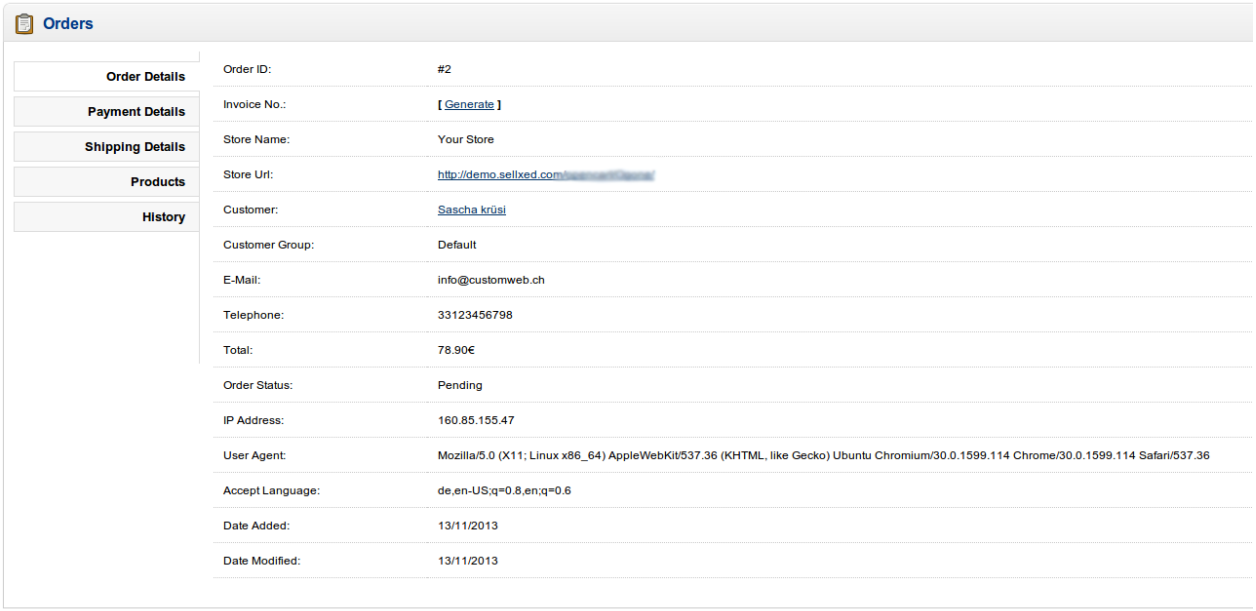


Figure 7.1: Transaction Information in OpenCart.

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 OpenCart Order confirmation (E-Mail)

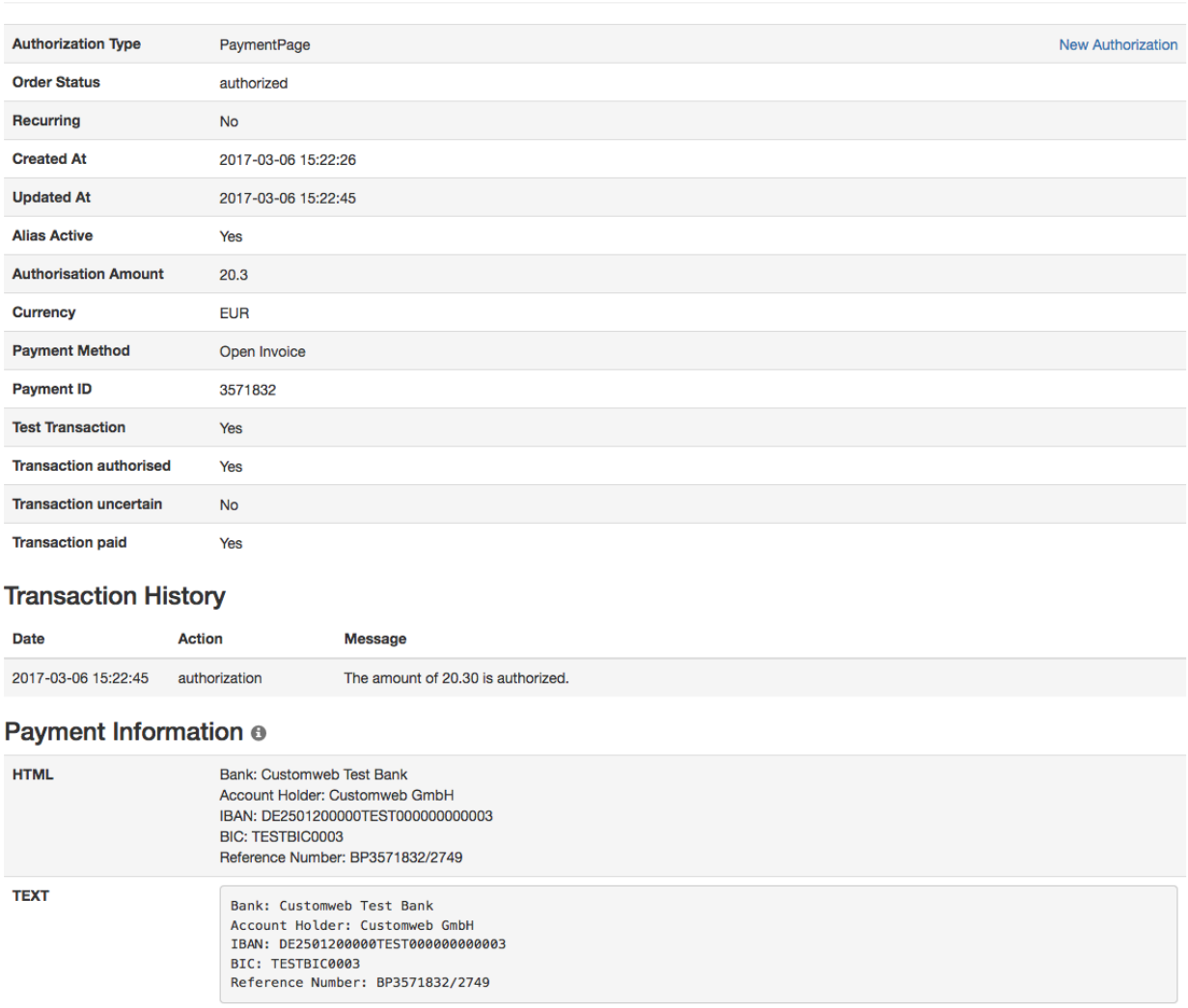
The "payment information" will be visible in the default "order confirmation e-mail" of OpenCart.

7.3.2 OpenCart Invoice (PDF)

This feature is not available by default in OpenCart that`s why it can not be supported by the payment gateway module.

7.3.3 OpenCart-Backend (Transaction details)

You can view the payment and transaction details in OpenCart under **Sales > Unzer Transactions**. If you can not find this menu point you have to clear the OpenCart modifications under **Extensions > Modifications >**.



Authorization Type	PaymentPage	New Authorization
Order Status	authorized	
Recurring	No	
Created At	2017-03-06 15:22:26	
Updated At	2017-03-06 15:22:45	
Alias Active	Yes	
Authorisation Amount	20.3	
Currency	EUR	
Payment Method	Open Invoice	
Payment ID	3571832	
Test Transaction	Yes	
Transaction authorised	Yes	
Transaction uncertain	No	
Transaction paid	Yes	

Transaction History

Date	Action	Message
2017-03-06 15:22:45	authorization	The amount of 20.30 is authorized.

Payment Information

HTML	Bank: Customweb Test Bank Account Holder: Customweb GmbH IBAN: DE25012000000TEST000000000003 BIC: TESTBIC0003 Reference Number: BP3571832/2749
TEXT	Bank: Customweb Test Bank Account Holder: Customweb GmbH IBAN: DE25012000000TEST000000000003 BIC: TESTBIC0003 Reference Number: BP3571832/2749

Figure 7.1: Transaction details within OpenCart.

7.3.4 OpenCart Success-Page

Due to technical limitations, it is currently not possible to display the "payment information" here.

7.4 Capturing / Cancelling of Orders

7.4.1 Capturing Orders

In order to capture orders, open the transaction manager under Sales > Unzer Transactions. Here you will find an overview over all transactions. Search for the order in the field with the order numbers. By clicking "view" you will open the transaction of the order.

7.4.1.1 Capturing Complete Orders or Partial Capturing

By clicking the button "Capture Transaction" a new window opens up. You can now capture every item individually in case you do not wish to send all items at once. The amount of possible partial captures depends on your contract. Please contact Unzer directly in order to clarify questions.

> **Capturing of Orders in the Back-End of Unzer**
>
> The transaction management between your shop and Unzer is not synchronised. 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 case you do not wish to capture all items of an order, you can close the transaction by clicking the box.

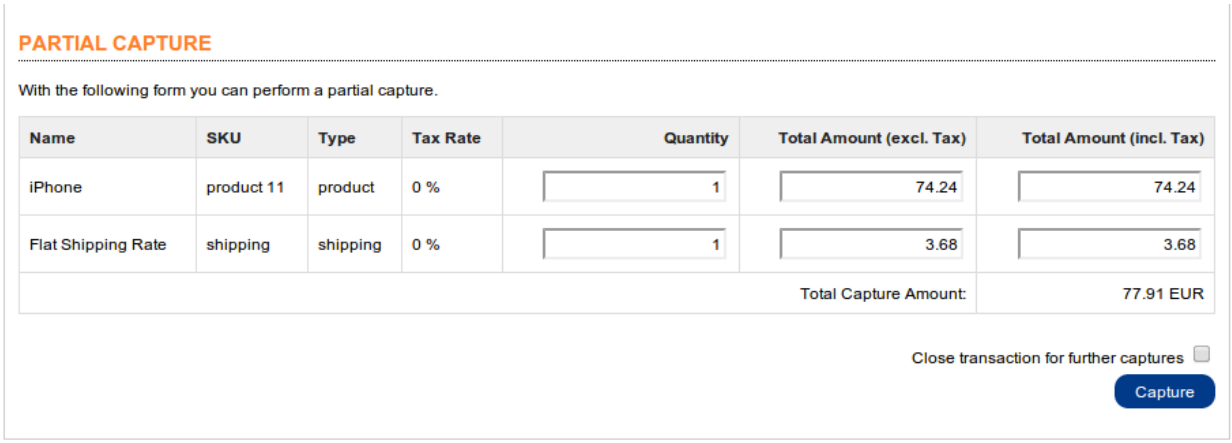


Figure 7.1: Capturing of Orders

> ### Partial Capturing
>
> Please find out if the capturing of partial amounts is supported by your Unzer contract. If this is not the case, it might happen that the transaction is closed for further transactions after a partial capture.

7.4.2 Cancel Orders

By clicking "Cancel Transaction", the transaction is **cancelled** and the reserved amount on your customer's card is released automatically.

Figure 7.1: Capture or Cancel in OpenCart.

7.5 Refunding Orders

In order to refund orders, open the transaction information (cf. above).

You can refund individual items or any amount of your choice by modifying the total amount or the item quantity.

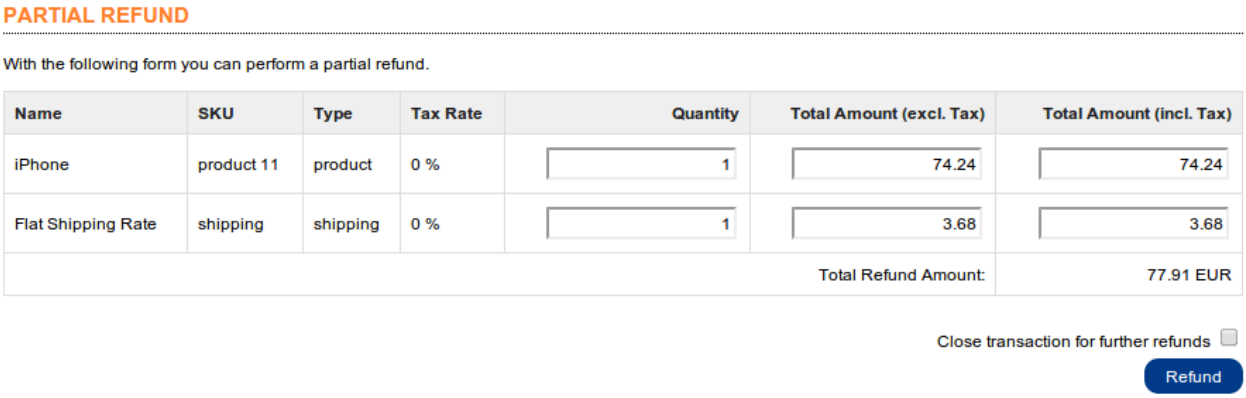


Figure 7.1: Refunds in OpenCart for Unzer.

> ### Maximal Refund
>
> With our module it is not possible to refund more than 100% of the originally authorised amount.

7.6 Set-up a cron job to activate the timed operations

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

In order to use the timed operations, please schedule a cron job in your server to the following controller:

https://www.your-shop.com/index.php?route=unzercw/cron/cron

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

7.7 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.

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:

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

9 Errors and their Solutions

9.1 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.

10 Compatibility with Third-Party Plugins

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

10.1 Birthday and gender in OpenCart

For certain payment service providers it is necessary to check the birthday an the gender of a customer. OpenCart does not check this by default.

How to enable gender and birthday checks in your shops checkout

1. Add two new custom fields to your checkout via your shops backend under "Customers > Custom Fields"
2. Modify the order context getters to return the value of your custom checkout field from the order / session (or wherever the previous step saves the data).

   Order Context Getters

   - AbstractOrderContext
   - getBillingDateOfBirth()
   - getBillingGender()

   These functions can be found in "system/library/cw/Unzer/AbstractOrderContext.php".

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 Unzer settings.

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

11.2 Log Location

The log file is stored in the default log folder of OpenCart. The path is configured in the config.php of your shop system. (Default Path: {shopRootDirectory}/system/logs or {shopRootDirectory} /system/storage/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.

In your script initialize the base of OpenCart.

Opencart 1.x

```
require_once('config.php');

require_once(DIR_SYSTEM . 'startup.php');

require_once(DIR_SYSTEM . 'library/customer.php');
require_once(DIR_SYSTEM . 'library/affiliate.php');
require_once(DIR_SYSTEM . 'library/currency.php');
require_once(DIR_SYSTEM . 'library/tax.php');
require_once(DIR_SYSTEM . 'library/weight.php');
require_once(DIR_SYSTEM . 'library/length.php');
require_once(DIR_SYSTEM . 'library/cart.php');

$registry = new Registry();
$loader = new Loader($registry);
$registry->set('load', $loader);
$config = new Config();
$registry->set('config', $config);
$db = new DB(DB_DRIVER, DB_HOSTNAME, DB_USERNAME, DB_PASSWORD,
DB_DATABASE);
$registry->set('db', $db);
```

Opencart 2.x

```
require_once('config.php');
require_once(DIR_SYSTEM . 'startup.php');
// Registry
$registry = new Registry();

// Config
```

```
$config = new Config();
$config->load('default');
$config->load('catalog');
$registry->set('config', $config);
$loader = new Loader($registry);
$registry->set('load', $loader);
$registry->set('db', new DB($config->get('db_engine'), $config->get
('db_hostname'), $config->get('db_username'), $config->get
('db_password'), $config->get('db_database'), $config->get('db_port')));
```

Opencart 3.x

```
require_once('config.php');
require_once(DIR_SYSTEM . 'startup.php');
// Registry
$registry = new Registry();

// Config
$config = new Config();
$config->load('default');
$config->load('catalog');
$registry->set('config', $config);
$loader = new Loader($registry);
$registry->set('load', $loader);
$registry->set('db', new DB($config->get('db_type'), $config->get
('db_hostname'), $config->get('db_username'), $config->get
('db_password'), $config->get('db_database'), $config->get('db_port')));
```

Include the module specific files and set registry.

```
require_once DIR_SYSTEM.'library/cw/init.php';
require_once DIR_SYSTEM.'library/cw/UnzerCw/Util.php';
require_once DIR_SYSTEM.'library/cw/UnzerCw/Entity/Transaction.php';
UnzerCw_Util::setRegistry($registry);
```

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
}
```