

# Installation Manual for Unser Gambio

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

**Release Date:** Tue, 07 Dec 2021 09:12:59 +0100  
**Version:** 1.0.16

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 Gambio Shop .....</b>	<b>8</b>
3.1	Installation .....	8
3.2	Language .....	8
3.3	Updates and Upgrades .....	8
3.3.1	Update Checklist .....	8
3.3.2	Update Instructions .....	9
<b>4</b>	<b>Module Configuration in the Gambio Shop .....</b>	<b>10</b>
4.1	Configuration of the Main Module .....	10
4.1.1	Create Pending Orders .....	10
4.1.2	Database Encoding .....	10
4.2	Direct Capturing of Transactions .....	11
4.3	Uncertain Status .....	11
4.3.1	Setting the order state .....	11
4.4	Optional: Validation .....	11
<b>5</b>	<b>Settings / Configuration of Payment Methods .....</b>	<b>12</b>
5.1	General Information About the Payment Methods .....	12
5.2	Information on Payment Status .....	12
5.2.1	Order status "pending" / imminent payment (or similar) .....	12
5.2.2	Order status "cancelled" .....	12
5.3	Invoice .....	13
5.3.1	Code Adjustments for the MasterPass Integration (Gambio GX 1.x) .....	13
5.3.2	Further information .....	13
5.4	Prepayment .....	13
5.5	Instalment .....	14
5.6	Secured Invoice .....	14
<b>6</b>	<b>The Module in Action .....</b>	<b>15</b>

6.1	Recommended Practice .....	15
6.2	Useful Transaction Information on the Order .....	15
6.3	Using Invoice Details of a Processor .....	16
6.3.1	Gambio Order Confirmation (E-Mail) .....	16
6.3.2	Gambio Invoice (PDF) .....	16
6.3.3	Gambio-Backend (Transaction details) .....	17
6.3.4	Gambio Success-Page .....	17
6.3.5	Refunding Orders .....	18
6.4	Capturing / Canceling of Orders .....	18
6.4.1	Capturing / Canceling of Orders .....	18
6.5	Setup a Cron Job to Activate the Timed Operations .....	19
6.6	Partial Cancel .....	19
<b>7</b>	<b>Testing .....</b>	<b>20</b>
7.1	Test Data .....	20
<b>8</b>	<b>Errors and their Solutions .....</b>	<b>22</b>
8.1	The Referrer URL appears in my Analytics Tool .....	22
<b>9</b>	<b>Error Logging .....</b>	<b>23</b>
9.1.1	Log Levels .....	23
9.1.2	Log Location .....	23
<b>10</b>	<b>Advanced Information .....</b>	<b>24</b>
10.1	Transaction Object .....	24

# 1 Introduction

This manual explains the installation, configuration and usage of the payment module for Gambio 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
- Gambio 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 Gambio. 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.

### Gambio v3.2.x

Gambio in version 3.2.x uses a different checkout process than other versions, and is not supported. If you use this version, please upgrade to a newer version of Gambio.

## 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.

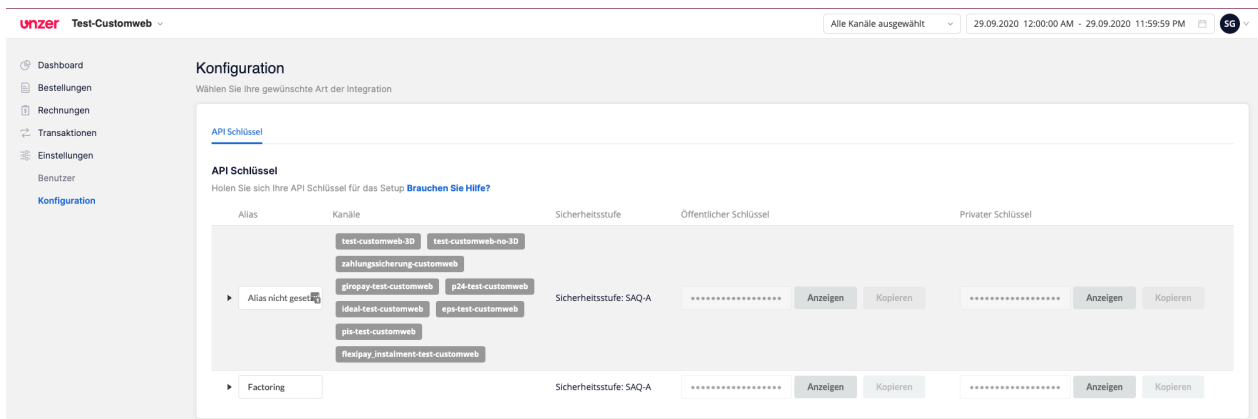


Figure 2.1: API Keys

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 Gambio 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 Language

1. In the unzipped folder navigate to the folder "lang"
2. Using your preferred FTP client upload **entire content** of this folder into the lang directory of your shop.  
Make sure that the contents aren't replaced but merely merged.

### 3.3 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.

#### 3.3.1 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.

### 3.3.2 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 Gambio 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 > Module-Center(alt) > Unser Base Module"**. Install the module and click on "Open Method Configuration". A new window should open up. Enter the individual options such as described above in the configuration of the administration interface of Unser. Should one of the options not be clear to you, you can find further information by moving the cursor to the question mark. The most central settings are explained in more detail in this manual.

#### 4.1.1 Create Pending Orders

You have the possibility of creating pending orders. In this case a temporary order will be created with the status "pending payment". This allows you to reserve the item in your shop and to transmit the identical order number as in the shop to Unser. In the case when a customer cancels his or her purchase, the order remains in your shop - except if you activate the option "Remove cancelled orders". This makes sure that cancelled orders are deleted regularly. If you do not wish to create pending orders, simply keep the default setting of xt:Commerce. However, this also means that the reference number (payment ID) with Unser doesn't correspond to the order number in the shop. You are, however, free to use both options.

#### 4.1.2 Database Encoding

If you haven't selected UTF-8 as a default for your database, this might lead to encoding problems with parameters with special characters. In this case you must select the option that data is UTF-8 encoded before being sent to Unser.

## 4.2 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.

## 4.3 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.

### 4.3.1 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.

## 4.4 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

## 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.

### 5.3.1 Code Adjustments for the MasterPass Integration (Gambio GX 1.x)

If you use GX 2.x, the following code adjustments are not relevant for you.

In order for the button "Pay with MasterPass" to appear in the shopping cart when using external checkout with address transmission, the following adjustments in the code have to be made. If you prefer to simply use MasterPass as a regular payment method without address transmission, no changes are necessary.

The files containing the necessary code adjustments can be found in the zip files in /changes/\*.txt. Please paste the codes contained in these files into the location described below.

#### 5.3.1.1 Adjusting shopping\_cart.php

Please copy the code snippet from the file /changes/shopping\_cart.php.txt in front of the following code:

```
$smarty->assign('language', $_SESSION['language']); (ca. line 140)
```

#### 5.3.1.2 Adjusting templates/[current-active-template]/module/shopping\_cart.html

Please copy the code found in /changes/shopping\_cart.html.txt in front of the **tag-{/if}**:

### 5.3.2 Further information

MasterPass with External Checkout cannot be used in combination with vouchers. If the customer wants to redeem a voucher, she has to go through standard checkout.

## 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 Unser module.

### 6.1 Recommended Practice

Generally the most of the actions can be carried out from within the shop backend or from within the Unser 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.

CREDIT CARD INFO	
<b>Credit Card Number:</b>	
<b>Security Code (CVV):</b>	
<b>Credit Card Expires:</b>	
<b>Payment Information</b>	Visa (400222_16)

Figure 6.1: Transaction Information

### 6.2 Useful Transaction Information on the Order

You can find an overview of the transaction information in within the order detail view. Among others, this information allows for the definite attribution of the orders to their corresponding transaction, seen in the backend of Unser.

# Transaktionsnummer

Transaktions ID	12
Transaktionsnummer	9_12
Bestell ID	9
Erstellt am	2013-11-26 18:03:28
Erneuert am	2013-11-26 18:03:59
Kunden ID	1
Zahlungsid	25691316
Zahlungs-Klasse	goldmaster_visa
Autorisierungsbetrag	125
Währung	EUR
Zahlungsmethode	Visa
Zahlungsid	25691316
Verbuchter Betrag	125
Transaktion autorisiert	Ja
Transaktion ist unsicher ?	Nein
Transaktion verbucht	Ja
Transaktion bezahlt	Ja
Akzeptanz	test123
Kartennummer	XXXXXXXXXXXX0003
Kartenablaufdatum	06/16
Händler Referenz	9_12_2013_11_26_18_03_59

Figure 6.1: Transaction Information

## 6.3 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:

### 6.3.1 Gambio Order Confirmation (E-Mail)

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



### 6.3.2 Gambio Invoice (PDF)

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

### 6.3.3 Gambio-Backend (Transaction details)

You can view the transaction details in Gambio under **Modul-Center (old) > Unser Base Module > open main module configuration > (above) Transaction Management**.

Home
Base Configuration
Transaction Management
Über
Setup
Payment Page Design

Verbuchung
Stornieren

#### Transaktionsinformation der Transaktion 400210\_1

Transaktions-ID	1
Transaktionsnummer	400210_1
Autorisierungsstatus	successful
Bestell ID	400210
Erstellt am	2017/03/07 18:33:30
Aktualisiert am	2017/03/07 18:33:47
Kunden ID	1
Zahlungs-ID	3575612
Zahlungs-Klasse	customweb_openinvoice
Autorisierungsbetrag	30
Währung	EUR
Zahlungsmethode	Open Invoice
Zahlungs-ID	3575612
Testtransaktion	Ja
Transaktion autorisiert	Ja
Transaktion ist unsicher	Nein
Transaktion bezahlt	Ja
Zahlungsinformationen	Bank: Customweb Test Bank Kontoinhaber: Customweb GmbH IBAN: DE1200TEST333222110000546468 BIC: TESTBIC0001 Referenznummer: BP3575612/2749

#### Transaktionsübersicht

Datum	Aktion	Nachricht
2017/03/07 18:33:47	authorization	Der Betrag von '30.00' wurde autorisiert.

Figure 6.1: Transaction details within Gambio.

### 6.3.4 Gambio Success-Page

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

### 6.3.5 Refunding Orders

In order to refund orders, open the transaction information (cf. above). There you will find information on the transaction. Click on 'Refunding' and select the order which you would like to create a refund for. By clicking on 'Refunding', the refund is transmitted to Unzer.

Zurück

### Teilgutschrift

Name	SKU	Typ	MwSt.	Menge	Total (exkl. MwSt.)	
Testartikel	ABC123	product	8 %	1	20.54	2
Selbstabholung (Selbstabholung der Ware in unserer Geschäftsstelle.)	shipping	shipping	0 %	1	0.00	0
Total gutgeschriebener Betrag:						

Transaktion für weitere Rückvergütungen schliessen ☐

**Figure 6.1:** Refund button

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

## 6.4 Capturing / Canceling 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.

### 6.4.1 Capturing / Canceling of Orders

In order to capture orders, open the transaction information (cf. above). There you will find information on the transaction. Further below you will find an input field in which you can enter the amount you wish to capture. By clicking on "capture", the capture will be transmitted to Unzer. You no longer need to log into the backend of Unzer.

By clicking on "Cancel Transaction" you cancel the transaction and the reserved amount is immediately released on the card of your customer.

## Transaktion verbuchen

Betrag vergüten ?

Transaktion schliessen ?

☐ Ja, Transaktion schliessen

Verbuchen

## Transaktion abbuchen

Transaktion abbuchen

Figure 6.1: capture oder cancel button.

## 6.5 Setup a Cron Job to Activate the Timed Operations

To activate the timed operations of the plugin (e.g. update service, deleting pending orders, etc.), please activate in a regular request to the file **unzercw\_cron.php**. The regular call triggers the appropriate actions. We recommend an intervall of about 15 minutes.

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.

## 6.6 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	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 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.

### 9.1.1 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.

### 9.1.2 Log Location

The log messages are visible in the Gambio backend under "**Modules > Module-Center > Unzer Base Module > Log Messages**".

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

```
include ('includes/application_top.php');
```

Include the module main file and module classes.

```
require_once DIR_FS_DOCUMENT_ROOT.'admin/includes/UnzerCw/init.php';
require_once DIR_FS_DOCUMENT_ROOT.'admin/includes/UnzerCw/classes/UnzerCw/Entity/Util.php';
```

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

Load the transaction by Id:

```
$transactionById = UnzerCw_Entity_Util::findTransactionByTransactionId
($transactionId);
$transactionObject = $transactionById->getTransactionObject();
```

Load the transaction by external Id:

```
$transactionById = UnzerCw_Entity_Util::
findTransactionEntityByTransactionExternalId($externalId);
$transactionObject = $transactionById->getTransactionObject();
```

Load transactions by Order ID:

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