Installation Manual for TWINT WooCommerce

This manual describes the installation and usage of the TWINT extension for WooCommerce.

Release Date:	Mon, 07 Aug 2023 14:01:06 +0200
Version:	4.0.85

wallee AG

General-Guisan-Strasse 47

CH-8400 Winterthur

E-Mail: info@sellxed.com

Telefon:

CH: +41 (0)44 505 13 60

DE: +49 (0)40 2999 60117

UK: +44 (0)161 850 6890

US: +1 (0)205 557 5869

####conditional####

Table of Contents

1 Introduction

This manual describes how to install, configure and use the payment module for WooCommerce and TWINT.

Before you begin with the installation make sure you have all of the necessary data. You will need:

- TWINT UUID of your shop
- TWINT Certificate Password (created by you)
- TWINT Certificate
- WooCommerce Payment module from sellxed.com/shop
- Login data to your Server and Shop

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

1.1 Installation Process

This document contains all information necessary to install the module. Strictly follow the steps on the check list when setting up the module. This is the only way to ensure a safe usage in compliance with all security regulations.

1. Before beginning with the installation you will need to register in the TWINT dealer portal
2. Configuration of the Main Module
3. Configuration of the accepted payment methods and the respective authorization methods
4. Conducting a test order

Installation Service

Our payment plugins should have per default the correct settings for most of our customers' preferences. That means once you have entered the required credentials in the plugin configuration to connect your account to your website, the plugin should be fully operational. Should you be willing to receive detailed information on a setting you do not know, you may contact our support team who will be able to assist you further.

Our support team is at your disposal during regular business hours at: http://www.sellxed.com/support. Furthermore, you have the option of ordering our installation service. We will make sure the plugin is installed correctly in your shop: http://www.sellxed.com/shop/de/integration-und-installation.html

1.2 System Requirements

In general, the plugin has the same system requirements as WooCommerce. 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.

2 Configuration of the TWINT - Dealer Portal

2.1 Certificate and UUID of your Store Download

You will receive the necessary access data directly in the dealer portal of TWINT. Log in to the corresponding location in the TWINT Dealer Portal.

> ### What steps do you need to take for TWINT?
>
> 1. First you need to register in the dealer portal
> 2. Set up an online shop configuration. In case you would like to order an installation service, transmit the certificate as well as the certificate password which you will have created in the process of setting up your store, to customweb.
>
> For further information and frequently asked questions please visitwww.twint.ch

2.1.1 Entering the SystemID

You will receive the **TWINT UUID** in the initial email from TWINT. It is important that you save the UUID as we need it to set up your account. Copy the TWINT UUID and paste it into the corresponding location in the Main Module.

2.1.2 Converting the Certificates

The **certificate** will be retrieved and ordered automatically during the process of creating your store in the dealer portal of TWINT. If you have forgotten your certificate password or are having problems with generating the certificate please contact TWINT directly.

You will need to convert the certificate into a .pem file first. To do so you can use a service (but we strongly advise against it as you should not pass along your certification password to anyone except customweb).

You may want to use **openssl** to convert the certificates. Openssl is preinstalled on most unix based systems. On Windows it can be installed. The conversion can be done as following *openssl pkcs12 -in certificate.pfx -out certificate.pem*.

As soon as you are in possession of the .pem certificate, copy the complete content of the .pem file (certificate incl. -----BEGIN CERTIFICATE----- and -----END RSA PRIVATE KEY-----) into the text field.

Make sure you don't forget to also fill in the password into the main module.

> ### Which steps do you need to take for TWINT?
>
> If you have ordered the integration service you will need to transmit the certificate which you created above to customweb.

2.1.3 Time Out Settings

Technically, TWINT does not send payment notifications to your shop. The server pulls the result of the transaction periodically. In order for this to work you need to have set the Server Polling Time Out. However the standard values shouldn't be altered. Make sure that you have set up the Cron Job in your module. You will find more information in the manual further down.

2.2 Configuration of the Payment Methods

The payment method TWINT will also be delivered in the module. You will find it in the configuration of the payment methods in your shop. Activate this payment method in order for it to be available for your customers in your shop.

Further information regarding the configuration of the payment methods and the implementation of the different settings can be found here.

3 Module Installation and Update in the WooCommerce Shop

3.1 Installation

At this time you should already be in possession of the module. Should this not be the case, you can download the necessary files in your customer account in the sellxed shop (Menu "My Downloads Downloads"). In order to install the module in your shop, please carry out the following steps:

1. Download the plugin. The download can be found in your sellxed.com account under "My Downloads".
2. Unzip the archive you have just downloaded.
3. In the unzipped folder navigate to the folder "files"
4. For some shops there are different versions of the plugin provided. If this is the case open the folder which corresponds to your shop version.
5. Using your preferred FTP client upload **entire content** of this folder into the root directory of your shop. For some shops there is a specific folder containing the plugins. If that is the case upload the plugin into this folder. Make sure that the folders aren't replaced but merely merged.
6. If you haven't yet done so, log back into your shop.

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.

More information regarding the subscription of this RSS feed can be found under: http://www.sellxed.com/en/updates_upgrades.

We only recommend an update if something doesn't work in your shop, if you want to use new feature or if there is a necessary security update.

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. However, if you decide to perform the update directly in your live shop there is the possibility of a downtime of the shop of more than two days depending on the availability of our support if you do not want to book our complementary support.

Depending on the version it could be that the database has to be migrated. We recommend you therefore, to perform the updates in times when the shop is not visited too frequently by your customers.

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 WooCommerce 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 can activate the plugin by clicking "activate" under the module WooCommerce TWINT in the menu **Plugins**.

After the Installation of the plugin a new menu on the dashboard will appear called **TWINT** . Enter all data in the corresponding fields. Each option is, furthermore, explained in short info texts in the shop.

4.2 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 **WooCommerce > Settings > Checkout** (In earlier version it was **WooCommerce > Settings > Payment Gateways**). 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 **Enable** the payment method is activated in your shop. For more detailed information on the settings, please refer to information provided directly within the module configuration.

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 TWINT and the module**
>
> It may be that settings saved in the payment modules overwrite settings saved in TWINT.

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

4.6 Redirection / Payment Form Error

If the successful redirection to the payment page suddenly stops working or you are only able to see the header and footer of your template instead of the payment form, you might be affected of the following issue. As a first step, you should update your payment plugin to the latest version, deactivate it in the backend of WooCommerce and activate it again.

If you are using **WPML** it can happen that the page for the redirection of the customer to TWINT payment page results in a '404 Page Not Found' error. In this case, you will have to duplicate our checkout page under 'Pages' in the corresponding languages. In order to do so please follow these steps:

1. Go to the WooCommerce backend and select: Pages > All Pages. There you should find a Page called "TWINT Checkout".
2. Delete all existing pages called "TWINT Checkout".
3. In the plugin section of your Wordpress dashboard, deactivate the WooCommerce TWINT plugin and activate it again. This will help you to generate a new redirection page.
4. Navigate again to Pages > All Pages. Open the configuration of the new redirection page.
5. On the right hand side under 'languages' you can now duplicate the content of this page for all of your installed languages. When you have successfully saved the changes, this page will be active for all of your languages and the error should have disappeared.

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

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.

6 The Module in Action

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

6.1 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 TWINT.
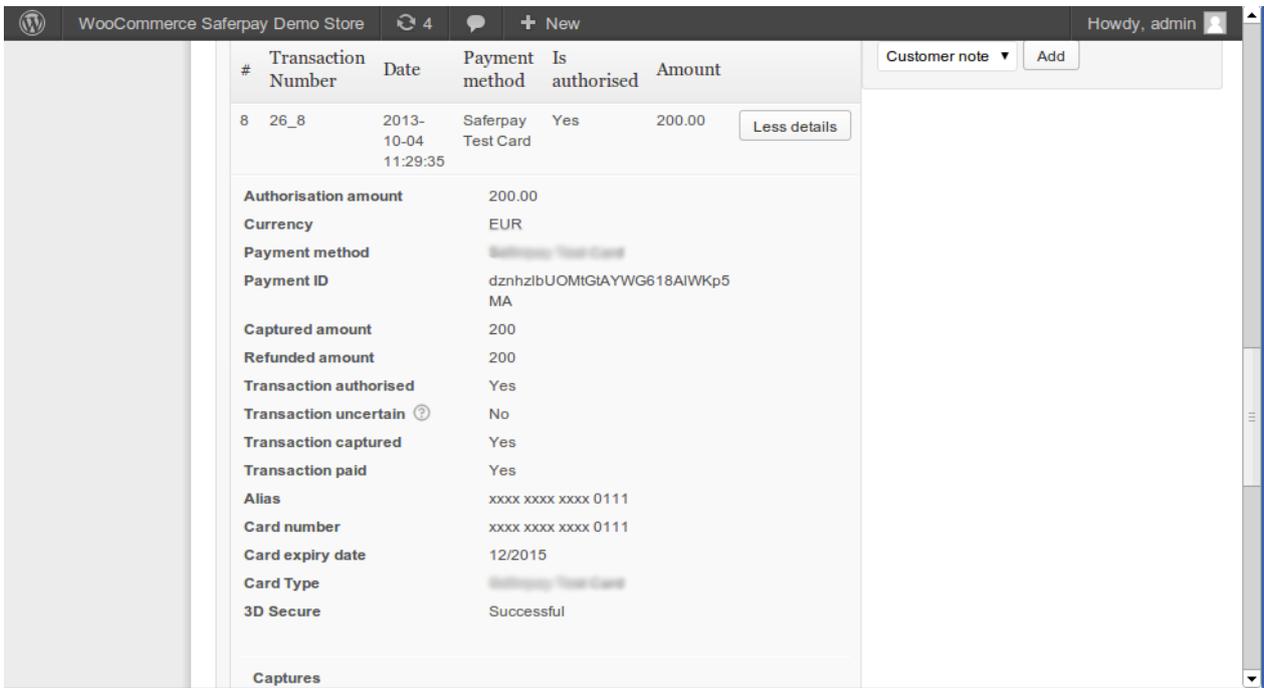
Figure 6.1: Transaction Information in WooCommerce.

6.2 Refunds

You can create refunds for charged transactions and transmit them to TWINT automatically. In order to do this, open the invoice for a captured order. By clicking on "Credit Memo", a new dialogue for refunds opens. Adjust the amount you would like to refund and click "Refund" to submit it to TWINT.

The process demonstrated by screenshots:

Figure 6.1: 1. Open the invoice

Figure 6.1: 2. Click on Credit Memo - Please note that the order has to be opened first.

Figure 6.1: 3. Adjust the amount (cf. below Refund to based on quantity or any amount)

Figure 6.1: 4. Click on **Refund** to transmit.

Figure 6.1: Wait until the refund is confirmed.

6.2.1 Refund Based on Quantity

You can create refunds based on the product quantity by adjusting the number of products you wish to refund in the credit memo dialogue.

You can create refunds until the complete original amount has been credited. The order status switches to **complete** automatically.

Figure 6.1: Refund overview

6.2.2 Refunding Customized Amounts

Magento also allows you to define any amount to refund. In order to do this, follow these steps:

1. Open the Credit Memo Dialogue as described above
2. Set the number of products you wish to refund to 0
3. Enter any amount in the field **Adjustment Refund**

6.2.3 Deducting a Processing Fee for Refunds

You can charge a processing fee for the refunds by following these steps:

1. Open the Credit Memo Dialogue as described above
2. In the field **Adjustment Fee**, define any amount as a processing fee.

Adjusting the amounts for several refunds

It is possible to conduct several refunds for one order. Please note, that the maximum amount for the total of refunds is 100% of the original amount. If you already conducted a refund you need to make sure not to refund more than the original amount by adjusting the quantity and the adjustment fee.

6.3 Capturing of Orders

Please note

The transaction management between your shop and TWINT is not synchronized. If you capture payments with TWINT, 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.

Open the order and look for the TWINT transaction tab at the bottom. Capturing via the internal operation of WooCommerce is not supported.

Customweb Transactions

#	Transaction Number	Date	Payment Method	Authorised	Amount	
6	16	2017-01-06 10:23:49	MasterCard	Yes	10.00	Less Details

Authorisation Amount	10	**Currency**	CHF	
Payment Method	MasterCard	**Payment ID**	4S2Wtjb3E7d7tASKGxK1bCEWx hdA	
Test Transaction ❓	Yes	**Transaction authorised**	Yes	
Transaction uncertain ❓	No	**Transaction paid**	Yes	
Payment Method Name	MasterCard	**Payment Details**	xxxx xxxx xxxx 0000	
Card holder name	Customweb	**Card Expiry Date**	12 / 20	
Card Country Origin	DE	**Acquirer Reference**	000000	
3ds Authentication	Yes	**3ds Liability Shift**	Yes	
Authorisation Method	PaymentPage			

Capture

Cancel

Previous actions

Date	Action	Message
2017-01-06 10:24:21	authorization	The amount of 10.00 is authorized.

Figure 6.1: Transactiondetails in the transaction tab of a specific order.

Click on "More Details" to view the transaction details. By clicking **Capture** button, you get into the following context.

Name	SKU	Type	Tax	Quantity	Total Amount (excl. Tax)	Total Amount (incl. Tax)
Testproduct	Testproduct	product	0 %	1	10.00	10.00
				Total captured amount:		10.00CHF

Back

Capturing

Figure 6.1: Deferred/Manual capture of transactions within WooCommerce

Enter the amount of your choice that you wish to capture from the customer's card. By clicking **Capture** a direct capturing of the order occurs with TWINT.

Please be aware that cancelling the transaction does not cause a status change of the order.

6.4 Refunds

You can refund already captured transactions and automatically transmit them to TWINT.

> **Please note**
>
> That the default WooCommerce functionality is not supported by our plugin.

Customweb Transactions

#	Transaction Number	Date	Payment Method	Authorised	Amount	
4	13	2016-11-04 14:50:07	MasterCard	Yes	20.00	Less Details

Authorisation Amount	20	**Currency**	EUR	
Payment Method	MasterCard	**Payment ID**	b3YMh8bj8d4pvAnj0ddGAGvj3 t9A	
Captured amount	20	**Test Transaction** ❓	Yes	
Transaction authorised	Yes	**Transaction uncertain** ❓	No	
Transaction captured	Yes	**Transaction paid**	Yes	
Payment Method Name	MasterCard	**Payment Details**	xxxx xxxx xxxx 0000	
Card holder name	Customweb	**Card Expiry Date**	12 / 20	
Card Country Origin	DE	**Acquirer Reference**	000000	
3ds Authentication	Yes	**3ds Liability Shift**	Yes	
Authorisation Method	PaymentPage			

Refund

Captures

Date	Amount	State
2016-11-04 14:51:34	20	succeed

Previous actions

Date	Action	Message
2016-11-04 14:51:33	authorization	The amount of 20.00 is authorized.
2016-11-04 14:51:34	capturing	The amount of 20.00 is captured.

Figure 6.1: Transactionsdetails via Orders > TWINT Transactions.

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

Figure 6.1: Refund of transactions within WooCommerce

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 TWINT 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.

Furthermore, please be aware that processing a refund will not cause a status change of the order.

6.5 Transaction Process

Each action affiliated with transaction management is logged by the module.
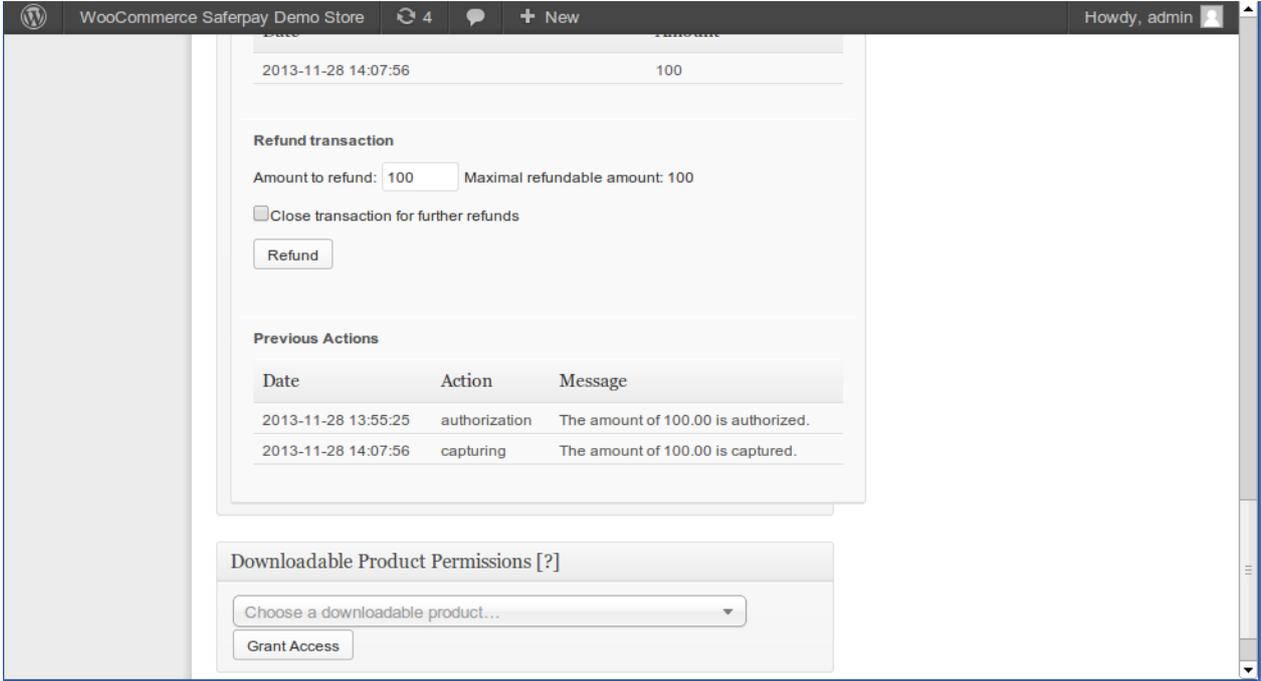
Figure 6.1: Transactions in the Overview of the Transaction Process

6.6 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.) make sure that you set up the WooCommerce Cron engine. Especially the update function allows you to automatically retrive additional information or changes of your order directly via the API of TWINT. Please note it could be that in order to use the update feature it may be necessary that TWINT activates additional options in your account.

The module uses the standard cron engine of WooCommerce. More infromation regarding the set up can be found here.

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:

7.2 Testing with TWINT

Make sure you are in the Live Mode and carry out a test transaction with your smart phone. For the tests we recommend you set the booking in the configuration of the payment method to 'delayed' so that the money won't be debited directly. Testing in test mode is only possible if you have the special Twint developer app. This needs to be setup for you by Twint directly. If you are interested in this please contact Twint.

8 Errors and their Solutions

You can find detailed information under http://www.sellxed.com/en/faq. Should you not be able to solve your problem with the provided information, please contact us directly under: http://www.sellxed.com/en/support

8.1 Page Not Found Error prior to the redirection to Payment Page

If you are using WPML it can occur that the Pages for the display of the payment methods or for the redirection to TWINT end in a Page Not Found Error.

In this case you have to duplicate our Pages in the corresponding languages. In order to do so please follow these steps:

1. Go to the main menu and select **Pages > All Pages**. There you should find a Page called TWINT Checkout.
2. Open the configurations.
3. On the right hand side under languages you can now duplicate the content of this page for all of your installed languages. When you have successfully saved the changes this page will be active for all of your languages and the Error should have disappeared.

8.2 The Referrer URL appears in my Analytics Tool

When a customer and the notification are redirected via Header Redirection, the TWINT 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 WooCommerce

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

- Checkout Fees for WooCommerce

9.2 Multilingual WooCommerce sites

To run multilingual WooCommerce sites you will need the following 3rd-Party plugin.

- WPML for WooCommerce

9.3 Birthday and gender in WooCommerce

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

> ### How to enable gender and birthday checks in your shops checkout
>
> 1. Either add custom checkout field yourself or download Flexible Checkout Fields to do so.
> 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 "wp-plugins/woocommerce_twintcw/classes /TwintCw/AbstractOrderContext.php".

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 TWINT settings.

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

10.2 Log Location

The log file is stored in **wp-content/uploads/wc-logs**. Please make sure the path exists and it is writable by the webserver.

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 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/woocommerce_twintcw/classes
/TwintCw/Util.php';
```

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

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

Load the transaction by the external Id:

```
$transactionByExternal = TwintCw_Util::getTransactionByTransactionNumber
($externalId);
$transactionObject = $transactionByExternal->getTransactionObject();
```

Load transactions by Post ID, this is the internal WooCommerce order id:

```
$transactionsByPostId = TwintCw_Util::getTransactionsByPostId($orderId);
foreach($transactionsByPostId as $transaction){
    $transactionObject = $transaction->getTransactionObject();
    //Do something with each object
}
```