Installation Manual for Dimoco e-Commerce

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

Release Date:	Mon, 07 Aug 2023 14:01:06 +0200
Version:	1.0.161

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

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

####conditional####

Table of Contents

1 Introduction

This manual leads you through the installation, configuration and usage of the payment module for e-Commerce and DIMOCO Payment Services (Dimoco).

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

- Your Dimoco API Username and Password
- Access to your merchant account of Dimoco - On this plattform you can download the access information.
- e-Commerce payment module from sellxed.com/shop.
- Access to your server.

In case you don't yet have a contract with Dimoco, we please ask you to contact their support. In case you do not have a contract with Dimoco yet, we kindly ask you to contact Dimoco Sales (payment-merchants@dimoco.eu).

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

1.1 Process 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 by doing so, can a secure usage in correspondence with all security regulations be guaranteed.

1. Configuration of the basic settings of the payment module
2. Configuration of the payment methods
3. Carrying out of a test purchase by means of the attached test data at the end of this document
4. Get in touch with your Technical Account Manager at Dimoco. This person will help you set up your account correctly for the successful handling of transactions.
5. If your test was successful, do not forget to switch the operating mode to live mode. Once you have switched to live, contact your Technical Account Manager to ensure that the live transactions can now be processed.

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

.htaccess Directory Protection

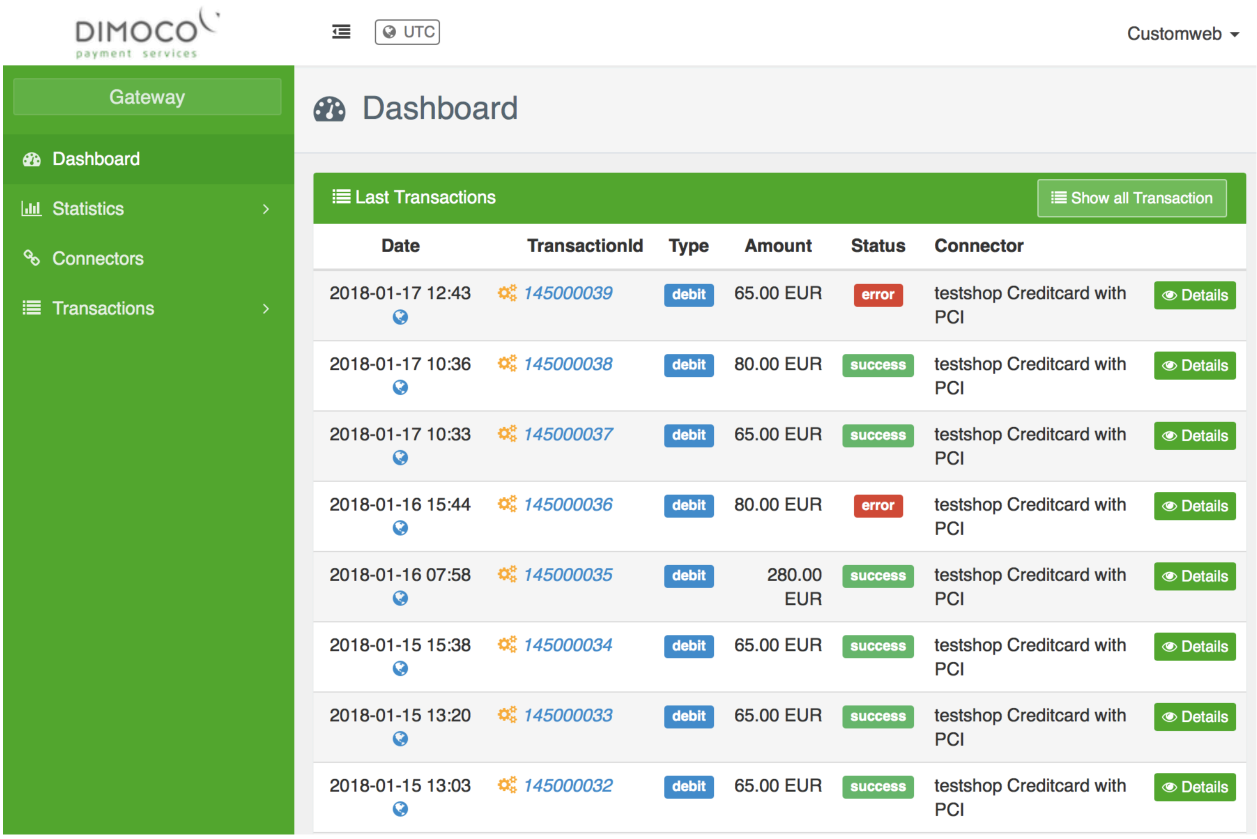
In order to test the module, any kind of directory protection or IP blocking on your server must be deactivated. This is crucial as otherwise the payment feedback of Dimoco might not get through to the shop.

2 Configuration of the Payment Module and the Payment Methods

In order to configure and set up the main module in e-Commerce please double check that you received the following information:

- API Username (provided by Dimoco on setup email)
- API Password (provided by Dimoco on setup email)
- API Key (From your Connector's Settings)
- Shared Secret (From your Connector's Settings)

All the required information is provided directly from Dimoco or can be seen within the tab 'Connectors' at your Dimoco merchant account.



Date	TransactionId	Type	Amount	Status	Connector	
2018-01-17 12:43	145000039	debit	65.00 EUR	error	testshop Creditcard with PCI	Details
2018-01-17 10:36	145000038	debit	80.00 EUR	success	testshop Creditcard with PCI	Details
2018-01-17 10:33	145000037	debit	65.00 EUR	success	testshop Creditcard with PCI	Details
2018-01-16 15:44	145000036	debit	80.00 EUR	error	testshop Creditcard with PCI	Details
2018-01-16 07:58	145000035	debit	280.00 EUR	success	testshop Creditcard with PCI	Details
2018-01-15 15:38	145000034	debit	65.00 EUR	success	testshop Creditcard with PCI	Details
2018-01-15 13:20	145000033	debit	65.00 EUR	success	testshop Creditcard with PCI	Details
2018-01-15 13:03	145000032	debit	65.00 EUR	success	testshop Creditcard with PCI	Details

Figure 2.1: Overview of the backend (My Payments) of Dimoco.

Enter the required credentials directly into the main module to allow payments over Dimoco. If you want to have additional information regarding how to use the backend of Dimoco we ask you kindly to directly call the customer support of Dimoco.

2.1 Live and Test Credentials

> Please note that the configuration is different for live mode than for test mode as described in the following sections.

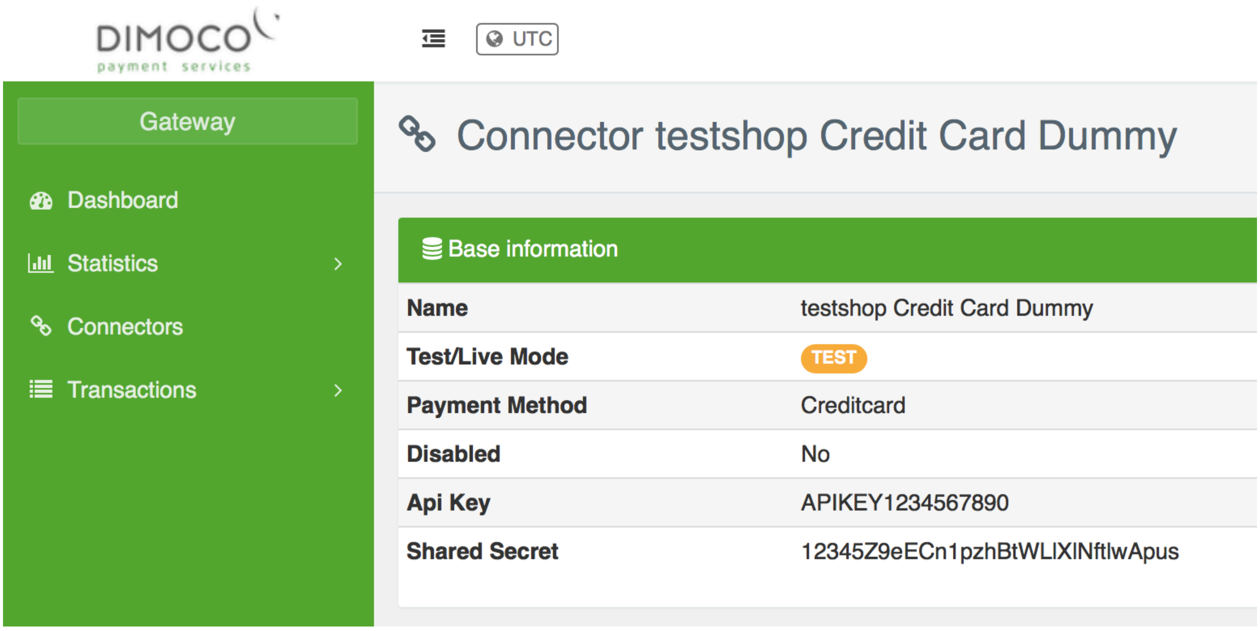
2.1.1 Live Credentials

Enter the API Username and the API Password into the main configuration of the plugin. In your Dimoco account under "Connectors" you will find a connector for every payment method which has been set up by Dimoco.

Inside these connectors you will find the parameters API Key and Shared Secret.For each connector enter the API Key and Shared Secret into the configuration of the corresponding payment method configuration in your shops backend.

2.1.2 Test Credentials

A general test connector can be set for all payment methods. Specifically for credit cards and SEPA, it is possible to set your own test connector. If this is not set, the general test connector will be used. These connectors are used to test different payment flows.



Connector testshop Credit Card Dummy

Base information

Name	testshop Credit Card Dummy
Test/Live Mode	TEST
Payment Method	Creditcard
Disabled	No
Api Key	APIKEY1234567890
Shared Secret	12345Z9eECn1pzhBtWLIXINftlwApus

Figure 2.1: The access credentials within the connector at Dimoco.

> The connectors for live and test mode will be set up by Dimoco. If you wish to have additional connectors setup please contact Dimoco directly.

2.2 Payment Method Specific Configurations

2.2.1 Credit Card 3D Secure

You have the option to activate or deactivate 3D Secure authorisation within the payment method Credit Card. You can change the 3D Secure behaviour by navigating to the payment method settings and changing your 3D Secure settings to Off, Optional or Mandatory.

2.2.2 SEPA Direct Debits Mandates

If you have activated the payment method SEPA Direct Debits, you will need to make sure to configure the 'Mandate Creditor ID' and the 'Mandate Merchant Name'.

2.3 Integrated Payment Form (Ajax Authorization)

If you wish to use the embedded form (Ajax) you may activate the payment method called Credit Card (Ajax) instead. You will have to enter a 'Public Integration Key' which will be provided to you by Dimoco. You can also change the layout of the CVV element using CSS.

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. 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 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 Dimoco in the **Plugin** menu.

The settings for the basic module can be found at **Dimoco > Dimoco** . Fill out the fields, you should have already entered the necessary data in the Dimoco backend or you received it as part of your correspondence with Dimoco. 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 Dimoco and the module
>
> It may be that settings saved in the payment modules overwrite settings saved in Dimoco.

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

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.

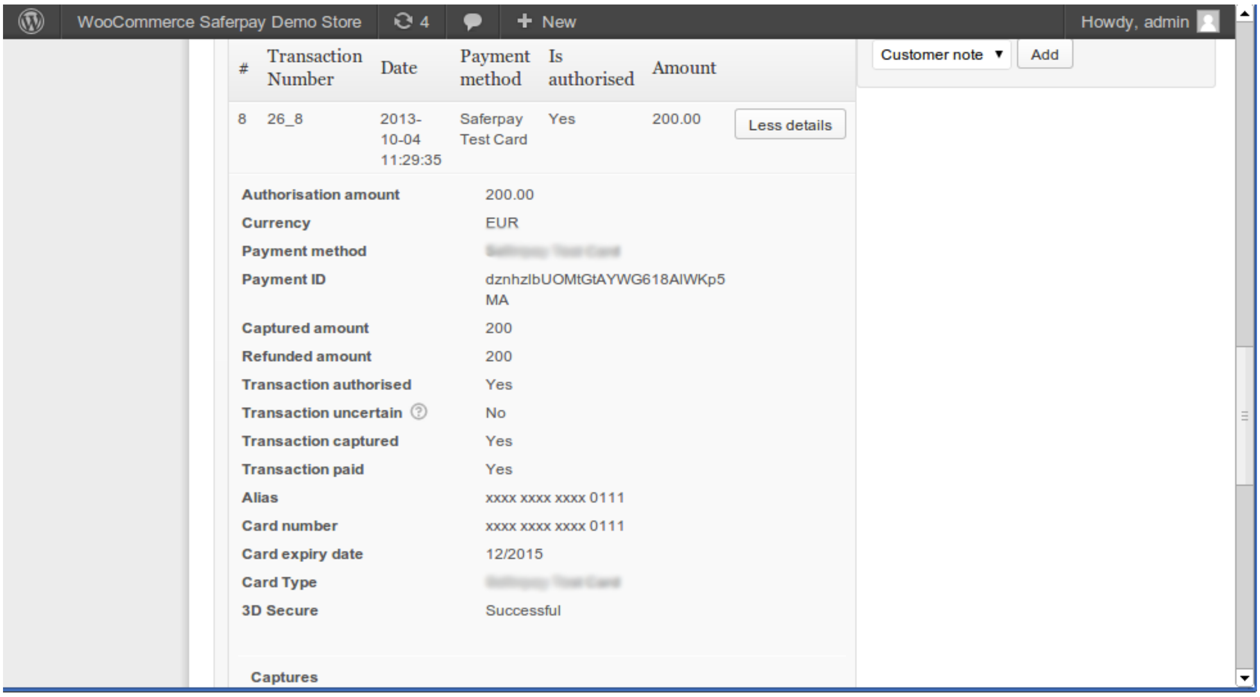
6 The Module in Action

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

6.1 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
8	26_8	2013-10-04 11:29:35	Saferpay Test Card	Yes	200.00

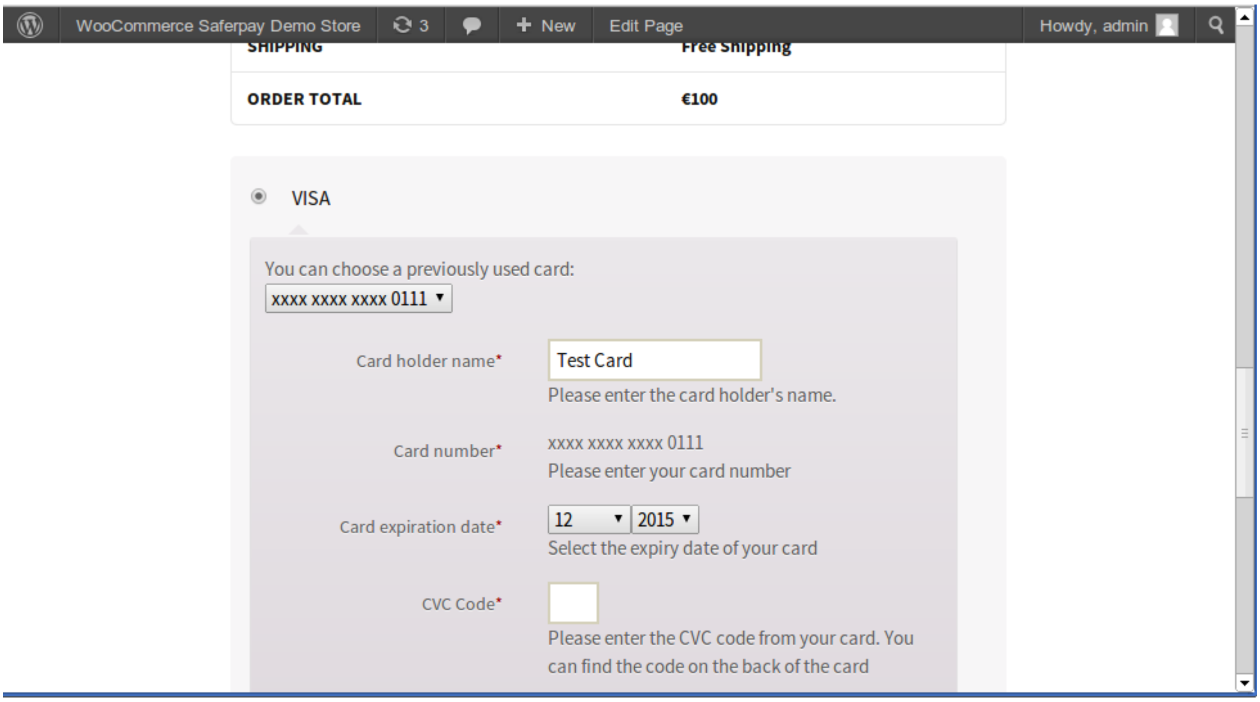
Authorisation amount	200.00
Currency	EUR
Payment method	[redacted]
Payment ID	dznhzlbUOMtGtAYWG618AIWKp5MA
Captured amount	200
Refunded amount	200
Transaction authorised	Yes
Transaction uncertain	No
Transaction captured	Yes
Transaction paid	Yes
Alias	xxxx xxxx xxxx 0111
Card number	xxxx xxxx xxxx 0111
Card expiry date	12/2015
Card Type	[redacted]
3D Secure	Successful

Captures

Figure 6.1: Transaction information in WordPress E-Commerce.

6.2 Alias Manager / Token Solution Usage

The Alias Manager allows you to store your customers' credit card data safely with Dimoco. You can enable this by activating the option "Alias Manager" in the configuration of the Payment Method. The customer can then select between the saved cards when making purchases in the future and does not have to enter all credit card data again.



SHIPPING Free Shipping

ORDER TOTAL €100

○ VISA

You can choose a previously used card: xxxx xxxx xxxx 0111

Card holder name*: Test Card
Please enter the card holder's name.

Card number*: xxxx xxxx xxxx 0111
Please enter your card number

Card expiration date*: 12 2015
Select the expiry date of your card

CVC Code*:
Please enter the CVC code from your card. You can find the code on the back of the card

Figure 6.1: Alias Manager in WooCommerce

> Alias Manager Options
>
> In order to use the Alias Manager, the option has to be unlocked by Dimoco. Please contact the support service if you wish to activate the option.

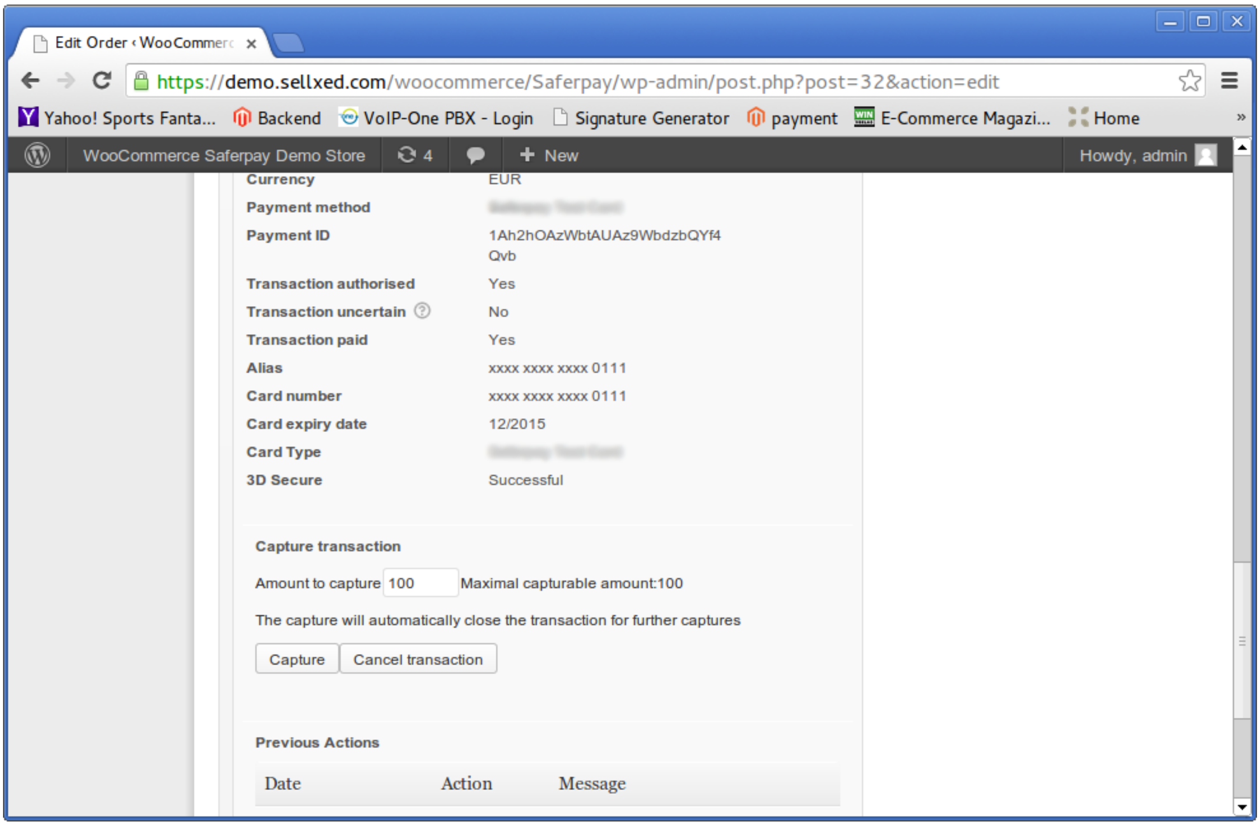
6.3 Capture / Cancel Orders

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



Currency	EUR
Payment method	[redacted]
Payment ID	1Ah2hOAzWbtAUAz9WbdzbQYf4 Qvb
Transaction authorised	Yes
Transaction uncertain	No
Transaction paid	Yes
Alias	xxxx xxxx xxxx 0111
Card number	xxxx xxxx xxxx 0111
Card expiry date	12/2015
Card Type	[redacted]
3D Secure	Successful

Capture transaction

Amount to capture 100 Maximal capturable amount:100

The capture will automatically close the transaction for further captures

Previous Actions

Date	Action	Message

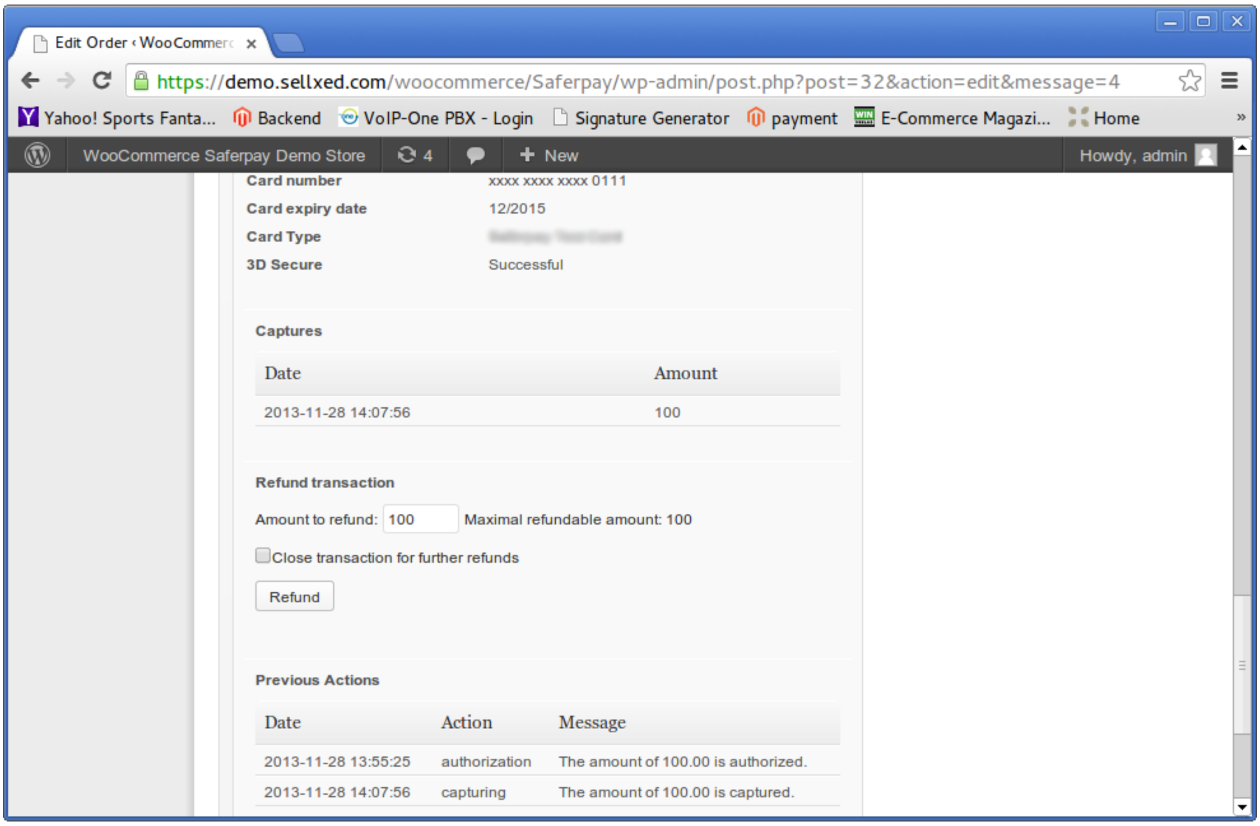
Figure 6.1: Cancelling and capturing orders

> ## Partial Capturing
>
> Please inquire, whether multiple captures are supported within your Dimoco 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.



Card number	xxxx xxxx xxxx 0111
Card expiry date	12/2015
Card Type	[redacted]
3D Secure	Successful

Captures

Date	Amount
2013-11-28 14:07:56	100

Refund transaction

Amount to refund: 100 Maximal refundable amount: 100

□ Close transaction for further refunds

Refund

Previous Actions

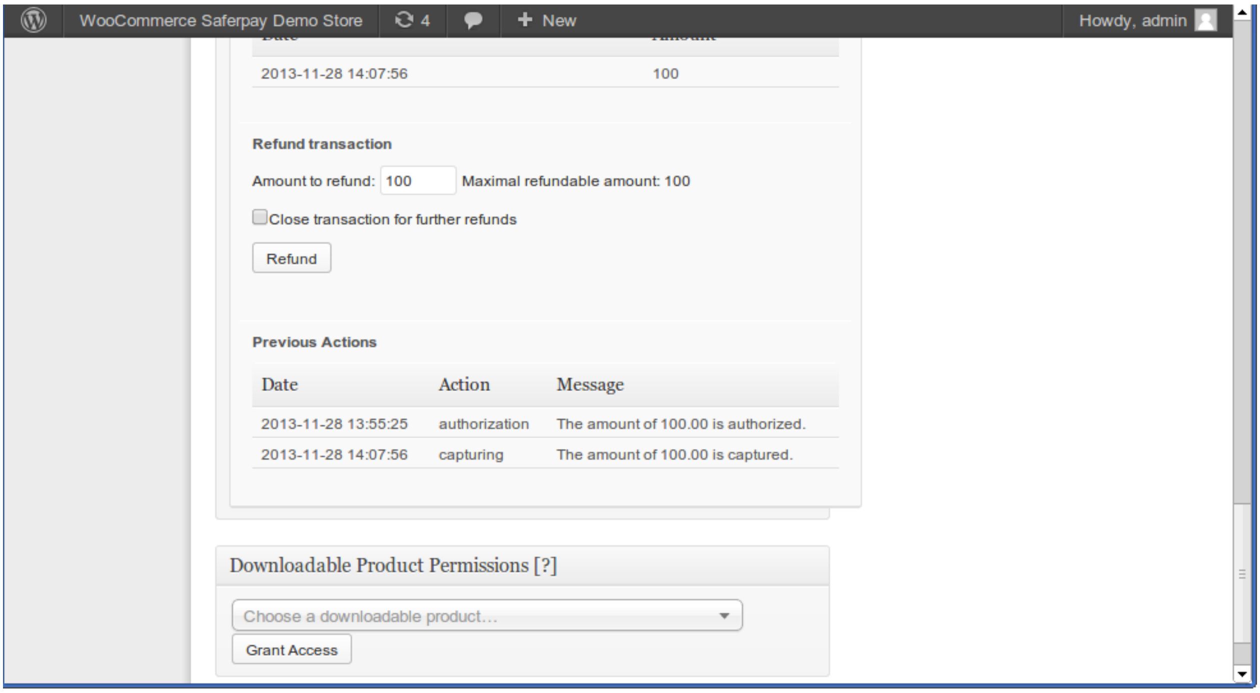
Date	Action	Message
2013-11-28 13:55:25	authorization	The amount of 100.00 is authorized.
2013-11-28 14:07:56	capturing	The amount of 100.00 is captured.

Figure 6.1: Refunds in WooCommerce for Dimoco.

> ### 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.



Date	Amount
2013-11-28 14:07:56	100

Refund transaction

Amount to refund: 100 Maximal refundable amount: 100

□ Close transaction for further refunds

Refund

Previous Actions

Date	Action	Message
2013-11-28 13:55:25	authorization	The amount of 100.00 is authorized.
2013-11-28 14:07:56	capturing	The amount of 100.00 is captured.

Downloadable Product Permissions [?]

Choose a downloadable product…

Grant Access

Figure 6.1: Transaction overview in the transaction history.

6.6 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 Dimoco 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.

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 Card

Card number	4111 1111 1111 1111	Visa Success
Expiry Date	12/2020	
CVC	123	
Card number	4242 4242 4242 4242	Visa Failure
Expiry Date	12/2020	
CVC	123	
Card number	5555 5555 5555 4444	MasterCard Success
Expiry Date	12/2020	
CVC	123	
MasterCard (Failure)	5105 1051 0510 5100	MasterCard Failure
Expiry Date	12/2020	
CVC	123	

SOFORT

Instructions	Set the last name of the customer to the given value to simulate an outcome.	Simulation control
ignore	Do not simulate any behaviour	
success	The transaction succeeds immediately	
failed	The transaction fails immediately	
redirectSuccess	A redirection occurs, and then the transaction succeeeds	
redirectError	A redirection occurs, and then the transaction fails	
pending	The transaction stays pending	
html	Submit a given form, then the transaction succeeds	
htmlError	Submit a given form, then the transaction fails	

giropay

Instructions	Set the last name of the customer to the given value to simulate an outcome.	Simulation control
ignore	Do not simulate any behaviour	
success	The transaction succeeds immediately	
failed	The transaction fails immediately	
redirectSuccess	A redirection occurs, and then the transaction succeeeds	
redirectError	A redirection occurs, and then the transaction fails	
pending	The transaction stays pending	
html	Submit a given form, then the transaction succeeds	
htmlError	Submit a given form, then the transaction fails	

EPS

Instructions	Set the last name of the customer to the given value to simulate an outcome.	Simulation control
ignore	Do not simulate any behaviour	
success	The transaction succeeds immediately	
failed	The transaction fails immediately	
redirectSuccess	A redirection occurs, and then the transaction succeeeds	
redirectError	A redirection occurs, and then the transaction fails	
pending	The transaction stays pending	
html	Submit a given form, then the transaction succeeds	
htmlError	Submit a given form, then the transaction fails	

paysafecard

Instructions	Set the last name of the customer to the given value to simulate an outcome.	Simulation control
ignore	Do not simulate any behaviour	
success	The transaction succeeds immediately	
failed	The transaction fails immediately	
redirectSuccess	A redirection occurs, and then the transaction succeeeds	

redirectError	A redirection occurs, and then the transaction fails	
pending	The transaction stays pending	
html	Submit a given form, then the transaction succeeds	
htmlError	Submit a given form, then the transaction fails	

Przelewy24

Instructions	Set the last name of the customer to the given value to simulate an outcome.	Simulation control
ignore	Do not simulate any behaviour	
success	The transaction succeeds immediately	
failed	The transaction fails immediately	
redirectSuccess	A redirection occurs, and then the transaction succeeeds	
redirectError	A redirection occurs, and then the transaction fails	
pending	The transaction stays pending	
html	Submit a given form, then the transaction succeeds	
htmlError	Submit a given form, then the transaction fails	

iDEAL

Instructions	Set the last name of the customer to the given value to simulate an outcome.	Simulation control
ignore	Do not simulate any behaviour	
success	The transaction succeeds immediately	
failed	The transaction fails immediately	
redirectSuccess	A redirection occurs, and then the transaction succeeeds	
redirectError	A redirection occurs, and then the transaction fails	
pending	The transaction stays pending	
html	Submit a given form, then the transaction succeeds	
htmlError	Submit a given form, then the transaction fails	

Sepa Direct Debits

		Simulation control

Instructions	Set the last name of the customer to the given value to simulate an outcome.
ignore	Do not simulate any behaviour
success	The transaction succeeds immediately
failed	The transaction fails immediately
redirectSuccess	A redirection occurs, and then the transaction succeeeds
redirectError	A redirection occurs, and then the transaction fails
pending	The transaction stays pending
html	Submit a given form, then the transaction succeeds
htmlError	Submit a given form, then the transaction fails

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 The Referrer URL appears in my Analytics Tool

When a customer and the notification are redirected via Header Redirection, the Dimoco 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 Dimoco 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_dimococw/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_dimococw/classes
/DimocoCw/Util.php';
```

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

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

Load the transaction by the external Id:

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

Load transactions by the order id:

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