

Installation Manual for PayUnity Magento

This manual describes the installation and usage of the PayUnity extension for Magento.

Release Date: Mon, 07 Aug 2023 14:01:06 +0200
Version: 6.0.55

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	6
1.1	Installation Procedure	6
1.2	System Requirements	7
2	Configuration	8
2.1	Basic configuration of the main module	8
2.2	Configuration of the Payment Methods	8
2.3	Configuration Webhook	10
2.3.1	Configuration Webhook by Merchant	11
2.3.2	Configuration Webhook by PayUnity	11
2.4	Activation and Testing	12
3	Module Installation and Update in the Magento Shop	13
3.1	Installation	13
3.1.1	Magento 1	13
3.1.2	Magento 2	13
3.2	Updates and Upgrades	14
3.2.1	Update Checklist	15
3.2.2	Update Instructions	15
3.3	Database Migration	16
3.3.1	Magento 1	16
3.3.2	Magento 2	16
4	Module Configuration in the Magento Shop	17
4.1	Configuration of the Main Module	17
4.1.1	Magento 1	17
4.1.2	Magento 2	17
4.2	Configuration of the Payment Method	18
4.2.1	Magento 1	18
4.2.2	Magento 2	19
4.3	Direct Capturing of Transactions	19
4.4	Uncertain Status	19
4.4.1	Setting the order state	19
4.5	Optional: Validation	20
4.5.1	Usage of the Integrated Multishop Functionality of Magento	20
5	Settings / Configuration of Payment Methods	21
5.1	General Information About the Payment Methods	21

5.2	Information on Payment Status	21
5.2.1	Order status "pending" / imminent payment (or similar)	21
5.2.2	Order status "cancelled"	21
5.3	Klarna	22
5.3.1	Supported / Unsupported Functions	22
5.3.2	Set Up / Configuration of the Payment Method	22
5.3.3	Canceling the Invoice	22
5.3.4	Full Activating Invoices	23
5.3.5	Partial Refund and Partial Activation	23
5.3.6	On Hold Orders (Pending)	23
5.3.7	Testing	23
5.3.8	Payment fees	23
6	The Module in Action	24
6.1	Capturing / Cancelling of Orders	24
6.1.1	Magento 1	24
6.1.2	Magento 2	24
6.2	Place Orders in the backend of Magento	26
6.2.1	Magento 1	26
6.2.2	Magento 2	27
6.3	Refunds	27
6.3.1	Magento 1	27
6.3.2	Magento 2	29
6.4	Useful Transaction Information on the Order	32
6.4.1	Magento 1	32
6.4.2	Magento 2	33
6.5	Usage of the Alias Managers / Token Solution	34
6.5.1	Magento 1	34
6.5.2	Magento 2	35
6.6	Tax Calculation Settings	37
6.6.1	Magento 1	37
6.7	OneStepCheckout	37
6.7.1	Magento 1	37
6.7.2	Magento 2	38
6.8	Setup a Cron Job to Activate the Timed Operations	38
6.8.1	Magento 1	38
6.8.2	Magento 2	38
6.9	Magento 1 API Manual	38
6.9.1	General Information on Using the API	39
6.9.2	API Calls	39
7	Testing	40
7.1	Test Data	40

8	Errors and their Solutions	43
8.1	Module is not shown	43
8.2	The Referrer URL appears in my Analytics Tool	43
9	Compatibility with Third-Party Plugins	44
9.1	Magento 1	44
9.1.1	Fees and discounts within Magento	44
9.2	Birthday and gender in Magento 1	44
9.3	Birthday and gender in Magento 2	44
10	Error Logging	45
10.1	Log Levels	45
10.2	Log Location	45
11	Advanced Information	46
11.1	Transaction Object	46
11.1.1	Magento 1	46
11.1.2	Magento 2	46

1 Introduction

This manual explains the installation, configuration and usage of the payment module for Magento and PayUnity.

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 PayUnity
- Magento payment module by sellxed.com/shop
- Access data to your server and shop

In case you don't yet have a contract with PayUnity, you can acquire it directly through us.

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 Procedure

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 a secure usage in correspondence with all security regulations can be guaranteed.

1. Configuration of the test environment by means of the integration data from PayUnity. These can be found on the test platform under <https://test.payunity.com/bip/login>
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://payunity.com/bip/login>

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 PayUnity might not get through to the shop.

1.2 System Requirements

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

- PHP Version: 5.4.x or higher
- Magento: 1.7.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

2.1 Basic configuration of the main module

The access data for the test environment (<https://test.payunity.com/bip/login>) is provided to you by PayUnity.

Navigate to the [main module](#) in your Shop and fill in the following access data:

- Authorization Bearer Token (Access Token)
- Entity ID (CHANNEL ID)
- The User ID and Password are deprecated and not required anymore, use the new Authorization Bearer Token instead.
- You no longer need the parameter **Security Sender and Security and Hash Secret** for the integration. You can simply ignore this entry.
- The boxes for "Security Hash" and "Security Hash Secret" can be left empty unless your Payment Service Provider instructs you to do otherwise

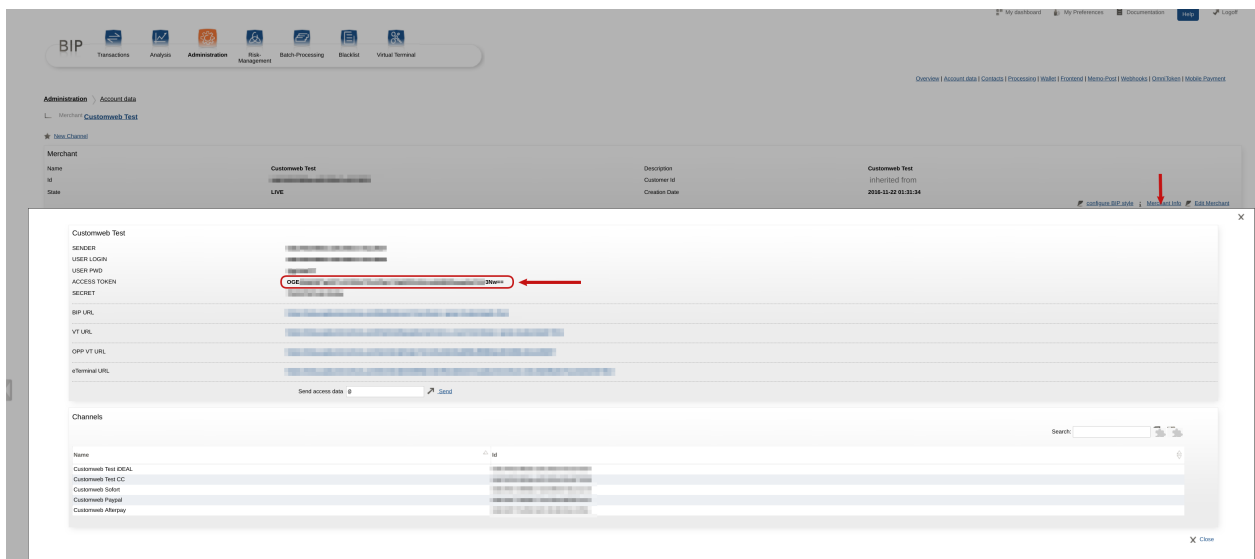


Figure 2.1: The Authorization Bearer Token in the PayUnity backend.

You also have more entry options. For example the **Custom Parameters**. In principle you leave these blank unless PayUnity requires you to do otherwise.

2.2 Configuration of the Payment Methods

In order to activate the payment methods and to change other payment specific settings, navigate to the settings. (A description as to where to find these settings is listed here [hier](#))

In the configuration of the payment methods you can define which Entity ID (Channel ID) should be used for which amount. This allows you to use a non-3D Secure Channel in case of small figures. In order for you to be able to set up and get the corresponding contract please contact PayUnity.

Channel Conditions

the unit which comes transactions into the system.

[STORE VIEW]

▲ Beside the default channel ID a set of channel IDs depending on the order total amount can be defined. Each line must contain a lower amount, an upper amount and the applicable Channel ID. The format is as follow 'lower amount;upper amount;Channel ID' as shown in the following example: '10.00;200.00;23413113213131231353'. The upper boundaries are not included. If you leave this field empty or if an amount is outside any range the default channel id is applied. If multiple conditions match the last one is taken.

Entity ID (MoTo)

[STORE VIEW]

▲ If this payment method should support recurring and moto transactions, you have to specify a second entity ID without 3D secure.

COPYandPAY Style

[STORE VIEW]

▲ Define the style to be used for the COPYandPAY payment form.

Authorised status

[STORE VIEW]

▲ This status is set when the payment was successful and it is authorised.

Uncertain status

[STORE VIEW]

▲ You can specify the order status for new orders that have an uncertain authorisation status.

Captured status

[STORE VIEW]

▲ You can specify the order status for orders that are captured either directly after the order or manually in the back-end.

Approved payment Status

[STORE VIEW]

▲ You can specify the order status for orders that are approved after being in an uncertain state.

Denied Payment Status

[STORE VIEW]

▲ You can specify the order status for orders that are denied after being in an uncertain state.

Authorisation Method

[STORE VIEW]

▲ Select the authorisation method to use in order to process this payment method.

Figure 2.1: Configuration of the payment method using the example Magento (icon image, can differ from your version of the image).

PCI: New Authorization Method **Widget**

Please note that the use of the Hidden Mode comes with additional certification requirements (compare with our [blog entry](#) regarding this subject). For this reason we will no longer supply our new versions with the hidden mode. If you still use the hidden mode in you payment methods please solely use **Widget**. The support of the hidden mode will be terminated shortly.

2.3 Configuration Webhook

In some special cases it may happen that the payment notification sent by PayUnity can not be proceed by your Magento shop. We recommend to configure a so called webhook in the PayUnity backen under **Administration > Webhook**. Please make sure that you configure the correct URL which you can find in your Magento shop under **System > PayUnity for Magento 1 and under Stores > PayUnity for Magento 2 > Setup**.

2.3.1 Configuration Webhook by Merchant

If you have to configure the the webhook on your own, you can find the webhook URL in your Magento shop at **System > PayUnity for Magento 1 and under Stores > PayUnity for Magento 2 > Setup** . Make sure you configure it in the PayUnity Backend under **Administration> Webhook**. Furthermore you have to define a so called "secret for encryption" key. Make sure that you also save the same values in the corresponding fields, as on the Picture below.

Customweb "secret for encryption" Generator

Please note that only ASCII characters may be used for the generation of this key pass phrases. Please use our ["secret for encryption" generator](#) so that you do not enter invalid characters.

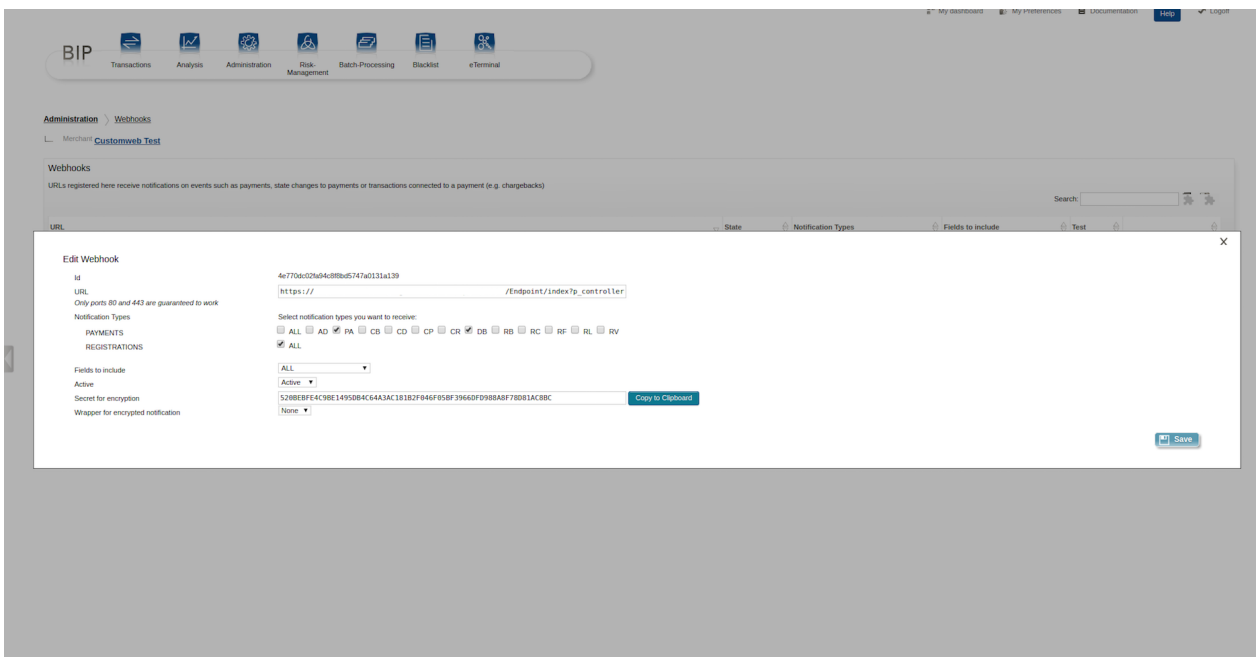


Figure 2.1: PayUnity webhook configuration in the backend under **Administration > Webhook**.

2.3.2 Configuration Webhook by PayUnity

In some PayUnity accounts you can not configure the webhook by yourself. Therefore you have to provide the webhook URL to PayUnity once they setup it for you, they will provide you a specific "secret for encryption" key, which you have to configure in the main module configuration of the module in your shop.

2.4 Activation and Testing

When you have activated and configured the payment methods you can run a test with help from the manual [Test data](#).

If the tests were successful you now switch the operation mode from test mode to "Live Mode" and replace the safety features above with your personal access data for the live platform. You will receive this data directly from your Payment Service Provider. Normally the USER ID etc. should be identical for the Live Mode.

3 Module Installation and Update in the Magento 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.

In the following you will find the installation manual for Magento 1 as well as Magento 2

3.1.1 Magento 1

Compiler Cache / Caching

Before you start the installation or after doing an update of the module, please make sure, that the compiler cache is disabled. Also make sure that you cleared the cache otherwise it could be that the settings page is not visible.

1. Create a back-up of your shop.
2. Extract the content of the ZIP file.
3. Copy the **entire content** of the folder "files" into the main directory of the server of your shop respectively to the folder where the plugins are copied normally. In order to do so, use your usual ftp program. Make sure that the folders aren't replaced but merely merged.

3.1.2 Magento 2

Im Folgenden werden die Schritte für die Installation und das Update der Extensions erklärt.

3.1.2.1 Installations Requirements

Bitte beachten Sie, folgende Voraussetzungen sind für den Betrieb von Magento 2 zwingend zu erfüllen:

1. System requirements von Magento (<http://devdocs.magento.com/guides/v2.0/install-gde/system-requirements.html>)
2. Der Cron Job muss korrekt eingerichtet sein, sonst können keine Extensions installiert werden. Weitere Informationen finden Sie hier: <http://devdocs.magento.com/guides/v2.0/config-guide/cli/config-cli-subcommands-cron.html>

3.1.2.2 Installation

For the installation of the plugin in your Magento 2 store please follow these steps:

1. Readiness Check

Before uploading the plugin it is necessary to do the Readiness Check for Magento 2

1. Save [this script](#) as cw-setup.php on your computer.
2. Upload this cw-setup.php Script into the root directory of your Magento 2 installation.
3. Access your server via SSH and run this command: `php cw-setup.php`
4. The script will present you with additional commands which you need to execute.

About the cw-setup.php script

The provided script generates a set of commands which should be executed to clean up Magento 2. The script itself doesn't execute anything. It only generates the necessary commands.

2. Upload

Upload the content of the folder **files_2** into the root directory of your server. Execute the commands provided by the cw-setup.php script again.

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.

3.2.2.1 Magento 1

1. Download the newest version of the plugin. The download is available in your sellxed.com account under "My Downloads".
2. Extract the ZIP file.
3. Upload the folder **files_1** into the root directory of your shop.
4. Clear the Magento cache

3.2.2.2 Magento 2

1. Download the newest version of the plugin. The download is available in your sellxed.com account under "My Downloads".
2. Extract the ZIP file.
3. Upload the folder **files_2** into the root directory of your shop.
4. Clear the Magento cache

3.3 Database Migration

This section describes how you can run the database migration script

This only needs to be done if your Magento installation shows error messages regarding missing tables or columns in the database.

3.3.1 Magento 1

Clear the Magento cache

If this does not resolve the issue, remove the entry `payunitycw_setup` from the `core_resource` table and clear the cache again.

3.3.2 Magento 2

From the command line run the following command:

```
php bin/magento setup:upgrade
```


4 Module Configuration in the Magento 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

4.1.1 Magento 1

The settings for the main module can be found under **System > Configuration > PayUnity**. Enter the various options, some of which you have already specified in the configuration with PayUnity. Information on the functionalities as well as the impact of the individual settings can be found directly by the option.

The field "Time-Out for pending Payments" defines after how many minutes the order status should switch from pending to cancelled (0 means that nothing should be changed) if no successful feedback concerning the payment is received from PayUnity. This might occur e.g. if the customer has closed his or her browser. Further information on this topic can be found under [Order Status](#).

4.1.2 Magento 2

You will find the settings of the main module via **Stores > Configuration > Sales > PayUnity > General**.

Fill in the fields according to the configuration of PayUnity or with the data which you received directly from PayUnity. You can find more exact explanations regarding each option in the help text by the input boxes.

Shop



Kunden Alias Management	<input type="text" value="Nein"/>	[STORE VIEW]
	Sie können Ihren Kunden gestatten ihre Aliasse auf ihrem Konto zu verwalten/löschen.	
Benachrichtigung abwarten	<input type="text" value="Nein"/>	[WEBSITE]
	Es kann in einigen Fällen ein paar Sekunden dauern bis die Transaktion erfolgreich genehmigt wird. Wenn diese Option aktiviert ist, wird die Benachrichtigung erwartet.	

General



Betriebsmodus	<input type="text" value="Test Modus"/>	[STORE VIEW]
	If the test mode is selected the test account id is used. Otherwise the configured live account id is used.	
Live account ID	<input type="text"/>	[STORE VIEW]
	The live account ID as given by the Salesway .	
Live API password	<input type="text"/>	[STORE VIEW]
	The live API password as given by Salesway .	
Test Account ID	<input type="text"/>	[STORE VIEW]
	The test account ID as given by the Salesway . Generic test account: 401860-17795278	
Test API password	<input type="text"/>	[STORE VIEW]
	The test API password as given by Salesway . Password for generic test account: 8e7Yn5yk	
Liability Shift Behavior	<input type="text" value="Mark transactions without liability shift as uncertain"/>	[STORE VIEW]
	If no liability shift is given by a transaction, should the transaction be marked as uncertain?	
Bestell-Präfix	<input type="text"/>	[STORE VIEW]
	Here you can insert an order prefix. The prefix allows you to change the order number that is transmitted to Salesway . The prefix must contain the tag {id}. It will then be replaced by the order number (e.g. name_{id}).	

Figure 4.1: Magento 2: Configuration of the basic module. Please note, that this is an example and the settings can differ.

4.2 Configuration of the Payment Method

4.2.1 Magento 1

After having successfully installed the main module, you will find the individual payment modules under System > Configuration > Payment Methods. You can save 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.

4.2.1.1 Direct Capturing of Invoices

The "Invoice Creation" specifies if invoices should be captured directly or not. Captured invoices can no longer be modified. If you wish to be able to modify the ordered article for debiting, please set this option to "Deferred".

4.2.2 Magento 2

After having successfully configured the main module, you will find the settings for each payment method in your shop via Stores > Sales > Payment Methods. For each payment method you can change the settings individually to optimize the payment processing in your shop.

4.2.2.1 Accounting

For each payment method you need to choose whether you want to register directly or delayed. A delayed booking only reserves the amount. Therefore it is important that you follow the instructions for such transactions as listed below. Reservations are usually only valid for a few days so we recommend to register directly. The delayed booking isn't available for every payment method.

4.2.2.2 Authorization Method

For some payment methods you can choose between different authorization methods. The authorization method determines how the payment form is presented in your shop. Please note that for some authorization methods with credit cards there are extended PCI requirements. (<https://www.sellxed.com/en/blog/pci-300-changes>).

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 PayUnity and the module

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

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

4.5.1 Usage of the Integrated Multishop Functionality of Magento

The payment module supports the multishop feature of Magento. No further modifications are necessary. The module automatically recognises the shop the order belongs to. In order for the multishop functionality to work, it is, however, necessary that the individual sub-shops within Magento have been configured correctly.

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

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 Klarna

You can process Klarna directly via PayUnity. In the following we will expand on the characteristics of the Klarna setup.

5.3.1 Supported / Unsupported Functions

The module does not support the following functions:

- You can not create or change any orders in the backend of Magento .
- Partial activation and refunds can only be performed through the backend of PayUnity (<https://payunity.com/bip/login>) Full activation and full refunds can be performed directly inside Magento. For more information visit the Chapter around [transaction management](#).

Edit Klarna transaction within Magento

Please make sure that you **never** modify or edit any Klarna transaction in your Magento backend.

The remaining functions are supported by the module. In the following you will find a description of the most important functions.

5.3.2 Set Up / Configuration of the Payment Method

Activate the payment method as usual. Please note the following regarding the installation:

5.3.2.1 Displayed Name of the Payment Method

You may change the displayed name in the frontend via **title**. Please use the title **Invoice**. Furthermore, you can define whether the Klarna logo is visible to your customers in the frontend via **Display Logo**.

5.3.2.2 Description of the Payment Method

You may change the description of the payment method in the frontend via **description**. Please make sure you display the following description in your frontend: "**Pay within 14 days**".

5.3.2.3 Select Authorization Method

We recommend to set **Server** as authorization method. In this case you will not be redirected to the payment page of PayUnity.

5.3.2.4 Klarna Merchant ID

In the payment method you can also define the **Klarna Merchant ID (EID)**. You will either find this in your Klarna account or ask for it directly at PayUnity.

5.3.3 Canceling the Invoice

How to cancel invoices will be explained below in the chapter Activating / Canceling Orders. For further questions please refer to this section.

5.3.4 Full Activating Invoices

For more information visit the Chapter around [transaction management](#).

5.3.5 Partial Refund and Partial Activation

Partial refunds and partial activations can only be done through the [BIP](#) of PayUnity.

5.3.6 On Hold Orders (Pending)

Pending orders are not supported by the module. In order you want those to be handled correctly by the system, please leave the field "Status Check interval" blank in your BIP. Please contact PayUnity to make this setting.

5.3.7 Testing

In order for you to test the payment method Klarna you will need specific test data. But first make sure the the **operation mode is on test and the test mode is on external**.

You will find the official test data for Klarna directly in the developer portal via this link: <https://developers.klarna.com/en/de+php/kpm/test-credentials>.

5.3.8 Payment fees

If you want to charge payment fees for a specific payment method, feel free to use the compatible [payment fees](#) modules. Those fees are directly transmitted to Klarna.

Payment Fees

Please make sure to not charge payment fees within the PayUnity [BIP Backend](#). Please use exclusively the compatible modules as described above.

6 The Module in Action

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

6.1 Capturing / Cancelling of Orders

6.1.1 Magento 1

Please note:

The transaction management between your shop and PayUnity is not synchronised. If you capture payments with PayUnity, the status in the shop will not be updated and a second capturing in the shop is not possible.

6.1.1.1 Case 1: Invoice Capturing Directly After the Order

In the following, the process is explained for the case that you have chosen to directly create the invoice after having placed the order and only authorise payments - not capture them directly - (further information on the configuration can be found under [Creating the Invoice](#)).

In order to capture the order, open the invoice and click "capture". If you click "Cancel", the authorisation will be cancelled.

6.1.1.2 Case 2: Deferred Invoice Capturing

In case you have chosen the deferred invoice creation, you first have to create the invoice by clicking on the button "Invoice". Select "Capture online" below the total amount and then click on "capture invoice". The invoice has now been captured.

6.1.1.3 Cancel of orders/transactions

In order to cancel a transaction you first have to open the invoice and click on the cancel button. After that open the order and click on cancel here again. This will send the cancel of the order to PayUnity. Please note that you need to cancel the invoice and the order.

6.1.2 Magento 2

6.1.2.1 Capture Bug in Magento (v. 2.0.0)

In version 2.0.0 of Magento there is a bug. It could be that invoices cannot be captured. Please make the following adjustments in the file `/vendor/magento/module-sales/Controller/Adminhtml/Order/Invoice/Capture.php`:

1. Delete the lines in the picture bellow which begin with a - .

2. Add the lines which has been marked with a + .

Here you can find the codes to be inserted in the form of a text:

```
+ )->addComment( __('The credit memo has been created automatically.'));
```

Method: execute()

Diff:

```

-     try {
-         $invoiceManagement = $this->_objectManager->get('Magento\Sales\Api\InvoiceManagementInterface');
-         $invoiceManagement->setCapture($invoice->getEntityId());
+         $invoice->capture();
+         $invoice->getOrder()->setIsInProcess(true);

```

Figure 6.1: Opening the Invoice

6.1.2.2 Performing Captures

Please note

The transaction management between PayUnity and your shop isn't synchronized. When you capture the payments at PayUnity , the status is not adjusted and a second capture isn't possible in your shop.

In the configuration of the payment method you can decide whether the invoice should be captured after having been created or if only the payment is authorized. You also decide if you already want to create an invoice. If you create an invoice you can't make any adjustments for the capture.

In order to capture invoices please take the following steps:

1. Open the order and create an invoice.
2. If you have set the Invoice Settlement to deferred you can still make adjustments to the invoice.
3. Click on Capture Online.

Invoice Settlement	Deferred settlement ▼	[WEBSITE]
	Choose settlement after order in case you want an invoice to be created with your order. The state of the invoice will set according to your capture status.	
Customer Address	No address is sent ▼	[STORE VIEW]
	Should the customer address be sent?	
Capturing	Deferred ▼	[STORE VIEW]
	Should the amount be captured automatically after the order (direct) or should the amount only be reserved (deferred)?	
Authorisation Method	Payment Page ▼	[STORE VIEW]
	Select the authorisation method to use in order to process this payment method.	
Alias Manager	Inactive ▼	[STORE VIEW]
	The alias manager allows the customer to select from a credit card previously stored.	

Figure 6.1: Capture options

6.1.2.3 Cancelling / Voiding Orders

In order to issue a cancellation of the payment you must use the "Void" button. If you wish to use the "Cancel" button you must cancel the payment in the PayUnity backend portal.

6.2 Place Orders in the backend of Magento

6.2.1 Magneto 1

With the Magento payment module you can place orders and capture the payment directly in the shop; so called Mail Order / Telephone Order (MOTO).

1. Go to the order overview and click on "Create New Order"
2. The usual window for the creation of orders will be opened. Select the customer in question, the item and the address
3. Select the preferred payment method and click on "Submit Order".
4. Depending on the authorisation method that you have saved for the selected payment method, you will either be redirected to the Payment Page of PayUnity or the mask for the credit card will appear such as in the image below. Enter the credit card data of the customer.

Mail order/telephone order authorization

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

Card number *
Please enter your card number

Card expiration date *
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: MoTo Transactions in Magento

6.2.2 Magento 2

With the Magento payment module you can trigger orders directly from your shop, so called Mail Order / Telephone Order (MOTO).

1. Go to the overview of the order (Sales > Orders) and click on "Create New Order" at the top right.
2. The familiar window to create an order will open. Choose the desired customer, article and address.
3. In the payment method box, choose the desired payment method and click on "Finalize order".
4. After that you will be redirected either to the Payment Page of PayUnity or you will be asked to enter your payment information in the shop, depending on your settings.

If an alias of the customer exists, you can alternatively use that for the order.

6.3 Refunds

6.3.1 Magento 1

You can refund already captured transactions and automatically transmit them to PayUnity. In order to do so, open the invoice of the captured order. By clicking on "Credit Memo" the window for refunds is opened. Enter the amount you wish to refund. By clicking on "Refund" the refund is transmitted to PayUnity.

Below you find a step-by-step guide including screenshots:

Order View Order # 200000003 | 17.04.2014 16:40:42 Back Edit Send Email Credit Memo Hold Ship Reorder

Information Page 1 of 1 pages | View 20 per page | Total 1 records found Reset Filter Search

Invoice #	Bill to Name	Invoice Date	Status	Betrag
200000003	Thomas Lueth	17.04.2014 16:41:41	Paid	20,99 €

Figure 6.1: 1. open the invoice

Invoice #100000026 | Paid | 23.04.2014 18:52:08 (the invoice email is not sent) Back Send Email Credit Memo Print

Order # 100000043 (the order confirmation email was sent)		Account Information	
Order Date	23.04.2014 18:51:23	Customer Name	Test Test
Order Status	Processing	Email	info@customweb.com
Purchased From	Main Website Main Store English	Customer Group	General

Figure 6.1: 2. click on credit memo - Important - open the invoice first!.

Refund Totals

Subtotal	156,00 €
Refund Shipping	<input type="text" value="5"/>
Adjustment Refund	<input type="text" value="0"/>
Adjustment Fee	<input type="text" value="0"/>
Grand Total	161,00 €

Append Comments
 Email Copy of Credit Memo

Figure 6.1: 3. Adjust the refunding amount (see more info below [Refunds Based on Quantity](#))

Refund Totals

Subtotal	156,00 €
Refund Shipping	<input type="text" value="5"/>
Adjustment Refund	<input type="text" value="0"/>
Adjustment Fee	<input type="text" value="0"/>
Grand Total	161,00 €

Append Comments
 Email Copy of Credit Memo

Figure 6.1: 4. To process the refund click on Refund.

✔ The credit memo has been created.

Order # 100000041 | 23.04.2014 18:26:16 Back Send Email Reorder

Order # 100000041 (the order confirmation email was sent)		Account Information	
Order Date	23.04.2014 18:26:16	Customer Name	Test Test
Order Status	Closed	Email	info@customweb.com
Purchased From	Main Website Main Store English	Customer Group	General
Placed from IP	160.85.155.47		

Figure 6.1: Wait for the confirmation of the refund.

6.3.1.1 Refunds Based on Quantity

You can make refunds based on product quantity. In order to do so, open the credit memo window and enter the amount of products you wish to refund to your customer's card

You can carry out as many individual refunds as you wish until you have reached the quantity of the original order. The status of the order then automatically switches to **complete**.

Credit Memo #	Bill to Name	Created At	Status	Refunded
200000001	Leandra Simitovic	02.12.2013 16:42:10	Refunded	35,00 €
200000002	Leandra Simitovic	02.12.2013 16:47:03	Refunded	146,00 €

Figure 6.1: Overview of all Refunds

6.3.1.2 Refunding of Any Amount

As each position of the order needs to be transmitted to PayUnity we can't support the function **Adjustment Refund** - meaning the refund of any amount. Only whole items can be refunded by adjusting the refund amount.

Adjusting the amount of several refunds

It is possible to carry out several refunds on an order. Please note that you can refund a maximum of 100% of the original amount. If you have already carried out a refund you need to make sure not to refund more than the initial amount by adjusting the amount and the Adjustment Fee.

6.3.2 Magento 2

6.3.2.1 Resolving a Current Magento Bug (v. 2.0.0)

In the 2.x version of Magento there is a bug. Please make the following adjustments to the file `/vendor/magento/module-sales/Model/Order/Payment.php` :

1. Delete the lines in the picture bellow which begin with a - .
2. Add the lines which has been marked with a + .

Method: registerRefundNotification(\$amount)

Diff:

```

$creditmemo->setPaymentRefundDisallowed(
    true
)->setAutomaticallyCreated(
    true
- )->register()->addComment(
+ )->addComment(
    __('The credit memo has been created automatically.')
);

```

Figure 6.1: Adjustments so you are able to enter a credit memo.

Here you can find the codes to be inserted in the form of a text:

```
+ )->addComment( __('The credit memo has been created automatically.'));
```

6.3.2.2 Performing Credits

For registered transactions you can create credits and transmit them automatically to PayUnity . In order to do so open an invoice of an already registered order. By clicking on "Credit Memo" a dialog will open for the refunds. Now you can adjust the number of products which you would like to credit in the credit memo dialogue.

Here you can find each step again including screenshots:

#000000003 🔍 🔔 👤 admin ▾

← Back Send Email Credit Memo Hold Ship

ORDER VIEW

- Information
- Invoices
- Credit Memos
- Shipments
- Comments History
- Transactions

Search by keyword 🔍
🔽 Filters
👁 Default View ▾
⚙ Columns ▾
📄 Export ▾

Select It... ▾ 1 records found 20 ▾ per page < 1 of 1 >

	Invoice ↓	Invoice Date	Order Number	Order Date	Bill-to Name	State	Amount	Action
<input type="checkbox"/>	000000003	Dec 14, 2015 7:55:37 AM	000000003	Dec 14, 2015 7:54:41 AM	Test Customweb	Paid	CHF39.00	View

Figure 6.1: 1. Opening the Invoice

New Memo for #000000003

admin ▾

[← Back](#)
[Reset](#)

Order & Account Information

Order # 000000003 (The order confirmation email was sent)

Order Date	Dec 14, 2015, 7:54:41 AM
Order Status	Processing
Purchased From	Main Website Main Website Store Default Store View
Placed from IP	212.90.201.186

Account Information

Customer Name	Guest
Email	info@customweb.com
Customer Group	NOT LOGGED IN

Address Information

Billing Address [Edit](#)

Test Customweb
 Teststrasse 2
 Zürich, Zug, 8032
 Switzerland
 T: 555466684

Delivery address [Edit](#)

Test Customweb
 Teststrasse 2
 Zürich, Zug, 8032
 Switzerland
 T: 555466684

Figure 6.1: 2. Click on Credit Memo - Important the invoice needs to be opened first.

Refund Totals

	Subtotal	156,00 €
	Refund Shipping	<input type="text" value="5"/>
	Adjustment Refund	<input type="text" value="0"/>
	Adjustment Fee	<input type="text" value="0"/>
	Grand Total	161,00 €

Append Comments
 Email Copy of Credit Memo

Figure 6.1: 3. Choose which products you would like to credit and click on Refund Online.

#000000003

🔍 🔔 👤 admin ▾

← Back Send Email Credit Memo Hold Ship

ORDER VIEW

Information

Invoices

Credit Memos

Shipments

Comments History

Order History

Transactions

🔍
Filters
👁️ Default View ▾
⚙️ Columns ▾
📄 Export ▾

Select It... ▾
0 records found
20 ▾ per page
<
1
of 1
>

	Credit Memo	Created	Order	Order Date	Bill-to Name	State	Refunded	Action
We couldn't find any records.								

Figure 6.1: Overview of all the credits made

Adjusting the amounts for several refunds

It is possible to issue several credits on one order. Please note that you can't credit more than 100% of the original amount. If you have already issued a credit you will need to make sure you don't credit more than the original amount, by adjusting the amount and the Adjustment Fee.

6.4 Useful Transaction Information on the Order

6.4.1 Magento 1

In each order, processed via our module, you can find an overview of the most important information about the transaction as well as a transaction history.


Payment Information		
Authorisation amount	161.0000	
Currency	CHF	
Payment method	MasterCard	
Payment ID	23679571	
Captured amount	161	
Transaction authorised	Yes	
Transaction uncertain 	No	
Transaction captured	Yes	
Transaction paid	Yes	
Alias	XXXXXXXXXXXX0007 (12/13)	
Alias Token	8B1974E9-59EB-442C-A1D4-34BEC568EE23	
Acceptance	test123	
Card number	XXXXXXXXXXXX0007	
Card expiry date	12/13	
Merchant reference	sellxed_og_10000087	
Transaction history		
Date	Action	Message
2013-09-12 12:53:50	authorization	The amount of 161.00 is authorized.
2013-09-12 12:53:50	capturing	The amount of 161.00 is captured.
Order was placed using CHF		

Figure 6.1: Transaction Information.

6.4.2 Magento 2

In each order, which was processed by our module, you will find an overview of the most important information on the transaction and transaction history.

Payment & Shipping Method

Payment information



Visa

Transaction ID	14
Authorisation Amount	107
Currency	CHF
Payment period	Visa
Payment ID	h1ElhGAKOSCitASfE0IbOShnndA
Test Transaction	Yes
Transaction authorised	Yes
Transaction uncertain	No
Transaction paid	Yes
Card Type	Saferpay Test Card
Effective Payment Method Name	SaferpayTestCard
Authorisation Method	PaymentPage
3D Secure	Successful

Shipping & Handling Information

Flat Rate - Fixed CHF10.00

Dec 14, 2015 11:08:14 AM | Authorization
The amount of 107.00 is authorized. (Customer successfully returned from the Saferpay payment page.)

The order was placed using CHF.

Figure 6.1: Transaction Information.

6.5 Usage of the Alias Managers / Token Solution

6.5.1 Magento 1

With the Alias Manager, your customers can securely save their credit cards with PayUnity for later purchases. You can enable this by activating the option "Alias Manager" in the configuration of the [Payment Method](#). The customer can then choose from their saved credit cards without having to enter all the details again.

Visa

Card N°: 9451 1231 0000 0111 Date of Expiry: 12/2015 CVV: 123

Saved cards:

xxxx xxxx xxxx 0111

You may choose from one of the cards previously used on this site.

Card holder name *

Test Alias

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 Transaction

Attention

To use the Alias Manager tit may be necessary to activate certain options in your PayUnity account. Please contact PayUnity directly to do so.

6.5.1.1 Deleting Customer Aliases

You can easily delete a customer's alias by opening the customer in the backend of Magento. Switch to the tab PayUnity Alias and delete the saved alias directly here.

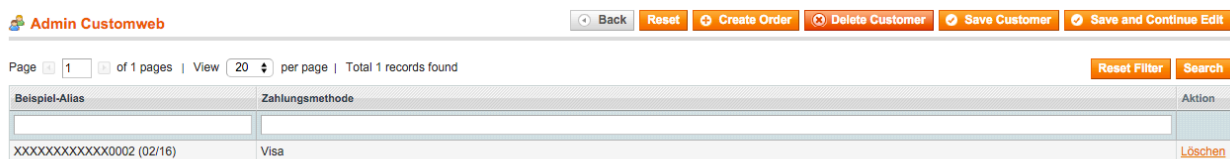


Figure 6.1: Delete previously saved aliases of your customers.

6.5.2 Magento 2

Due to the integration of the Alias Manager the customers credit card data can be saved for later purchases. The customer can choose between the cards that have already been saved and won't have to enter all of credit card data again.

6.5.2.1 Activating the Alias Manager / Token

In order for you to be able to use this you will need activate the option "Alias Manager" in the payment methods.

Payment

Check / Money order

Test Payment Method

VISA

Hans Muster
 Beispielstrasse 1
 Zürich, Zürich 8000
 Switzerland
 044 123 45 67
[Edit](#)



Stored Cards

xxxxxxxxxxxx0002

You may choose one of your stored cards.

Card holder name *

Hans Muster

Please enter the card holder's name.

Card number *

xxxxxxxxxxxx0002

Please enter your card number.

Expiry Date *

05 2018

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.

Place Order

Order Summary

Cart Subtotal CHF28.00

Order total CHF28.00

2 Items in Cart ^



LifeLong Fitness IV CHF28.00

Qty: 2



Figure 6.1: Alias Manager Transaction

6.5.2.2 Customers Deleting the Token

In Magento 2 you can allow your customers to manage the aliases in their customer account at "My PayUnity Aliases / Token". This feature can also be enabled in the settings of the payment method



Customer Alias Management	Yes	[STORE VIEW]
	You can allow the customers to manage/delete their aliases in their account.	
Await Notification	Yes	[WEBSITE]
	In some cases it may take a few seconds until the transaction is successfully authorized. When this option is enabled, the notification is awaited.	
External Checkout: Guest Checkout	Show Account Selection	[WEBSITE]
	When using an external checkout, the customer can either be asked to choose an option to authenticate (as guest, register or login) or he can always be checked out as guest. For the second option to work, guest checkout has to be enabled in Magento.	

Figure 6.1: Activating the Alias Manager for Customers.

6.6 Tax Calculation Settings

6.6.1 Magento 1

To ensure that the taxes are calculated correctly and transmitted correctly to a third party invoice provider inside Magento please make sure that you set the calculation settings in the menu under System > Configuration > Tax as showed in the screenshot below.

Calculation Settings		
Tax Calculation Method Based On	Total	[WEBSITE]
Tax Calculation Based On	Shipping Address	[WEBSITE]
Catalog Prices	Excluding Tax	[WEBSITE]
	▲ Whether catalog prices entered by admin include tax.	
Shipping Prices	Excluding Tax	[WEBSITE]
	▲ Whether shipping amounts entered by admin or obtained from gateways include tax.	
Apply Customer Tax	Before Discount	[WEBSITE]
Apply Discount On Prices	Including Tax	[WEBSITE]
	▲ Apply discount on price including tax is calculated based on store tax, if "Apply Tax after Discount" is selected.	
Apply Tax On	Custom price if available	[WEBSITE]
Enable Cross Border Trade	No	[WEBSITE]
	▲ When catalog price includes tax, enable this setting will fix the price no matter what the customer's tax rate is.	

Figure 6.1: Recommended tax calculation settings for Magento.

Please also make sure that inside the settings of your promotion in the tab "Actions" the discount is not applied to the shipping amount as Magento is not correctly transmitting the tax rate there.

6.7 OneStepCheckout

6.7.1 Magento 1

The checkout of Magento can be modified and reduced to one step. The so called OnePageCheckout. Our modules support the most important OnePageCheckout extensions. If you are thinking of using a OneStepCheckout extension in combination with our PayUnity Magento, please make sure that you are using one of the following:

- OneStepCheckout by www.onestepcheckout.com
- OneStepCheckout by [Ahead Works](#)
- LightCheckout by [GoMage](#)
- Firecheckout by [Templates Master](#)

6.7.2 Magento 2

There is no official support for any OnePageCheckout module for Magento 2.

6.8 Setup a Cron Job to Activate the Timed Operations

6.8.1 Magento 1

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

The module uses the standard cron engine of Magento. More information regarding the set up can be found [here](#).

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

6.8.2 Magento 2

The module uses the standard cron engine of Magento 2. More information regarding the set up can be found [here](#).

6.9 Magento 1 API Manual

Via API, our module provides plenty of information on a transaction. The following section describes which calls are possible and what information is provided.

6.9.1 General Information on Using the API

General information on using the Magento SOAP API can be found here: <http://www.magentocommerce.com/api/soap/introduction.html>

6.9.2 API Calls

This overview shows the API Versions and which API Calls they support.

6.9.2.1 API V1

- `$client->call($session, 'payunitycw_transaction.list');` //List of all transactions
- `$client->call($session, 'payunitycw_transaction.info', {transactionsId});` // Single Transaction

6.9.2.2 API V2

- `$client->payunitycwTransactionList($session);` //List of all transactions
- `$client->payunitycwTransactionInfo($session, {transactionsId});` //Single Transaction

6.9.2.3 XML-RPC

- `$client->call('call', array($session, 'payunitycw_transaction.list');` //List of all transactions
- `$client->call('call', array($session, 'payunitycw_transaction.info', {transactionsId}));` //Single Transaction

6.9.2.4 REST

- `$this->callApi('payunitycw/transactions');` //List of all transactions
- `$this->callApi('payunitycw/transactions/{transactionsId}');` //Single Transaction

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:

American Express

Card number	377777777777770	No 3D Secure
Expiry Date	12/2020	
CVC	123	
Card number	375987000000005	3D Secure
Expiry Date	12/2020	
CVC	123	

Carte Bleue

Card number	5555555555554444
Expiry Date	12/2020
CVC	123

Dankort

Card number	5019717010103742
Expiry Date	12/2020
CVC	123

Diners Club

Card number	36961903000009
Expiry Date	12/2020
CVC	123

Sepa Direct Debits

IBAN	AT152011128161647502	Austria (AT)
BIC	GIBAATWWXXX	
IBAN	DE23100000001234567890	Germany (DE)
BIC	MARKDEF1100	
IBAN	ES9121000418450200051332	Spain (ES)
BIC	CAIXESBBXXX	

Discover Card

Card number 6011587918359498
 Expiry Date 12/2020
 CVC 123

giropay

IBAN AT152011128161647502 Austria (AT)
 BIC GIBAATWWXXX

IBAN DE23100000001234567890 Germany (DE)
 BIC MARKDEF1100

IBAN ES9121000418450200051332 Spain (ES)
 BIC CAIXESBBXXX

JCB

Card number 3541599999092431
 Expiry Date 12/2020
 CVC 123

Klarna Invoice

Klarna (Approved)
 Klarna (Denied)
 Klarna (Pending -> Approved)
 Klarna (Pending -> Denied)

Maestro

Card number 67998510000000032
 Expiry Date 12/2021
 CVC 123

MasterCard

Card number 5454545454545454 No 3D Secure
 Expiry Date 12/2021
 CVC 123

Card number 5212345678901234 3D Secure
 Expiry Date 12/2021
 CVC 123

Visa

Card number 4200000000000000 No 3D Secure
 Expiry Date 12/2020
 CVC 123

Card number 4012001037461114 3D Secure
 Expiry Date 12/2020
 CVC 123

Card number 4000000000000010 3D Secure 2.0
 Expiry Date 12/2021

CVC 123

V PAY

Card number 48220000000000000003

Expiry Date 12/2020

CVC 123

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 Module is not shown

If the module is not shown even though the files have been downloaded onto the server correctly, please check the following:

- The local.xml can be found in the /app/etc directory
- Make sure `disable_local_modules` is set to `,false'`

8.2 The Referrer URL appears in my Analytics Tool

When a customer and the notification are redirected via Header Redirection, the PayUnity 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 a easier way.

9.1 Magento 1

9.1.1 Fees and discounts within Magento

To configure a PayUnity payment gateway based fee and discount, you will need the following 3rd-Party plugin.

- [Fooman Surcharge](#)

9.2 Birthday and gender in Magento 1

For certain payment service providers it is necessary to check the birthday an the gender of a customer. Magento 1 does not check this by default. this check can be enabled under "Go to System > Configuration | Customers > Customer Configuration > Name and Address Options".

9.3 Birthday and gender in Magento 2

For certain payment service providers it is necessary to check the birthday an the gender of a customer. Magento 2 does not check this by default. this check can be enabled under "Stores > Configuration | Customers > Customer Configuration > Name and Address Options".

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 PayUnity settings. The store log setting does not affect these log messages.

- 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 the Magento log directory. (Default Path: {shopRootDirectory}/var/log)

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.

11.1.1 Magento 1

Initialize the shops core functions.

```
define('MAGENTO_ROOT', getcwd());
$mageFilename = MAGENTO_ROOT . '/app/Mage.php';
require_once $mageFilename;
Mage::setIsDeveloperMode(true);
ini_set('display_errors', 1);
umask(0);
Mage::app();
```

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

Load the transaction object by Id:

```
$dbtransaction = Mage::getModel('payunitycw/transaction')->load($transactionId);
$transactionObject = $dbtransaction->getTransactionObject();
```

Load the transaction object by Order Number:

```
$dbtransaction = Mage::getModel('payunitycw/transaction')->load($orderId, 'transaction_external_id');
$transactionObject = $dbtransaction->getTransactionObject();
```

11.1.2 Magento 2

Initialize the shops core functions.

```
use Magento\Framework\App\Bootstrap;
require __DIR__ . '/app/bootstrap.php';
$bootstrap = Bootstrap::create(BP, $_SERVER);
$obj = $bootstrap->getObjectManager();
```

```
$state = $obj->get('Magento\Framework\App\State');  
$state->setAreaCode('frontend');
```

Get the transaction factory.

```
$factory = $obj->get  
( 'Customweb\PayUnityCw\Model\Authorization\TransactionFactory' )->create();
```

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

Load the transaction object by Id:

```
$dbTransaction = $factory->load($transactionId, 'entity_id');  
$transactionObject = $dbTransaction->getTransactionObject();
```

Load the transaction object by Order Number:

```
$dbTransaction = $factory->load($orderId, 'order_id');  
$transactionObject = $dbTransaction->getTransactionObject();
```

Load the transaction object by Payment Id:

```
$dbTransaction = $factory->load($paymentId, 'payment_id');  
$transactionObject = $dbTransaction->getTransactionObject();
```