Installation Manual for Stripe OXID

This manual describes the installation and usage of the Stripe extension for OXID.

Release Date: Mon, 07 Aug 2023 14:01:06 +0200
Version: 3.0.242

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 the installation, configuration and use of the payment module for OXID and Stripe.

Before you can begin with the installation, make sure that you have all the necessary data at hand. You will need:

- Stripe Public Key
- Stripe Secret Key
- OXID payment module from sellxed.com/shop
- Login data for you 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 the information necessary to install the module. When activating the module each step must be followed carefully. This ensures the safe use of the module in compliance with all security regulations.

1. Configuration of the Main Module
2. Configuration of the accepted payment methods as well as the authorization methods
3. Conducting a test order
4. Activating your Stripe account

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 Stripe might not get through to the shop.

1.2 System Requirements

In general, the plugin has the same system requirements as OXID. 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 Stripe Backend Configuration

You can find an overview of the transactions and settings in the backend of Stripe:

- Test Environment: https://dashboard.stripe.com/login
- Live Environment:

2.1 Configuration of the Main Module

This is a brief installation instruction of the main and most important installation steps. It is important that you strictly follow the checklist. Only by doing so, the secure usage in correspondence with all security regulations can be guaranteed.

- After having opened your account at Stripe, enter the access data that you received from Stripe into the main module in the space provided.
- Copy the Secret Key that you find in the Stripe Dashboard under Developers > API Keys.
- Copy the Publishable Key that you find in the Stripe Dashboard under Developers > API Keys.
- Create a new endpoint in the Stripe Dashboard under Developers > Webhooks, and enter the link provided in the setup form of the plugin.
- Open the previously created endpoint, and copy the Signing Key. The Signing Key needs to be entered in the main settings of the plugin in the field 'Webhook Secret'.
- Select 'Update Details' and activate the endpoint for the following events: charge.failed, charge.succeeded, source.canceled, source.chargeable, source.failed
- Optional: If you use SEPA, do not forget to configure your merchant name.
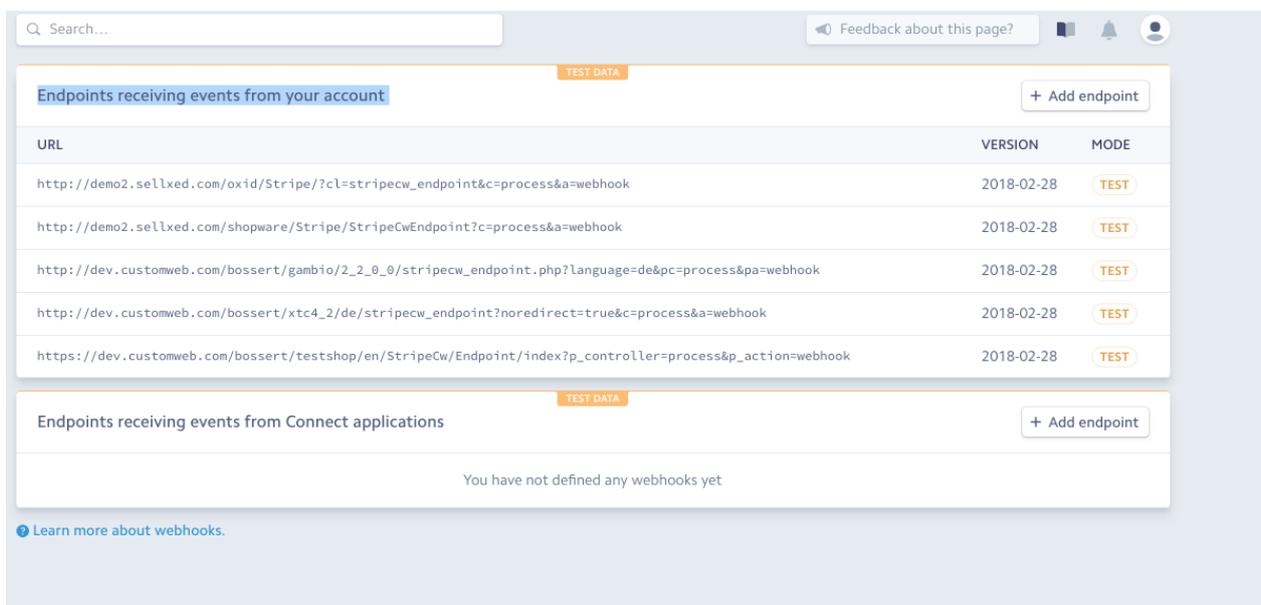
Figure 2.1: Create Webhook Endpoint
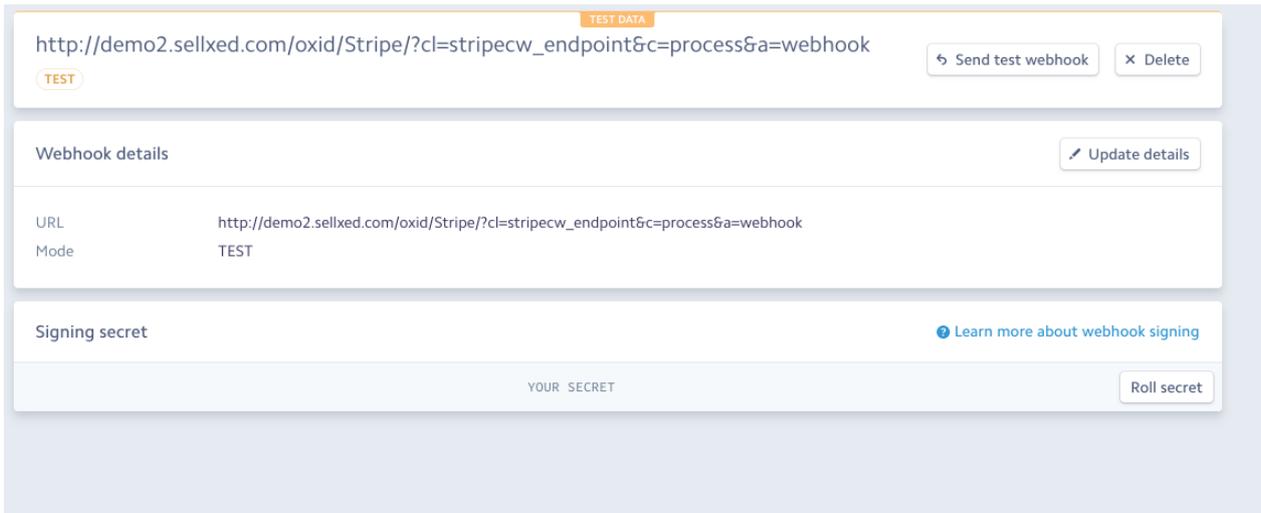
Figure 2.1: Copy Webhook Secret
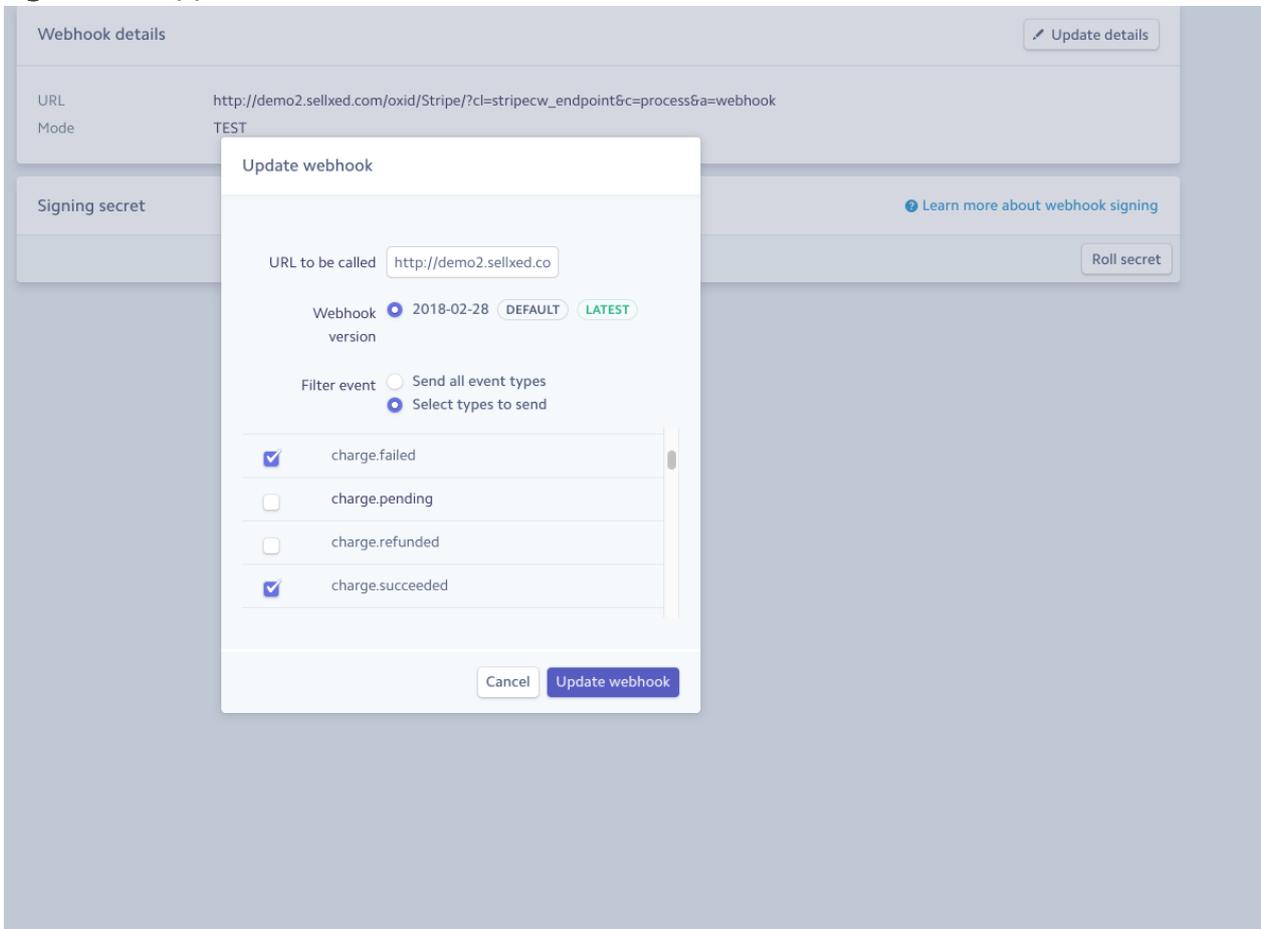
Figure 2.1: Select Events

2.2   Configuration of the Payment Methods

For further information about the configuration of the payment methods and the meaning of each setting, click here. After having activated and configured the payment methods you can run a test by means of the published test data in the manual. If all the tests were successful, switch the operating mode in the main module to **Live**.

2.3   Apple Pay, Google Pay, Microsoft Pay.

These payment methods are only supported on mobile devices and with the required versions (OS and Browser)

When a customer is paying via Apple Pay, Googlr Pay, Microsoft Pay a button will be displayed on the screen. ONce this button is pushed the plugin will automatically detect the customers device and open the corresponding app. To use apple pay you need to be registered, all information on this can be found here: https://stripe.com/docs/stripe-js/elements/payment-request-button#verifying-your-domain-with-apple-pay

2.4   Additional Styling via Stripe.css

It is possible to style the fields of your form with CSS. For this, you may simply edit the provided stripe.css file inside the module. The input style is defined in .StripeElement, the container is defined in .StripeWrapper. For more information you may visit this link.

3 Module Installation and Update in the OXID 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 OXID 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

The settings for the main module can be found under **Extensions > Module** . Install the module by clicking **Activate**.

By clicking on the tab **Settings** you get to the configuration of the main module. Fill in the boxes by using the data that you have received from Stripe and may have already saved in the backend of Stripe. Each option is additionally explained in a short info text in the shop.

4.1.1 Settings

4.1.1.1 Update interval for payments

Payments may be updated periodically (interval in minutes). To use this feature you have to setup a cron job.

4.1.1.2 Order Creation

Choose when the order should be created. If set to 'Before Payment', orders are created in any case. If set to 'After Payment', an order is created only if the payment is successful.

4.1.1.3 Order ID

Use this setting to enforce the behaviour of unique order IDs, even if they are not.

Changing Order Creation and Order ID Settings

Changing these settings may cause duplicate transaction ids to be distributed, which will cause errors during transaction processing. If you wish to change these after transactions have been created you must backup then empty the module transaction table in the database.

4.1.1.4 Delete Orders on Failure

Set to yes to delete the order when the authorization fails. Otherwise the internal order status is set to 'PAYMENT_FAILED'.

4.1.1.5 Logging Level

Messages of this or a higher level will be logged.

4.1.2 OXID 6 Installation

After uploading the plugin via S/FTP, log into your server via SSH and navigate to the root folder of OXID. In this directory, run the following commands so the plugin is available in the backend:

command vendor/console/bin oe:module:install-configuration source/modules/customweb /stripecw/

php vendor/bin/oe-console oe:module:activate stripecw

php vendor/bin/oe-console oe:module:apply-configuration

4.2 Defining the URL for the Transaction Feedback

To ensure correct payment processing in your shop, at least one Feedback URL has to be provided in the Stripe configuration. The URL to be entered can be found in your OXID Shop under: **Extensions > Stripe > Setup > Setup**

4.3   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 **Shop Settings > Payment Methods** . Each method is listed individually.

Install the payment methods you wish to offer your customers. Open the payment method and tick the box next to **Active**.

By clicking on Stripe Configuration, you can save different settings for each payment method and thereby optimally adapt the payment methods to your existing processes

Payment Methods not Displayed in the Checkout

Please check if you have attributed the payment method to the available countries as well as customer groups in case a should payment method not be visible in the checkout.

4.4   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 Stripe and the module

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

4.5   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.

4.5.1 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.

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

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 Stripe module.

6.1 Useful Transaction Information on the Order

In the tab **Stripe Transactions** you can find an overview of the transaction information. Among others, this information allows for the definite attribution of the orders to their corresponding transaction, seen in the backend of Stripe.

Autorisierungsbetrag	947.19
Währung	EUR
Zahlungsmethode	Visa
Zahlungsid	25839786
Verbuchter Betrag	947.19
Transaktion autorisiert	Ja
Transaktion ist unsicher	Nein
Transaktion verbucht	Ja
Transaktion bezahlt	Ja
Akzeptanz	test123
Kartennummer	XXXXXXXXXXXX0003
Kartenablaufdatum	12/15
Händler Referenz	ox_sel_pf_2

Figure 6.1: Transaction Information in OXID

6.2 Usage of the Alias Managers / Token Solution

With the Alias Manager, your customers can securely save their credit cards with Stripe 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 his or her saved credit cards without having to re-enter all the details.

Figure 6.1: Alias Manager Integration in OXID eSales

> ### Alias Manager Options
>
> The usage of the Alias Managers requires the activation of the correct option with Stripe. To do so, please contact the support directly.

6.3 Capturing / Cancelling of Orders

Our module allow you to capture payments directly from within Oxid eSales without having to log into Stripe.

6.3.1 Capturing Orders

In order to capture orders, open the transaction tab (Stripe Transactions). By clicking on the button **Capture** a new window will open up in which you can capture transactions.

You can now capture each item individually if you don't want to ship them all at once. The amount of possible partial captures depends on your contract. For more information on this, please contact Stripe directly.

Capturing of Orders in the backend of Stripe

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

I you do not want to capture all items of an orders, click on 'Close transaction for further captures'.

Teilbuchung

Mit dem folgenden Formular können Sie eine Teilbuchung durchführen.

Name	SKU	Typ	Steuersatz	Menge	Gesamttotal (exkl. MwSt.)	Gesamttotal (inkl. MwSt.)
Trapez ION SOL KITE 2011	1401	product	0 %	2	216.80	216.80
Transportcontainer THE BARREL	3788	product	0 %	1	20.97	20.97
Shipping	shipping	shipping	0 %	1	6.90	6.90

Gesamtbuchungsbetrag: 244.67 EUR

Transaktion für weitere Buchungen schliessen ☐

Buchung

Figure 6.1: Capturing of Orders in OXID eSales.

Partial Capturing

Please find out if the capturing of partial amounts is supported by your Stripe contract. If this is not the case, it might happen that no further captures can be carried out.

6.3.2 Cancelling Orders

By clicking **Cancel** the transaction is **cancelled** and the amount reserved on the customer's card will be released immediately.

6.4 Refunding Orders

In order to refund orders, open the transaction tab (Stripe Transactions) and click on **Refund**. A new window will open up.

You can refund every item individually or any amount of your choice by modifying the total amount to pay or the amount of items.

Figure 6.1: Refunds in OXID eSales for Stripe.

Maximal Refund

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

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

6.5   Setup a Cron Job to Activate the Timed Operations

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

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

http://urltoyourstore.com/index.php?cl=stripecw_cron

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

7 Testing

Before switching from test to live mode it is important that you test the module extensively.

> **Testing**
>
> Do not forget to switch the operating mode from test to live after having successfully tested the module.

7.1 Test Data

In the following section you can find the test data for the various payment methods:

Credit / Debit Card

Card number	4242424242424242	VISA
Expiry Date	12/2020	
CVC	123	

Card number	4000056655665556	VISA Debit
Expiry Date	12/2020	
CVC	123	

Card number	5555555555554444	MasterCard
Expiry Date	12/2020	
CVC	123	

Card number	5200828282828210	MasterCard Debit
Expiry Date	12/2020	
CVC	123	

Card number	378282246310005	American Express 1
Expiry Date	12/2020	
CVC	123	

Card number	371449635398431	American Express 2
Expiry Date	12/2020	
CVC	123	

Card number	6011111111111117	Discover 1
Expiry Date	12/2020	
CVC	123	

Card number	6011000990139424	Discover 2
Expiry Date	12/2020	
CVC	123	

Card number	3530111333300000	JCB 1
Expiry Date	12/2020	
CVC	123	

		JCB 2
Card number	3566002020360505	
Expiry Date	12/2020	
CVC	123	

		Diners 1
Card number	30569309025904	
Expiry Date	12/2020	
CVC	123	

		Diners 2
Card number	38520000023237	
Expiry Date	12/2020	
CVC	123	

Visa

		VISA
Card number	4242424242424242	
Expiry Date	12/2020	
CVC	123	

		VISA Debit
Card number	4000056655665556	
Expiry Date	12/2020	
CVC	123	

MasterCard

		MasterCard
Card number	5555555555554444	
Expiry Date	12/2020	
CVC	123	

		MasterCard Debit
Card number	5200828282828210	
Expiry Date	12/2020	
CVC	123	

American Express

		American Express 1
Card number	378282246310005	
Expiry Date	12/2020	
CVC	123	

		American Express 2
Card number	371449635398431	
Expiry Date	12/2020	
CVC	123	

Discover Card

		Discover 1
Card number	6011111111111117	
Expiry Date	12/2020	
CVC	123	

		Discover 2
Card number	6011000990139424	
Expiry Date	12/2020	
CVC	123	

JCB

		JCB 1
Card number	3530111333300000	
Expiry Date	12/2020	
CVC	123	

Card number	3566002020360505	JCB 2
Expiry Date	12/2020	
CVC	123	

Diners Club

Card number	30569309025904	Diners 1
Expiry Date	12/2020	
CVC	123	
Card number	38520000023237	Diners 2
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 The Referrer URL appears in my Analytics Tool

When a customer and the notification are redirected via Header Redirection, the Stripe 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 Birthday and gender in OXID

For certain payment service providers it is necessary to check the birthday and the gender of a customer. OXID does not check this by default. These checks can be enabled under "Settings > Other Settings > Mandatory fields in User Registration Form".

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 Stripe 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 the OXID log directory. Please make sure the folder and it is writable by the webserver. (Default Path: {shopRootDirectory}/logs/)

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

```
require_once dirname(__FILE__) . "/bootstrap.php";
```

Include the module module classes.

```
require_once 'modules/customweb/stripecw/classes/StripeCwHelper.php';
```

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

```
$transactionById = StripeCwHelper::loadTransaction(transactionId);
$transactionObject = $transactionById->getTransactionObject();
```