

ABOUT

Stripe Payments is a module for Magento 2 which allows merchants to accept payments using the **Stripe** payment gateway. This is the documentation for installing, configuring and using this module on **Magento 2**.

For the most up to date documentation, you can always visit the online version of this documentation by clicking [here](#).

INSTALLATION

Warning: We recommend that you test the module on a testing server before installing it on your live web server. Please see the **Troubleshooting** section if you come across any installation issue.

1. As a minimum requirement, you will need **cron** running and fully functional for the installation. If you have not set up cron already, please follow the [official instructions](#) from Magento before proceeding.
2. Ensure that your web server has full write access to the entire Magento directory. Official instructions can be found [here](#). However the simplest way is to change the ownership of the Magento directory to the same user under which the web server is running. This can be done with:

```
$ chown -R <www-username> /magento_directory
```

3. Navigate to the Setup Wizard at <http://yourdomain.com/admin/admin/backendapp/redirect/app/setup/>
4. To access your purchase from the marketplace, you will need to enter your Magento Marketplace authentication keys. If you haven't done so already, please follow [these instructions](#).
5. From the Setup Wizard page, click the **Component Manager** icon
6. In the top section of the page, you will see a **New Purchases** section with an **Install** link underneath. Click the Install link and follow the instructions to install the module. The available versions are in the format 10x.x.x where x.x.x represents the actual version number of the module.
7. The installation can take up to 5-6 minutes for cron to run its full cycle, upgrade Magento, re-deploy the static assets, recompile the code etc. Please be patient and leave the installation to finish without closing the window.
8. If for any reason the installation takes unusually long, or it gets stuck at "Update pending", check for errors in var/log/update.log. Alternatively, you can run the cron commands manually from a shell without the pipes at the end so that errors are displayed in the shell. If the command succeeds, the contents of the files under var/.update_* will also be updated. If any errors occur, simply contact us

at info@cryozonic.com with details on what happened so that we can help you complete the installation.

9. Proceed to [Stripe Webhooks Configuration](#) in your Stripe dashboard.

UNINSTALL

1. Navigate to the Setup Wizard at <http://yourdomain.com/admin/admin/backendapp/redirect/app/setup/>
2. Click the **Component Manager** icon
3. On the right hand side column, you will see a dropdown with an **Uninstall** link. Click the Uninstall link and follow the instructions to uninstall the module.

135 records found

20

per page

<

1

of 7

>

	Component Name ↓	Module Name	Version	Type	Vendor	Actions
●	cryozonic/module-stripe-payments	Cryozonic_StripePayments	101.1.0	module	cryozonic	Select ▲ Disable Uninstall
●	magento/framework		100.1.2	library	magento	
●	magento/language-de_de		100.1.0	language	magento	
●	magento/language-en_us		100.1.0	language	magento	

Click Uninstall →

UPGRADE

You can always download the latest version of the module from the [account section](#) or through the email that was sent to you when you purchased the module. Once you have the latest version, and

- the module was purchased from our website, simply uninstall the module using [these instructions](#) and re-install using [these instructions](#).
- the module was purchased from the Magento Marketplace, uninstall the module using [these instructions](#) and re-install using [these instructions](#).

CONFIGURATION

For full configuration instructions, please see our online documentation at <http://store.cryozonic.com/documentation/magento-2-stripe-payments>

TROUBLESHOOTING

Switching to Developer Mode

First and foremost, before debugging any kind of Magento 2 issue, you must make sure that you have switched to the **developer** Magento mode. It is very hard to debug Magento 2 errors when running in production mode. Please review [these steps](#) on how to switch to developer mode. In addition to the official

documentation, some people may set the Magento mode using the `$MAGE_MODE` environment variable from the web server's configuration files, i.e. inside Nginx's site configuration files or in Apache's `.htaccess` files. Make sure that if you have this configuration, it does not interfere with the mode that was set from the command line.

Error Logging and 500 Server-Side Errors

Magento will log any Errors and Exceptions thrown during application runtime in the `var/log` directory. The module will also throw Exceptions in a similar manner which will be also be logged there. You can find these errors in the following two files:

```
var/log/system.log
var/log/exception.log
```

If you have SSH access, you can quickly filter the error messages using the following command:

```
grep -i Stripe var/log/system.log
```

To monitor errors live as they happen, clear the console (Ctrl-L on Linux, Cmd-K on OSX) and run the following command:

```
tail -f var/log/system.log
```

This will show errors live. You can refresh the problematic page in your browser and if there are any errors, you will see them appear in the console.

If you do not have shell access, just download the file, open it with your favourite text editor and search for "Stripe".

Upgrades / Caching Issues

If you have upgraded the module but for some reason you don't see the new changes, you can manually clear the Magento 2 cache by deleting the contents of the following directories:

```
var/cache/*
var/page_cache/*
var/di/*
var/generation/*
var/view_preprocessed/*
var/composer_home/cache/*
pub/static/*
```

Once the above have been cleared, run the following commands:

```
php bin/magento setup:upgrade
php bin/magento cache:flush
php bin/magento setup:static-content:deploy
```

If you are running in production mode, you may also have to recompile:

```
php bin/magento setup:di:compile
```

If you are running Varnish, you must also restart Varnish after deleting the `var/cache/*` files.

Finally, sometimes browsers also cache Magento 2 requests, if you still have caching issues, try with a different browser.

Payment method not showing in admin area

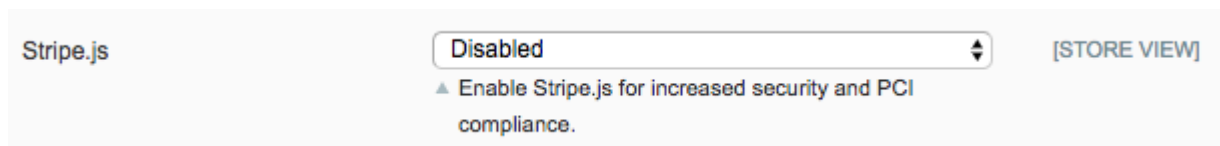
If you have recently upgraded to Magento 2.1.3, you may have noticed that the module's configuration section no longer displays in the Magento admin area. This is because of a backwards incompatible change in Magento that affects all 3rd party payment modules and is documented [here](#).

If you are affected by this issue, the solution is to edit line 12 of `app/code/Cryozonic/StripePayments/etc/adminhtml/system.xml` so that the **sortOrder** attribute is increased from 1 to 10 like so:

```
<group id="cryozonic_stripe" translate="label" type="text" sortOrder="10"
showInDefault="1" showInWebsite="1" showInStore="1">
```

Unable to use Stripe.js

This error indicates that Stripe.js is not working correctly on the page that displayed this error. You can test if this is the case by disabling Stripe.js from the module's configuration section:



Make sure to flush your configuration cache after disabling Stripe.js and before performing another test. Disabling Stripe.js will make your website insecure and non-PCI compliant, so we recommend to re-enable it as soon as the error is resolved.

The most common reason for this error is; if you are still developing the website, you are likely to have unstable modules with javascript crashes in the browser's debugging console. When a javascript crash occurs, all subsequent javascript execution is halted by the browser, resulting in Stripe.js not being initialized correctly at your checkout page. If you can see any javascript crashes in your browser console, fixing them should be the first thing to try in order for the checkout page to initialize correctly.

If your browser's javascript console is clean from errors, but you are still receiving this error, please get in touch with us at info@cryozonic.com

Compatibility with One Step Checkout modules

After requests from customers, we have tested several OSC modules with Stripe Payments. Following are the testing results for the benefit of other customers:

COMPATIBLE

- IWD OSC 1.0.2 - Tested against v1.0.1 of our module on Magento 2.1.1. No changes were necessary for compatibility with our module. We did find however an incompatibility between IWD OSC and newer versions of Magento 2.1.x, which can be fixed with [this patch](#).
- Magestore OSC 2.0.0 - Tested against v1.1.0 on Magento 2.1.2.
- Mageplaza OSC 2.1.1 - Tested against v1.1.3 on Magento 2.1.3.
- Mageplaza OSC 2.3.0 - Tested against v1.4.1 on Magento 2.1.4.

INTEGRATION STEP NECESSARY

- aheadWorks OSC - The OSC module breaks the Stripe.js bindings of the "Place Order" button, which are necessary for security by tokenization. However several customers have reported having this issue resolved after contacting the aheadWorks support team, which helped to integrate the two modules.

INCOMPATIBLE

- Mageplaza OSC 1.1.2 - Tested against v1.1.0 on Magento 2.1.2. The payment method cannot be not loaded at the checkout page. We cannot verify the cause of this at the moment.
- BSS Commerce OSC 1.0.8 - Tested against v1.5.3 on Magento 2.1.10. The Stripe API can be used to place regular orders, however when Stripe.js is enabled, the Knockout.js element bindings do not work correctly when the Place Order button is clicked.

Other OSC modules tested by our clients work correctly with no modifications. However, if you are using a OSC module that is causing problems and is not listed in the above list, please contact us at info@cryozonic.com.

An error occurred on the server. Please try to place the order again.

This error is related to a long standing open issue with Magento 2, which can be found [here](#). There are many possible reasons for this error, and each merchant may have a completely different reason for why this happened. We have observed that our customers typically receive this because of crashes in 3rd party modules that try to do something during the checkout process, for example, marketplace modules trying to adjust the sales emails, OneStepCheckout modules crashing because of a template issue, modules trying to wrap Plugin methods around the payment module functions and so on. The reason that this appears as a generic error is because Magento 2 does not handle exceptions thrown by these modules very well. This will be fixed in newer versions of Magento 2, however if you are getting this, you will need to do a bit of manual digging to find the error.

To identify the cause of the problem, you will first need to make sure that you have Magento running in developer mode. Then, try to place an order. Before clicking the Place Order button, open your browser's web inspector (the below example is using Chrome) and navigate to the Network tab so that you can see the requests. You will see a server side error in red as shown below:

Name	Method	Status	Type	Initi...	Size	Time	Waterfall
estimate-shipping-methods	POST	200	xhr	ique...	581B	1.12s	
shipping-information	POST	200	xhr	ique...	952B	1.45s	
loader-2.gif	GET	200	gif	Other	2.4KB	740ms	
?sections=messages&update_section_id=true&_=1492196572917	GET	200	xhr	ique...	732B	530ms	
1_6.jpg	GET	200	jpeg	Other	1.3KB	366ms	
tokens	POST	200	xhr	(ind...	1.3KB	924ms	
payment-information	POST	400	xhr	ique...	3.7KB	2.27s	
?sections=cart%2Cmessages&update_section_id=true&_=149219...	GET	200	xhr	ique...	1.2KB	708ms	

Open the network request and click on the Response tab. There, you will find the full error that caused the issue:

Name	x	Headers	Preview	Response	Cookies	Timing
estimate-shipping-methods						
estimate-shipping-methods						
shipping-information						
loader-2.gif						
?sections=messages&update_section_id=true&_=1492196935178						
tokens						
payment-information				1 Object (Magento\Framework\App\Request\Http)#6 \mnt\data/home/41971-83267.cloudway:		
?sections=cart%2Cmessages&update_section_id=true&_=1492196935179						

If there is no error backtrace here, the only other place to look is under `var/log/exception.log`. You should be able to get a full backtrace of the error in one of these two locations.

Once you have the full error backtrace, you can usually inspect it carefully to find which module it is coming from. You can disable any suspected module and try again to see if the error is fixed.

If however the error comes from the core Magento code, please contact our customer support so that we can offer advice.

Stuck loading spinner at checkout page

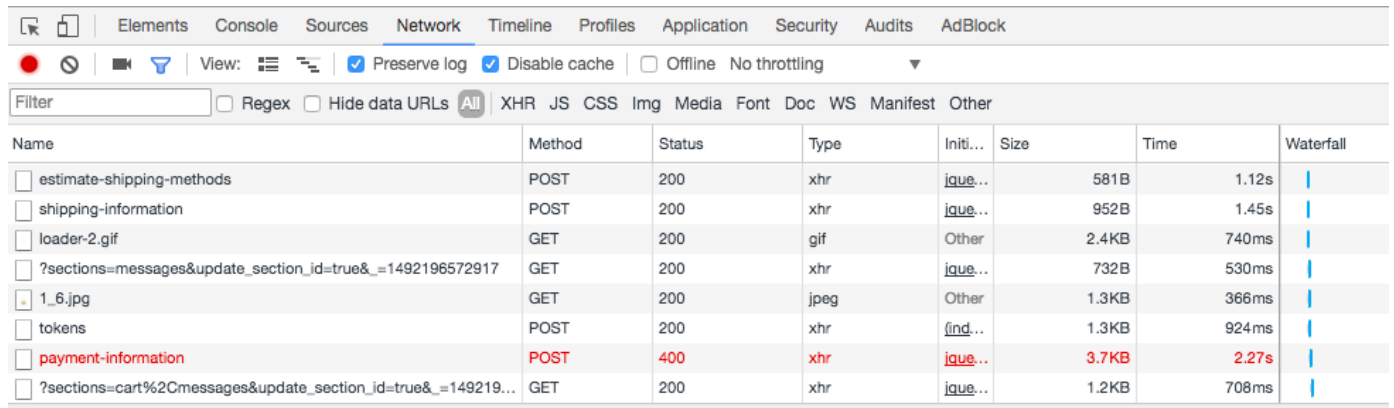
This may happen because of one of two reasons.

Possible reason 1: A server side crash

Server side crashes are logged in the Magento log files under `var/log` (see the [above section](#) on how to enable error logging). If you enabled error logging and still cannot see the error, please bear in mind that it may sometimes be logged in your web server's log files. If you are using a php-fpm variant, make sure to also check the log files of your fastcgi backend.

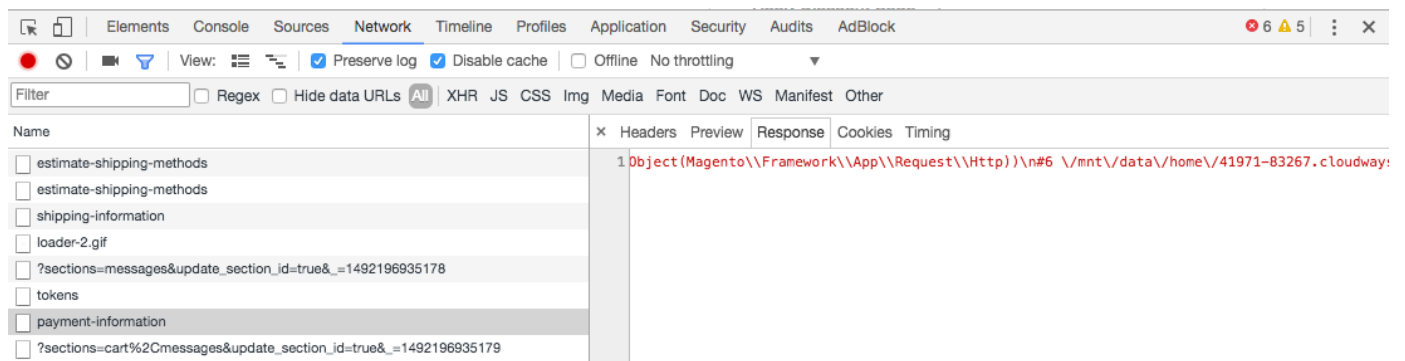
If the error is not logged in any of the above places, try using your browser's debugging console – for example in Chrome you can right-click and select Inspect.

If it is a 500 error, then under the Network tab of the console you will see a red network request as shown in the screenshot below:



Name	Method	Status	Type	Initi...	Size	Time	Waterfall
<input type="checkbox"/> estimate-shipping-methods	POST	200	xhr	ique...	581B	1.12s	
<input type="checkbox"/> shipping-information	POST	200	xhr	ique...	952B	1.45s	
<input type="checkbox"/> loader-2.gif	GET	200	gif		2.4KB	740ms	
<input type="checkbox"/> ?sections=messages&update_section_id=true&_=1492196572917	GET	200	xhr	ique...	732B	530ms	
<input type="checkbox"/> 1_6.jpg	GET	200	jpeg	Other	1.3KB	366ms	
<input type="checkbox"/> tokens	POST	200	xhr	(ind...	1.3KB	924ms	
<input type="checkbox"/> payment-information	POST	400	xhr	ique...	3.7KB	2.27s	
<input type="checkbox"/> ?sections=cart%2Cmessages&update_section_id=true&_=149219...	GET	200	xhr	ique...	1.2KB	708ms	

To find out what is causing the error, click on the line. You will see the error under the Response tab of the network request as shown below:

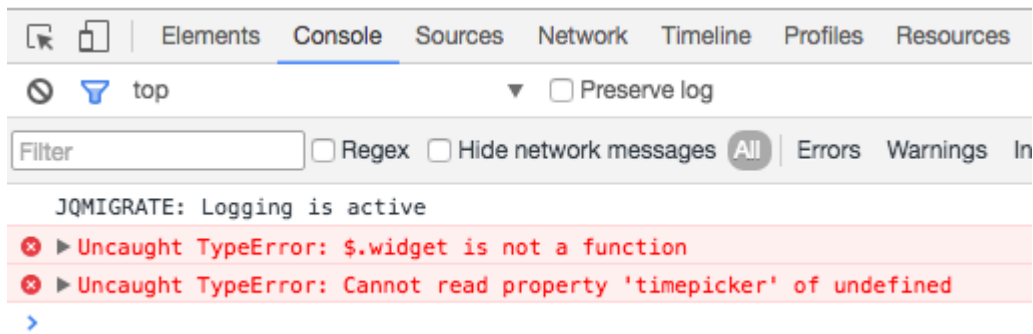


Name	Headers	Preview	Response	Cookies	Timing
<input type="checkbox"/> estimate-shipping-methods					
<input type="checkbox"/> estimate-shipping-methods					
<input type="checkbox"/> shipping-information					
<input type="checkbox"/> loader-2.gif					
<input type="checkbox"/> ?sections=messages&update_section_id=true&_=1492196935178					
<input type="checkbox"/> tokens					
<input type="checkbox"/> payment-information			<pre>1 Object(Magento\Framework\App\Request\Http)\n#6 /mnt/data/home/41971-83267.cloudway:</pre>		
<input type="checkbox"/> ?sections=cart%2Cmessages&update_section_id=true&_=1492196935179					

The error will give some clues as to where the crash is coming from. Usually it is some incompatibility with another module that tries to interfere with the checkout process, like triggering of custom emails, reward point modules etc. You can try to disable any such suspected modules by moving their xml file from `app/etc/modules/` into a different directory. Do not try to disable conflicting modules from the admin area – it doesn't always work.

Possible reason 2: A front-end javascript crash

Magento 2 and our Stripe module both use javascript to render the checkout view. If a javascript error occurs when you navigate to the checkout page, all subsequent javascript code execution stops, so the page may get stuck with a loading spinner or it may render the checkout page incorrectly. To check if a javascript crash may be causing this, you can use your browser's debugging console to check for errors, for example:



Errors such as the above, even if they are not caused by the module itself, they may prevent the customer from navigating or using the checkout page successfully. To make sure everything works as expected, make sure that the debugging console is always free of javascript errors.

Composer Authentication Required (repo.magento.com)

If you are installing composer for the first time, you may get this message in your console before installing the Stripe PHP library. The keys that you need to use are your Magento 2 marketplace keys. Please follow the instructions on [this page](#) to get your authentication keys.

PHP7 Invalid Stripe Secret Key

Magento versions 2.0.0 - 2.0.6 have a known issue with PHP7 where obscured/password fields are not decrypted correctly. You can either upgrade to Magento 2.0.7 or if not possible, please see [this commit](#) on how to apply a patch in your existing Magento installation.

Cannot create invoice for authorized order

If you have upgraded from Magento 2.0 to Magento 2.1 instead of installing 2.1 from scratch, you may face an issue where the invoice cannot be created for an authorized order - see [core issue 5546](#) for more details. We will be waiting for an official Magento fix for this issue. For the time being, our recommendation is to install Magento 2.1 from scratch. If you are already running in production and need to migrate your data, you can use the solution described [here](#), however this may cause system upgrade issues and must be taken into consideration when upgrading to Magento 2.1.1.

Failed/Partial/Corrupted Installations

Some crashes at the checkout page or the admin area may be caused by a partial/corrupted installation. Corrupted installations may happen because of incorrect filesystem write permissions. To fix a corrupted installation:

1. Carefully follow the [Uninstall Instructions](#)
2. Clean the database with:

```
DROP TABLE cryozonic_stripe_customers;  
DELETE FROM setup_module WHERE module = 'Cryozonic_StripePayments';
```


3. Check your filesystem permissions before the re-install. Most of these problems can simply be fixed by changing the ownership of the Magento directories to the user running the webserver with:

```
$ chown -R <www-username> /magento_directory
```

Alternatively you can configure your webserver to run as the same user that deploys these files to your website.

4. Carefully follow the [Installation Instructions](#)

CUSTOMER SUPPORT

If you have any other issues installing, configuring or using the module, please contact us at info@cryozonic.com