

Magento 2 - User Guide

Installation

Manual File Installation

Download and extract the module files on a local environment.

Once extracted add the **Stripe** folder to Magento under ***app/code*** - this should result in ***app/code/Stripe/{module code base}***

Tell Magento to setup and enable the module with the following commands:

- php bin/magento setup:upgrade
- composer install
- php bin/magento cache:flush

For production mode

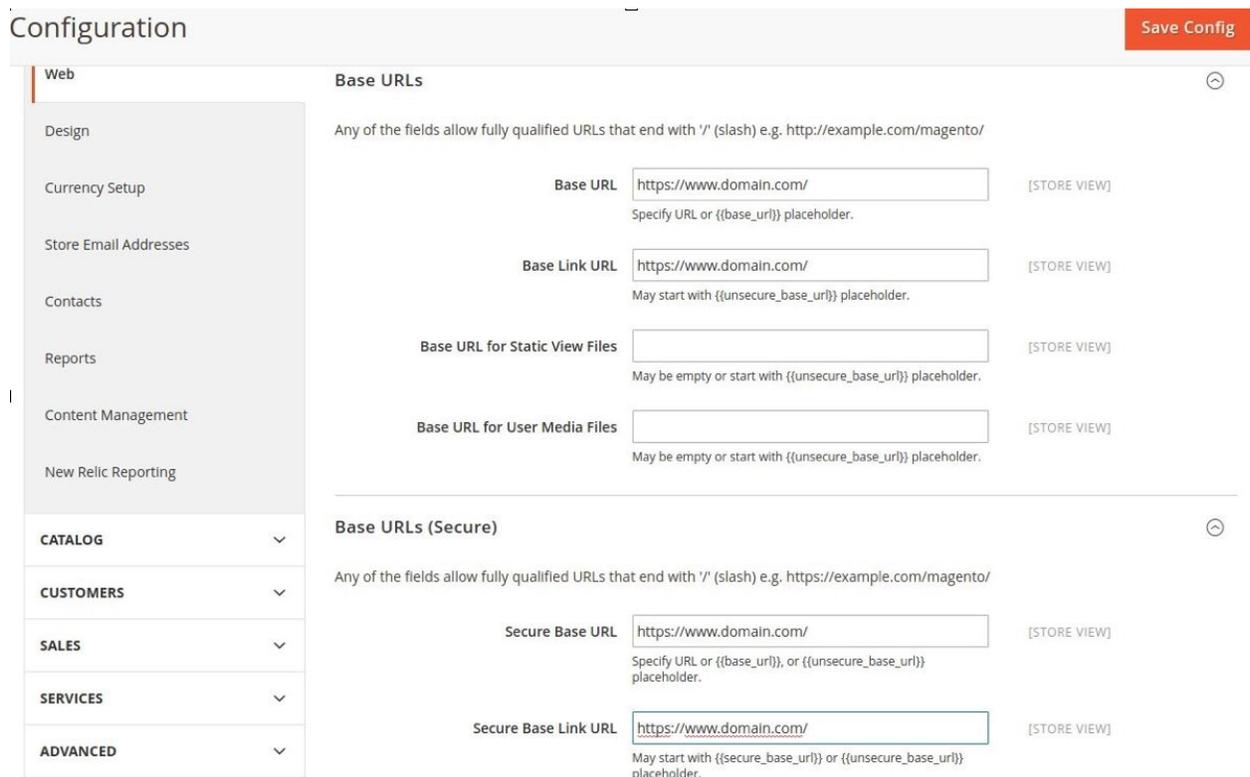
- php bin/magento setup:static-content:deploy
- php bin/magento setup:di:compile

Magento Configuration

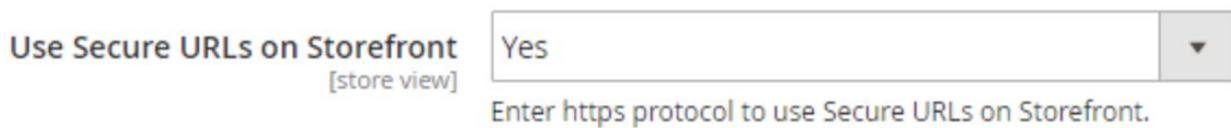
Checkout pages must be served over HTTPS for secure collection of card details with Stripe. If your checkout displays with http rather than https in your browser you should consult your hosting/website provider to fix this.

The below image shows where you define the secure base url. This must begin with https and not just http.

If you want Instant Checkout (Apple Pay / Google Pay) to work on non checkout pages you should use secure base url on all pages. One way of doing this is to set the base url as the same as the secure base url.



Alternatively from within the same section you can set it to use only the secure base url





Stripe Module Configuration

To activate the Stripe module and set the configuration navigate to **Stores > Configuration > Sales > Payment methods** expand **Stripe Payment Gateways** settings and expand **Stripe Core** settings within that.

Stripe Payment Gateways

The Stripe Payment Management Platform enables merchants to accept and manage payments, mitigate fraud, and secure their payment data.

[To Find Out More...](#)

Payment Methods Must be activated from your Stripe Dashboard Settings

[Here](#)



Notice: Instant Checkout and some other payment methods are design to work under Secure Sockets Layer enabled connection. It is strongly suggested to force HTTPS on your website

Stripe Core

Production Publishable API Key <small>[website]</small>	<input type="text"/>	This is your Publishable key for production
Production Secret API Key <small>[website]</small>	<input type="text"/>	
Sandbox Mode <small>[website]</small>	<input type="text" value="Yes"/>	
Sandbox Publishable API Key <small>[website]</small>	<input type="text" value="....."/>	This is your Publishable key for test
Sandbox Secret API Key <small>[website]</small>	<input type="text" value="....."/>	
Debug <small>[website]</small>	<input type="text" value="Yes"/>	



Sandbox Mode

To run the module in Sandbox Mode select **Sandbox Mode: Yes** in the Stripe Core settings and provide the test API keys from the Stripe Dashboard:

<https://dashboard.stripe.com/account/apikeys> (Note: for Sandbox Mode make sure that the keys include `_test_`, e.g. `pk_test_6pRNASCoBOKtIshFeQd4XMUh`)

In Sandbox Mode you can use the test payment details that Stripe provides:

<https://stripe.com/docs/testing>

Production Mode

Note that HTTPS usage needs to be enforced for secure card collection to work. See the [Installation](#) and [Magento Configuration](#) details above.

Stripe provides a different set of keys for production mode which include `_live_`. To switch into production, set Sandbox Mode: No, and provide the live keys in the Stripe Core settings.

You find the production keys in the Stripe Dashboard:

<https://dashboard.stripe.com/account/apikeys>. If they don't include `_live_`, make sure to untoggle "View test data."

Webhooks

Stripe provides Webhook notifications to inform you about changes in the system, e.g. asynchronous payment methods failing/succeeding. The module is designed to process webhooks at the following URL structure: <https://www.domain.com/stripe/webhook>.

Follow these steps to set up your Magento instance for Stripe webhooks:

- Navigate to <https://dashboard.stripe.com/account/webhooks> and click "Add Endpoint"
- Insert your shop URL with `/stripe/webhook` added at the end (see example above)
- **Webhook Version:** select the option that is labeled as LATEST.

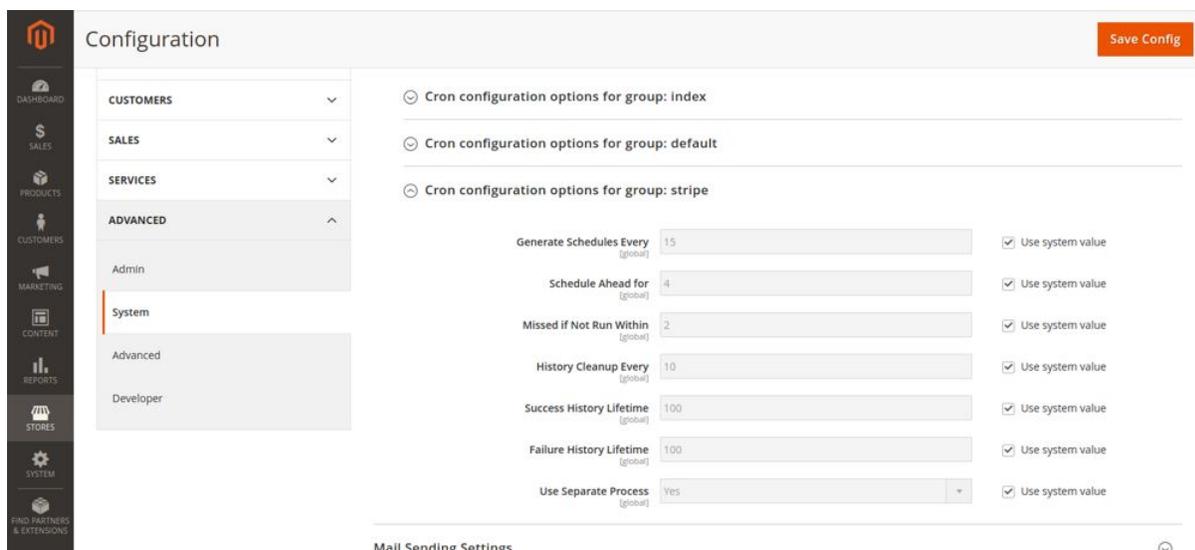
- **Filter event:** The module processes the following events that you want to select in the “Select types to send” option:
 - *charge.captured*
 - *charge.failed*
 - *charge.refunded*
 - *charge.succeeded*
 - *source.canceled*
 - *source.chargeable*
 - *source.failed*

Note: You have to set up webhooks separately for Sandbox and Production Mode. In the Stripe Dashboard you switch between the modes by using the “Viewing test data” toggle.

Cron Configuration

The Stripe webhook notifications are entered into a queue and processed via cron job. Having webhooks and cron jobs setup correctly is crucial for processing [asynchronous payment methods](#).

Navigate to **Stores > Configuration > Advanced > System** and check that the *Cron configuration options for group: stripe* is enabled correctly.



The screenshot shows the Magento Configuration interface. On the left is a vertical sidebar with navigation icons for Dashboard, Sales, Products, Customers, Marketing, Content, Reports, Stores, System, and Find Partners & Extensions. The main content area is titled "Configuration" and has a "Save Config" button in the top right. A left-hand menu lists categories: CUSTOMERS, SALES, SERVICES, and ADVANCED. Under ADVANCED, there are sub-sections for Admin, System (which is selected), Advanced, and Developer. The System section is expanded to show "Cron configuration options for group: stripe". This section contains several settings, each with a text input field and a checkbox for "Use system value":

Setting	Value	Use system value
Generate Schedules Every [global]	15	<input checked="" type="checkbox"/>
Schedule Ahead for [global]	4	<input checked="" type="checkbox"/>
Missed if Not Run Within [global]	2	<input checked="" type="checkbox"/>
History Cleanup Every [global]	10	<input checked="" type="checkbox"/>
Success History Lifetime [global]	100	<input checked="" type="checkbox"/>
Failure History Lifetime [global]	100	<input checked="" type="checkbox"/>
Use Separate Process [global]	Yes	<input checked="" type="checkbox"/>

At the bottom of the configuration area, there is a "Mail Sending Settings" section with a refresh icon.

You can find more details about Cron configuration in the [Magento documentation](#).



Accepting Card Payments

Note that HTTPS usage needs to be enforced for secure card collection to work. See the [Installation](#) and [Magento Configuration](#) details above.

Customising the Card Input Element

To enable PCI compliance and ensure secure transmission of card details the checkout page uses [Stripe Elements](#), a Stripe-hosted input field. Because this is Stripe-hosted, it won't automatically pick up your themes styling.

There is some basic configuration (font size and font color) enabled within the Stripe Cards module settings. Should you need more customization than that you can modify the styles object in the `app/code/Stripe/CreditCards/view/frontend/web/js/view/payment/method-renderer/stripecreditcards_gateway.js`

You can see all customisation options in the Stripe Elements reference:

<https://stripe.com/docs/stripe-js/reference#element-options>

3D Secure Options

Stripe provides Verified by Visa and MasterCard Secure functionality also known as 3D Secure. When a payment is successfully authenticated it is covered by a [liability shift](#).

The Stripe Cards settings provide a *Enforce 3D Secure* option. If set to Yes, 3D Secure authentication will be enforced to every card that supports 3D Secure and placing the order will only succeed if the authentication was successful. If set to No, 3D Secure authentication will only be applied to cards that explicitly require it.

This can be tested in Sandbox mode with these specific 3D Secure [test cards](#).

Handling Disputed Payments (Chargebacks)

Disputes/Chargebacks are currently not synced between Stripe and Magento. You will need to rely on email notifications from Stripe to get notified about disputed payments. Please



refer to the Stripe documentation for details about [chargeback notifications](#) and [evidence submission](#).

Auth & Capture Options

For card processing there are two modes: **Authorize only**, or **Authorize and capture immediately**. You can switch between these via the *Payment Action* setting. In authorize only mode the funds will be reserved on the customer's card and you will have to capture them by creating an invoice in Magento. In capture immediately mode the invoice is created automatically within Magento when the customer places the order.

Fraud detection with Stripe Radar

Stripe offers advanced fraud detection functionality, called [Stripe Radar](#). In [Authorization only mode](#) you can use the Stripe Dashboard to review authorizations with an elevated risk profile. In addition to Stripe's machine learning based risk evaluation you can define custom rules for which charges should be added into the review queue.

Please review the [Stripe documentation](#) for details on how to review and action authorized payments in your Stripe Dashboard.

Important: approving the review in Stripe will not automatically capture the charge. You will still need to click the capture button in Stripe or create an invoice in Magento.

Stored Cards

Logged-in customers have the option to store their payment details which enables faster checkout for subsequent orders. Payment details are securely stored in the Stripe vault and won't be exposed to your servers.

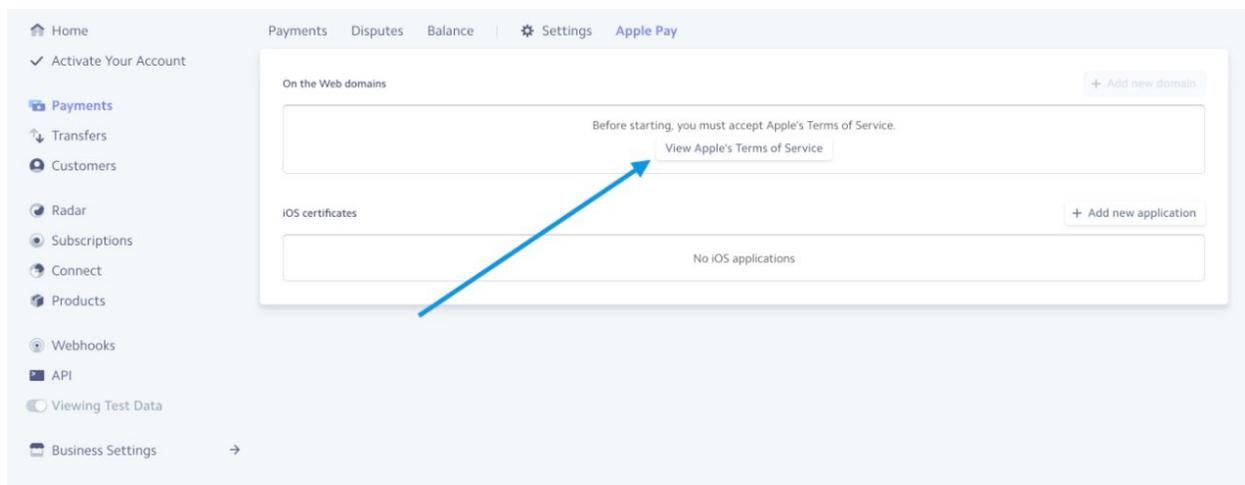


Instant Checkout (Apple Pay / Google Pay)

Note that HTTPS usage needs to be enforced on any page that Instant . See the [Installation](#) and [Magento Configuration](#) details above.

Apple Pay

Before you start using Apple Pay, you must accept the Terms of Service in your Stripe dashboard: https://dashboard.stripe.com/account/apple_pay



Afterward you need to follow the steps to register your domain with Apple Pay as outlined [here](#).

Google Pay

Google Pay is available within the native payment sheet if the customer has a card stored on their device or the Google wallet. For this no additional configuration is required except for HTTPS being enforced on the relevant pages.



Accepting Local Payment Methods

Some of these payment methods are processed asynchronously and require [Webhooks notifications](#) and Cron to be set up correctly.

Aside from standard credit and debit cards, this module supports local payment methods that can be used to target different markets. The following non-card payment methods are supported by the module:

- Alipay
- Bancontact
- Giropay
- iDEAL
- SEPA Direct Debit
- SOFORT (Sofortüberweisung)
- P24 (Przelewy24)

For detailed information about these payment methods, please see the [Stripe Payment Method Guide](#). Before using these methods in production mode you will need to activate them in the [payment settings](#) of your Stripe Dashboard.

SEPA Direct Debit

SEPA Direct Debit is asynchronous payment method that can take up to 14 days to succeed depending on customer country and initial or subsequent payment. Therefore it is important for Webhooks to be set up correctly as they will notify you about payment success and move the order into processing state. Make sure to not ship the order while the payment is pending!

SEPA Direct Debit requires a mandate to be accepted by the account holder. The mandate acceptance text needs to be adapted to include your business name in the Stripe SEPA Direct Debit module settings. When a customer accepts this, Stripe will generate and manage the SEPA Direct Debit mandate for this customer and send them a copy it including the mandate reference via email.

Payment Display Text For Payment Transaction
[website]

By providing your IBAN and confirming this payment, you are authorizing *** YOUR COMPANY NAME *** and Stripe, our payment service provider, to send instructions to your bank to debit your account and your bank to debit your account in accordance with those instructions. You are entitled to a refund from

This text will display on the frontend.

Even if the customer signs the mandate they can dispute the debit with their bank, therefore processing with SEPA Direct Debit carries a certain risk. Make sure to review the [Stripe documentation](#) about this before enabling this method.

In Sandbox mode you can test the different charge outcomes with these [test IBANs](#).

Redirect Flow Methods (Alipay, Bancontact, Giropay, iDEAL, SOFORT, P24)

When enabling these methods, by default they are configured to show up for customers in the [relevant geographies](#). You can change this in the Stripe module settings.

In Sandbox mode Stripe provides a test redirect page where you can test the outcome of the redirect based authentication via the “authorize” and “fail” buttons:

Complete the test payment

This is a test payment of €19.00 with Alipay.

Learn more about Alipay on the [Stripe payments guide](#) →

AUTHORIZE TEST PAYMENT

FAIL TEST PAYMENT

When showing multiple payment methods on the checkout page we recommend to change the Display Billing Address On setting to Payment Page instead of payment method. This setting is located in **Stores > Configuration > Sales > Checkout > Checkout Options**.

Checkout Options

Enable Onepage Checkout <small>[store view]</small>	Yes	<input checked="" type="checkbox"/> Use system value
Allow Guest Checkout <small>[store view]</small>	Yes	<input checked="" type="checkbox"/> Use system value
Enable Terms and Conditions <small>[store view]</small>	No	
Display Billing Address On <small>[store view]</small>	Payment Page	
Maximum Number of Items to Display in Order Summary <small>[store view]</small>	10	<input checked="" type="checkbox"/> Use system value



Refunding a payment in Magento

To issue a refund in Magento, follow these steps:

- In the **Magento Admin Panel** navigate to **Sales > Orders**
- Click on the order that you want to issue a refund for
- In the **Order View** menu on the left select **Invoices**
- In the order overview, select the invoice that you want to create a refund for
- Click **Credit Memo** in the top-right menu
- On the Credit Memo page scroll down to **Refund Totals**, here you have two options:
 - **Refund:** this option will initiate a full or partial refund in Stripe based on the Grand Total
 - **Refund Offline:** this will not automatically create a refund in Stripe and you will need to create a refund separately, for example from the Stripe Dashboard, or outside of Stripe.