

ParadoxLabs Stripe: User Manual

Version 2.1 – For Magento® 2 – Updated 2018-07-18

Table of Contents

Installation	2
If you purchased from Magento Marketplace	2
If you purchased from store.paradoxlabs.com	3
Updating Stripe.....	4
If you purchased from Magento Marketplace	4
If you purchased from store.paradoxlabs.com	4
Configuration	5
General.....	5
Payment Settings	7
Advanced Settings.....	8
Behavior Notes	9
Usage	10
Checkout Payment Form.....	10
Order status page	11
Customer ‘My Payment Data’ account area	12
Admin order form	12
Admin order status page.....	13
Admin customer ‘Payment Data’ account area	13
Admin transaction info	14
Frequently Asked Questions.....	16
Is ParadoxLabs Stripe PCI Compliance?	16
How do I do an online refund from Magento?	16
Error on checkout: “An error occurred on the server. Please try to place the order again.”	17
Do you support Apple Pay?.....	17
How does this payment method handle currency?	17
Technical / Integration Details.....	18
Architecture	18
Custom customer attributes	18

Custom database schema	18
Events.....	18
Magento API	19
Split Database	23
Support	23

Installation

The installation process differs based on where you purchased our extension.

If you purchased from Magento Marketplace

NOTE: You will not be able to install by downloading the extension files from Marketplace.

The Marketplace download does not include all of the necessary files. You must install using either the Web Setup Wizard or Composer, with the following directions.

Step 1: Install

We strongly recommend installing, configuring, and testing all extensions on a development website before installing and using them in production.

If you encounter any problems during this process, please contact Magento Marketplace Support.

Via Web Setup Wizard (recommended)

Follow the official guide here to install using the Web Setup Wizard:

http://docs.magento.com/marketplace/user_guide/quick-tour/install-extension.html

OR Via Composer (command-line/SSH)

If you would prefer, you can also install using Composer rather than the Web Setup Wizard. This requires proficiency with your server's command line. Ensure your server has composer set up and linked to your Magento Marketplace account (including repository <https://repo.magento.com>). Then in SSH, from your site root, run the following commands:

```
composer require paradoxlabs/stripe:*
php bin/magento module:enable -c ParadoxLabs-TokenBase ParadoxLabs_Stripe
php bin/magento setup:upgrade
```

If your site is in production mode, you will also need to run these commands to recompile sources:

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

These commands should load and install the extension packages from the Marketplace repository.

Composer installation is only available for Marketplace purchases.

Step 2: Configure

See the configuration section below.

If you purchased from store.paradoxlabs.com

NOTE: This file upload installation applies **only** to purchases from the ParadoxLabs Store. Marketplace purchases must follow the Marketplace installation directions above.

Step 1: Upload files

Upload all files within the **upload** folder into the root directory of Magento.

Folder in Download	Folder on Server
/upload/app/	→ /app/

Step 2: Run Installation

In SSH, from your site root, run the following commands:

```
composer require stripe/stripe-php:6.*
php bin/magento module:enable -c ParadoxLabs-TokenBase ParadoxLabs_Stripe
php bin/magento setup:upgrade
```

It is very important that you run all three lines above in that exact order.

These will install the Stripe API, enable the module, and then run the installation process.

If your site is in production mode, you will also need to run these commands to recompile sources:

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

Step 3: Configure

See the configuration section below.

Updating Stripe

All extension updates are free. Just follow these directions to update to the latest version.

If you purchased from Magento Marketplace

Via Web Setup Wizard

Follow the official guide here on using the Web Setup Wizard:

https://docs.magento.com/m2/ce/user_guide/system/web-setup-extension-manager.html

If you've already set up and installed with the Web Setup Wizard, you just need to open it, click 'Review Updates', and follow the process.

OR Via Composer (command-line/SSH)

If you installed with composer, you can update using the following commands, in SSH at your site root:

```
composer update paradoxlabs/*
php bin/magento setup:upgrade
```

This will download and update to the latest extension version compatible with your system.

If your site is in production mode, you will also need to run these commands to recompile sources:

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

If you purchased from store.paradoxlabs.com

Step 1: Upload files

Log into your account at store.paradoxlabs.com and download the latest version.

Open the extension archive and extract it onto your composer.

Upload all files within the **upload** folder into the root directory of Magento.

Folder in Download	Folder on Server
/upload/app/	→ /app/

Step 2: Run Update

In SSH, from your site root, run the following commands:

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

If your site is in production mode, you will also need to run these commands to recompile sources:

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

Configuration

Before proceeding: Sign up for a [Stripe merchant account](#) if you don't have one already. You will need to activate your account before you can accept real payments.

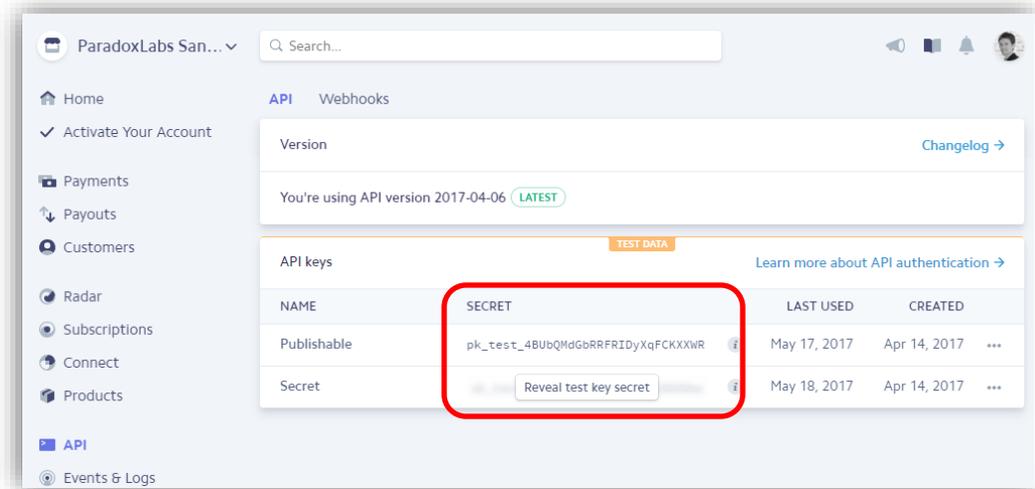
Open your Admin Panel and go to **Admin > Stores > Settings > Configuration > Sales > Payment Methods**. Toward the bottom of the page, you'll find a 'Stripe' settings section like the below.

General

The screenshot shows the 'Configuration' page for the Stripe payment method. The page has a dark sidebar on the left 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 'Stripe'. A 'Save Config' button is in the top right corner. The Stripe logo and 'by ParadoxLabs, inc.' are in the top right. The settings are as follows:

- Version Installed** [website]: 1.0.0.20170518
- API Test Results** [website]: Stripe account ParadoxLabs Sandbox connected successfully (TEST MODE)
- Enabled** [store view]: Yes
- Title** [store view]: Credit Card (Stripe)
- Public API Key** [website]: pk_test_4BubQMdGbRRFRIDyXqFCXXXWR
- Secret API Key** [website]: *****
Get these from your Stripe account API settings. For testing, use the `sk_test_*` and `pk_test_*` values.
- Payment Action** [website]: Authorize
Authorize and Capture to charge on checkout.
- New Order Status** [website]: Pending
Normally 'Pending' if 'Authorize Only' above; 'Processing' if not.
- Show Stripe Badge** [store view]: Solid (dark)
Choose a 'Powered by Stripe' badge to display on CC forms. This can enhance buyer confidence.

- **Version Installed:** This tells you the version of our extension currently installed on your website. Please include this in any support requests.
- **API Test Results:** If you've entered your Public and Secret API Keys, we will automatically connect to Stripe to verify that the API works successfully. If we cannot connect to Stripe, or your Secret Key is incorrect, this will tell you with a red message. Correct the error, then reload the page and it should show 'Stripe account ... connected successfully.' It will also show 'Test Mode' or 'Live Mode' based on the API keys you entered. Change your API keys to go from test mode to live mode, or vice versa.
- **Enable:** Yes to enable the payment method. If disabled, you will still be able to invoice/refund existing orders, but it will not show up as a payment option during checkout.
- **Title:** This controls the payment option label on checkout and order status pages.



- **Public API Key:** Get this from your Stripe account. To find it, log into your Stripe dashboard, then go to **API** on the left navigation. Your Publishable Key will be in the middle of the page, starting with “pk_test_” or “pk_live_”, followed by a bunch of random letters.
 - **Secret API Key:** Get this from your Stripe account. To find it, log into your Stripe dashboard, then go to **API** on the left navigation. Your Secret Key will be in the middle of the page, starting with “sk_test_” or “sk_live_”, followed by a bunch of random letters.
 - **Payment Action:** Choose from the following options.
 - **Save info (do not authorize):** This will require customers to enter a credit card on checkout, and store that credit card in Stripe. The credit card will be validated in the process. No funds will be captured or held from the credit card upon checkout. Invoicing the order will perform a standalone authorize+capture transaction, but is not guaranteed to go through (funds may not be available).
 - **Authorize:** This will authorize the order amount upon checkout, allowing for manual invoicing and capture of the funds later. The order amount will be reserved for exactly 7 days. If you do not invoice within that time, the authorization will expire, and invoicing will perform a standalone authorize+capture transaction instead (which is not guaranteed to go through).
 - **Authorize and Capture:** This will capture all funds immediately when an order is placed.
- Payment processors strongly recommend not capturing funds unless/until you are within three days of fulfilling/shipping the order
- **New Order Status:** Set this to your desired initial status for orders paid via Stripe. Default Magento behavior is ‘Pending’ for Authorize Only, and ‘Processing’ for Authorize and Capture.
 - **Show Stripe Badge:** If yes, checkout will display a ‘Powered by Stripe’ badge next to the payment form.

Payment Settings

Payment Settings

Allow cards to not be stored [website] Yes

If yes, customers can choose whether to save their credit card during checkout.

Accepted Currency [website] US Dollar

Payment from Applicable Countries [store view] All Allowed Countries

Minimum Order Total [store view]

Maximum Order Total [store view]

Sort Order [store view]

- Allow card to not be stored:** If yes, customers will have a ‘Save for next time’ checkbox on checkout. If no, logged in customers will see a message instead: *“For your convenience, this data will be stored securely by our payment processor.”* Guests will never be given the option to store a credit card. Note that all cards are always stored in CIM, regardless of this setting or the customer’s choice. This is necessary for payment processing. If the order is placed as a guest, or the customer chooses to not save their card, it will be automatically purged from all systems 120 days (the maximum refund period) after its last use. This ensures the info is available for edits, captures, and refunds, but respects the customer’s wishes. If a card is ‘not saved’, it will not display under the customer’s saved credit cards (Account > My Payment Data), nor will it be selectable during checkout. Note that as an admin, order ‘edit’ or ‘reorder’ will bypass this, always allowing reuse of the original payment info (unless it was since purged).
- Payment from Applicable Countries:** This setting allows you to limit which countries are able to use this payment method.
- Minimum Order Total:** This setting allows you to set a minimum order value for the payment option. For instance, set to 5 to only allow credit card checkout for orders of \$5 or more.
- Maximum Order Total:** This setting allows you to set a maximum order value for the payment option. For instance, set to 1000 to only allow credit card checkout for orders of \$1000 or less.
- Sort Order:** This setting allows you to change the order of payment options on checkout. Enter a number for this and all other payment methods according to the order you want them to display in.

Advanced Settings

Advanced Settings

Element Style [store view]

```
{
  "base": {"fontSize": "16px"},
  "complete": {},
  "empty": {},
  "invalid": {}
}
```

Use this to change the font and color of the CC field on frontend checkout and card management. Must be valid JSON. See 'style' documentation for options.

Transaction Descriptor [store view]

PDL MAGE2 %ORDERNO%

MAX 22 CHARS, letters/numbers/spaces only. This text will display on your customers' credit card statements. Use %ORDERNO% to include order number. Leave blank for none (default). [More info](#)

Reauthorize on Partial Invoice [website]

Yes

If yes, when you create a partial invoice, we will reauthorize any outstanding balance on the order. This helps guarantee funds, but can cause multiple holds on the card until transactions settle.

Auto-select 'save for next time' [website]

Yes

If yes, will be selected by default during checkout.

Verify SSL [website]

Yes

Strongly recommended. Do not disable unless you get SSL errors and your host is unable to fix them.

- **Element style:** Use this setting to customize the font and colors of the frontend credit card inputs on checkout and account credit card management. The possible keys and values are described in Stripe documentation under 'style': <https://stripe.com/docs/elements/reference#element-options>
- **Transaction Descriptor:** Use this to customize the text that displays on your customers' credit card statements for each order. Use %ORDERNO% to include Magento's order number. Must be no more than 22 characters (letters, numbers, and spaces only)—any longer will be truncated. Leave it blank to use Stripe's default for your account. [More info on transaction descriptors](#)
- **Reauthorize on Partial Invoice:** If yes, and you invoice part of an order, a new authorization will be created for the outstanding order balance (if any). This helps guarantee funds. Any failure during reauthorization is ignored.
- **Auto-select 'save for next time':** If yes, the 'save this card for next time' checkbox will be checked by default. If no, customers will have to explicitly select it to store and reuse their card.
- **Verify SSL:** If yes, the Stripe API connection will be verified against known SSL information for the API. **Do not disable** unless you encounter SSL errors from the API test results and your host is unable to fix the underlying problem. Disabling this will make your store vulnerable to MITM (man-in-the-middle) attacks.

Behavior Notes

For the most part this is a standard Magento payment method, with the addition of the advanced Stored Card functionality and everything related to that. However, there are some things worth note:

User Experience

One difference from standard Magento payment methods is that we use the unified Stripe Elements credit card field everywhere, rather than separate CC type, number, expiration date, and CCV fields. This allows customers to enter all of their CC info in one stream, with a highly optimized user experience. Separators are auto-entered, type is auto-detected, and input moves seamlessly from each field to the next.



Security

Stripe Elements and Stripe.js will always be active. There is no option to turn it off and revert to 'normal' credit card input. All credit card details (CC number, expiration date, CCV) are entered in iframes hosted by Stripe, and transmitted directly to Stripe's servers. At no time will credit card numbers ever touch your server for this payment method. This allows for the best possible mix of user experience and PCI compliance.

Some data about credit cards is kept in your database. This includes the card type, last 4 digits, expiration date, and identifiers, among other data. All of this data is available via the Stripe API, and no part of it is considered 'Confidential Cardholder Data' in the context of PCI compliance.

If you intend to collect payments from other sources using the Magento API, you will need to use the [Stripe API](#) to tokenize the credit card, then send that token to Magento.

Card Validation

Credit cards will always be validated before being stored.

Credit card validation codes will always be validated for new cards. For stored cards, the CCV was already validated, so it will not be required or requested again. The Stripe API does not provide any way to validate CCVs with stored-card transactions.

Card Storage

Credit cards will always be stored in Stripe to allow advanced functionality, even if the customer chooses not to save upon checkout. If the customer chooses not to save, or is a guest, the card will not show up for reuse within Magento. In this case, the card will be automatically purged from all systems after 120 days past its last use. This ensures the info is available for edits, captures, and refunds, but respects the customer's wishes.

There is no automatic card duplicate prevention: If a customer enters the same credit card as a new card 3 times, it will be stored within Stripe 3 times. There is no limit to the number of cards (sources) on a Stripe account.

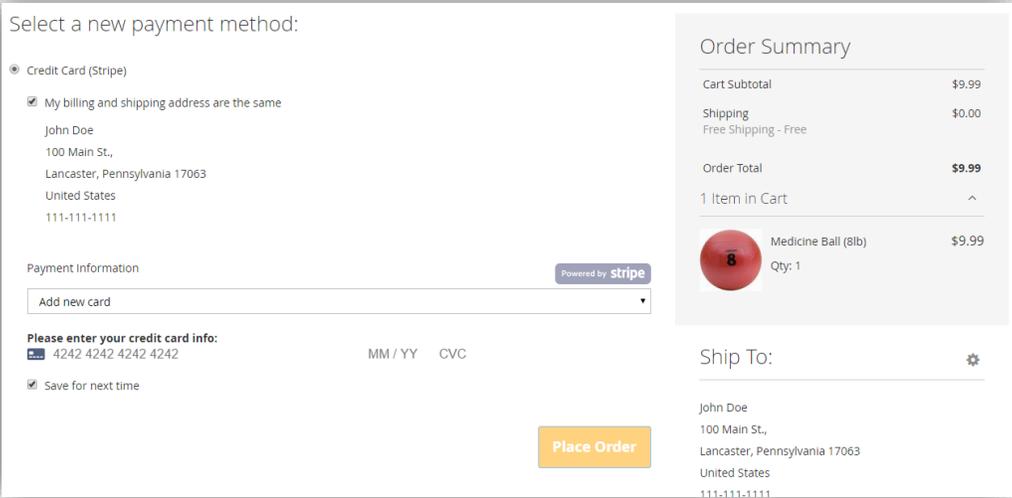
Stripe does not provide a way of changing the underlying card data for an existing stored card, so 'Editing' a stored card within Magento (which involves re-entering the full credit card info) will create a new card record within Stripe and remove the original.

Usage

There isn't much to using this extension in practice: It's a standard Magento payment method, and all interfaces should be pretty self-explanatory. Here's a rundown of what you get:

Checkout Payment Form

The frontend payment form lets you choose/enter billing address and credit card. You can choose an existing card (if any) from the dropdown, or to add a new one.



Select a new payment method:

- Credit Card (Stripe)
 - My billing and shipping address are the same
 - John Doe
 - 100 Main St.,
 - Lancaster, Pennsylvania 17063
 - United States
 - 111-111-1111

Payment Information Powered by stripe

Please enter your credit card info:
 MM / YY CVC

Save for next time Place Order

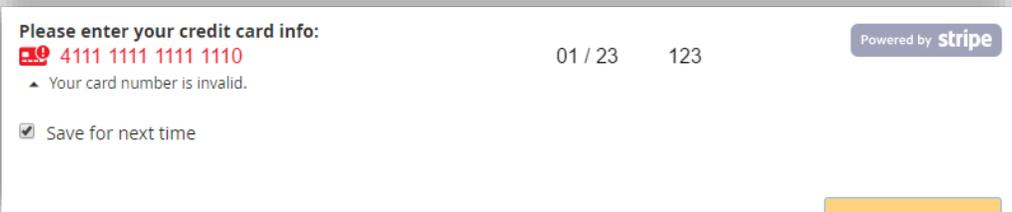
Order Summary

Cart Subtotal	\$9.99
Shipping	\$0.00
Free Shipping - Free	
Order Total	\$9.99
1 Item in Cart ^	
 Medicine Ball (8lb)	\$9.99
Qty: 1	

Ship To: ⚙️

John Doe
 100 Main St.,
 Lancaster, Pennsylvania 17063
 United States
 111-111-1111

Credit card type is detected automatically, and any format errors are immediately displayed below the input field.



Please enter your credit card info: Powered by stripe

 01 / 23 123

▲ Your card number is invalid.

Save for next time

If the customer has stored cards, their most recent one will be selected by default:

Select a new payment method:

Credit Card (Stripe)

My billing and shipping address are the same

John Doe
100 Main St.,
Lancaster, Pennsylvania 17063
United States
111-111-1111

Payment Information Powered by **stripe**

Visa XXXX-4242

Place Order

[Apply Discount Code](#) ▾

Order Summary

Cart Subtotal	\$9.99
Shipping Free Shipping - Free	\$0.00
Order Total	\$9.99
1 Item in Cart ^	
 Medicine Ball (8lb) Qty: 1	\$9.99

Ship To: ⚙

John Doe
100 Main St.,
Lancaster, Pennsylvania 17063

Order status page

Order # 000000416 Pending

Order Date: May 18, 2017

[Reorder](#) [Print Order](#)

- [Account Dashboard](#)
- [Account Information](#)
- [Address Book](#)
- [My Orders](#)
- [My Downloadable Products](#)
- [Newsletter Subscriptions](#)
- [Stored Payment Methods](#)
- [Billing Agreements](#)
- [My Product Reviews](#)
- [My Payment Data](#)
- [My Wish List](#)
- [My Subscriptions](#)

Recently Ordered

Medicine Ball (8lb)

Add to Cart [View All](#)

My Wish List

Items Ordered

Items Ordered

Product Name	SKU	Price	Qty	Subtotal
Medicine Ball (8lb)	med-ball	\$9.99	Ordered: 1	\$9.99
			Subtotal	\$9.99
			Shipping & Handling	\$0.00
			Grand Total	\$9.99

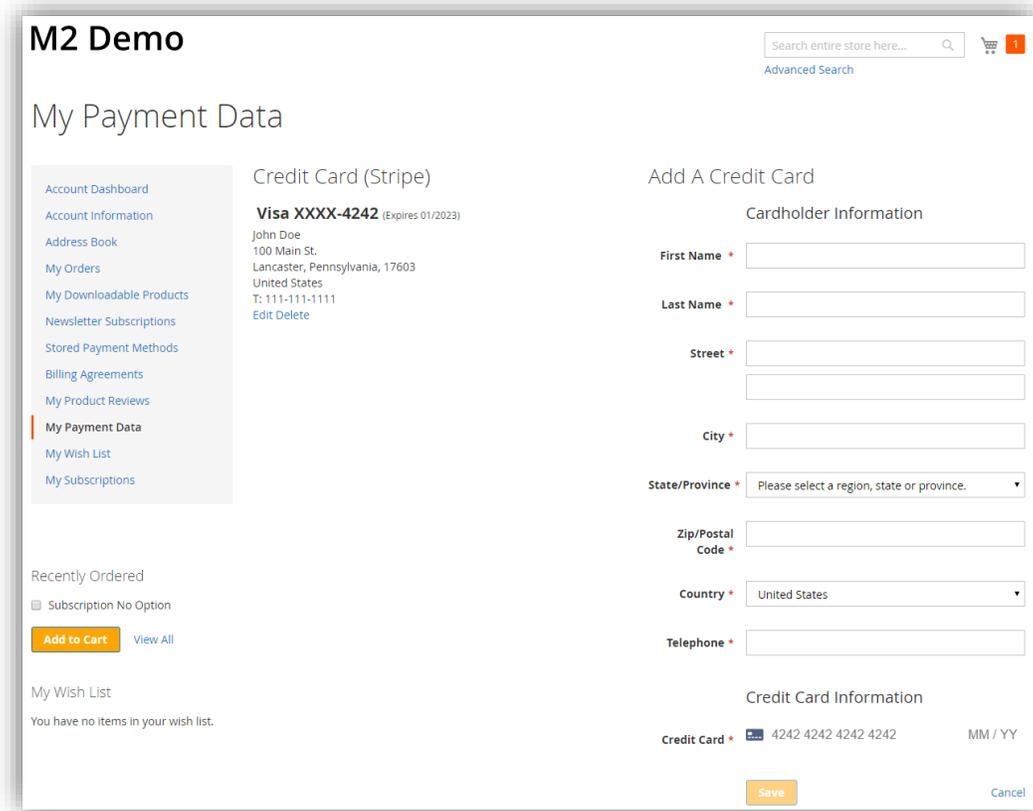
[Back to My Orders](#)

Order Information

<p>Shipping Address</p> <p>John Doe 100 Main St. Lancaster, Pennsylvania, 17063 United States T: 111-111-1111</p> <p>Billing Address</p> <p>John Doe 100 Main St. Lancaster, Pennsylvania, 17063 United States T: 111-111-1111</p>	<p>Shipping Method</p> <p>Free Shipping - Free</p> <p>Payment Method</p> <p>Credit Card (Stripe)</p> <p>Credit Card Type</p> <p>Visa</p> <p>Credit Card Number</p> <p>XXXX-4242</p>
--	---

Customer 'My Payment Data' account area

The My Payment Data section allows customers to see their stored cards, add, edit, and delete.



Note that cards associated with an open (uncaptured) order cannot be edited or deleted. They will display a 'Card In Use' message in place of the buttons. As soon as all orders paid by the card are completed, the 'Edit' and 'Delete' buttons will appear.

To prevent abuse, this section will only be available to customers after they have placed a successful order. If a customer attempts to access the page before then, they'll be redirected to the Account Dashboard with the message, "My Payment Data will be available after you've placed an order." Also to prevent abuse, if a customer receives errors trying to save a card five times, they will be blocked from access for 24 hours with the message, "My Payment Data is currently unavailable. Please try again later." Both of these behaviors can be adjusted or disabled if necessary; please contact us if you have a problem.

Admin order form

The admin form displays the same options as frontend checkout.

Payment & Shipping Information

Payment Method

Credit Card (Stripe)

Powered by **stripe**

Credit Card*

Save for next time

Admin order status page

The admin panel shows extended payment info after placing an order, including transaction ID and validation results. These details are not visible to the customer at any time.

Address Information

<p>Billing Address Edit</p> <p>John Doe 100 Main St. Lancaster, Pennsylvania, 17603 United States T: 111-111-1111</p>	<p>Shipping Address Edit</p> <p>John Doe 100 Main St. Lancaster, Pennsylvania, 17063 United States T: 111-111-1111</p>
--	---

Payment & Shipping Method

<p>Payment Information</p> <p>Credit Card (Stripe)</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 70%;">Credit Card Type:</td> <td>Visa</td> </tr> <tr> <td>Credit Card Number:</td> <td>XXXX-4242</td> </tr> <tr> <td>Transaction ID:</td> <td>ch_1AKs0SDS2ltrz1NCaoz5xdUV</td> </tr> <tr> <td>AVS Street-1 Check:</td> <td>Pass</td> </tr> <tr> <td>AVS Zipcode Check:</td> <td>Pass</td> </tr> <tr> <td>CVC Check:</td> <td>Pass</td> </tr> </table> <p><small>The order was placed using USD.</small></p>	Credit Card Type:	Visa	Credit Card Number:	XXXX-4242	Transaction ID:	ch_1AKs0SDS2ltrz1NCaoz5xdUV	AVS Street-1 Check:	Pass	AVS Zipcode Check:	Pass	CVC Check:	Pass	<p>Shipping & Handling Information</p> <p>Free Shipping - Free \$0.00</p>
Credit Card Type:	Visa												
Credit Card Number:	XXXX-4242												
Transaction ID:	ch_1AKs0SDS2ltrz1NCaoz5xdUV												
AVS Street-1 Check:	Pass												
AVS Zipcode Check:	Pass												
CVC Check:	Pass												

Admin customer 'Payment Data' account area

Viewing a customer, you will see an added 'Payment Data' tab. This shows all of the same information with all of the same functionality as the equivalent frontend section. You can use this to view, add, edit, and delete your customers' stored Stripe payment data.

CUSTOMER INFORMATION

- Customer View
- Account Information
- Addresses
- Orders
- Billing Agreements
- Newsletter
- Product Reviews
- Payment Data
- Wish List
- Subscriptions

Payment Data

Visa XXXX-4242 (Expires 01/2023) [Card in Use](#)

John Doe
100 Main St.
Lancaster, Pennsylvania, 17603
United States
T: 111-111-1111

Add A Credit Card

Cardholder Information

* First Name

* Last Name

* Street

Street Address 2

* City

* State/Province

* Zip/Postal Code

Admin transaction info

Viewing an order, you can also see full transaction info from the 'Transactions' tab.

#000000416 paradox

← Back Cancel Send Email Void Hold Invoice Ship Reorder **Edit**

ORDER VIEW

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

Search [Reset Filter](#) 1 records found 20 per page 1 of 1

ID	Order ID	Transaction ID	Parent Transaction ID	Payment Method	Transaction Type	Closed	Created
From <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	From <input type="text"/>
To <input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	To <input type="text"/>
185	000000416	ch_1AK90SD52trz1NCaoz5x0UV		Credit Card (Stripe)	Authorization	No	May 18, 2017, 1:01:45 PM

Click into a transaction, and you'll see all of the raw transaction data returned by Stripe. This same data is accessible from the transaction record in code.



#ch_1AKsUbDS2ltrz1NCPQ8Bbp2s

Transaction Details

Key	Value
id	ch_1AKsUbDS2ltrz1NCPQ8Bbp2s
object	charge
amount	999
amount_refunded	0
application	
application_fee	
balance_transaction	
captured	0
created	1495128773
currency	usd
customer	cus_AgENBk5OAXhLwd
description	000000419
destination	
dispute	
failure_code	
failure_message	
invoice	
livemode	0
on_behalf_of	
order	
outcome.network_status	approved_by_network

-  DASHBOARD
-  SALES
-  PRODUCTS
-  CUSTOMERS
-  MARKETING
-  CONTENT
-  REPORTS
-  STORES
-  SYSTEM
-  FIND PARTNERS & EXTENSIONS

Frequently Asked Questions

Is ParadoxLabs Stripe PCI Compliance?

PCI compliance is a complex and multifaceted issue, covering every aspect of your business. We can't guarantee that your business is PCI-compliant. That depends on your server, passwords, business processes, regular security scans, any other payment methods, and a lot more. What we can tell you is that this extension will not prevent you from being PCI compliant. We don't log confidential cardholder data or do anything else that would bring you under scrutiny.

This extension implements **Stripe Elements** for all credit card forms, and does not support collecting credit card data by any other means. According to Stripe's official documentation, this makes the ParadoxLabs Stripe payment method eligible for **PCI v3.2 Self-Assessment Questionnaire A (SAQ A)**, the simplest possible SAQ form.

For more information, see the Stripe documentation: [PCI DSS guidelines](#)

All Stripe users must be compliant with the PCI Data Security Standards (PCI DSS). If you're using either [Stripe Checkout](#) or [Stripe Elements](#) to collect payment details, you are eligible for the simplest PCI validation method: [Self-Assessment Questionnaire A \(SAQ A\)](#). Elements and Checkout host all form inputs containing card data within an IFRAME served from Stripe's domain.

As long as you serve your payment pages over TLS, and use either Checkout or Elements as the only way of handling card information, Stripe automatically creates a combined SAQ A and Attestation of Compliance (AOC) for you. This pre-filled document is available in your account's [compliance settings](#).

Note that you **must** have SSL (TLS) enabled on all checkout and login forms, and that this eligibility **only** applies to this specific payment method. Any other payment methods or credit card handling your business may perform will have its own SAQ eligibility, and may require you to complete a more stringent SAQ form (A-EP or D).

For details on the SAQ types and what eligibility means, see "[Self-Assessment Questionnaire Instructions and Guidelines \(3.2\)](#)" (PDF, by PCI Standards Security Council).

For more information on Stripe's security policies and infrastructure, see Stripe documentation: [Security at Stripe](#)

How do I do an online refund from Magento?

In order to process an 'online' refund through Stripe, you have to go to the **invoice** you want to refund, and click the 'Credit Memo' button from there.

If you've done that correctly, at the bottom of the page you should see a button that says 'Refund'.

If you only have one button that says 'Refund Offline', it's because you clicked 'Credit Memo' from the order instead of from the invoice.

The reason for this is that the refund needs to be associated with a particular capture transaction. An order can contain any number of capture transactions, but every capture has an invoice that's directly related. You refund an invoice, not an order.

Error on checkout: “An error occurred on the server. Please try to place the order again.”

Magento made a change in 2.1.x that means no payment error messages actually make it out to the customer. When these error messages occur, the underlying error is usually some payment failure, like AVS failure, or invalid CCV, or transaction declined. These messages will be recorded in the transaction log (`var/log/tokenbase.log`), but the customer will only ever be given the generic failure message. Yes, this makes for bad user experience, but it's not something we can control.

The issue is resolved in Magento 2.2. If you are still on 2.1, you can fix it by overwriting two core files with the corresponding files from 2.2:

`vendor/magento/module-checkout/Model/GuestPaymentInformationManagement.php` to [new version](#)

`vendor/magento/module-checkout/Model/PaymentInformationManagement.php` to [new version](#)

Making these changes will mean customers get the precise error message we intend, and can fix their payment information accordingly.

Do you support Apple Pay?

This should be possible, but it's a feature we're still exploring. It is not currently supported by the extension.

How does this payment method handle currency?

Transactions are automatically processed in the base currency for the website customers are purchasing from. Any alternate currencies selected on the frontend are converted to the website's base currency by Magento's built-in currency handling, based on conversion rates provided by a web service configured in your Magento Admin Panel.

Magento allows for a separate base currency per website, if configured to do so. In order to define explicit prices in multiple currencies, each currency must have its own website where it is set as the base currency. All currency setup is configured outside of our payment extension settings.

Technical / Integration Details

Architecture

The payment method code for this method is `paradoxlabs_stripe`.

`ParadoxLabs_Strip` is the payment method module, built heavily on the `ParadoxLabs-TokenBase` module. `TokenBase` defines a variety of interfaces and architecture for handling tokenization and stored cards cleanly.

The payment method class is `\ParadoxLabs\Stripe\Model\Method`. This talks to Stripe through `\ParadoxLabs\Stripe\Model\Gateway` (via proxy methods for the Stripe PHP library), and stores card metadata in instances of `\ParadoxLabs\Stripe\Model\Card`. Each of these extends an equivalent abstract class in `TokenBase`, and implements only the details specific to the Stripe API.

Card instances are stored in table `paradoxlabs_stored_card`, and referenced by quotes and orders via a `tokenbase_id` column on payment tables.

In all cases, we strongly discourage any customization by editing our code directly. We cannot support customizations. Use Magento's preferences or plugins to modify behavior if necessary. If your use case isn't covered, let us know.

Custom customer attributes

- `stripe_customer_id`

Custom database schema

- Added table: `paradoxlabs_stored_card`
- Added column: `quote_payment.tokenbase_id`
- Added column: `sales_order_payment.tokenbase_id`

Events

- `tokenbase_before_load_payment_info` (`method`, `customer`, `transport`, `info`): Fires before preparing method-specific information for the order payment info blocks (frontend, admin, and emails). Use this to add additional information to the payment info block.
- `tokenbase_after_load_payment_info` (`method`, `customer`, `transport`, `info`): Fires before preparing method-specific information for the order payment info blocks (frontend, admin, and emails). Use this to add additional information to the payment info block, or modify what's there by default.
- `tokenbase_before_load_active_cards` (`method`, `customer`): Fires before loading a customer's available stored cards.
- `tokenbase_after_load_active_cards` (`method`, `customer`, `cards`): Fires after loading a customer's available stored cards. Use this to modify cards available to the customer or admin.

Magento API

This module supports the Magento API via standard interfaces. You can use it to create, read, update, and delete stored cards.

If you have a specific use case in mind that is not covered, please let us know.

You can generate new cards by creating an order with our payment method (code `paradoxlabs_stripe`), and a valid Stripe credit card token. To place an order with a stored card, pass that card's hash in as `additional_data -> card_id`.

Note that raw credit card numbers (`cc_number`) will not be accepted. You must tokenize the payment data and pass in that token. See Stripe documentation on how to tokenize a credit card in various circumstances.

Available REST API requests below. Some response data has been omitted for brevity.

Create and update (POST, PUT) requests take three objects: `card` with primary card data, `address` with address information, and `additional` for card metadata. In responses, `address` and `additional` will be nested within `card` as `address_object` and `additional_object`. This is done for technical reasons. The data formats differ, and not all fields that are returned can be set via API (EG `in_use`, `label`). This means you cannot take a card record and directly post it back to the API to update.

GET /V1/tokenbase/:cardId (get one card by ID)

Example request:

```
GET /rest/v1/tokenbase/1 HTTP/1.1
Host: {host}
Authorization: Bearer {api_key}
```

Example response:

```
{
  "id": 1,
  "in_use": true,
  "additional_object": {
    "cc_type": "DI",
    "cc_last4": "0012",
    "cc_exp_year": "2019",
    "cc_exp_month": "6",
    "cc_country": "US",
    "object": "card",
    "funding": "credit",
    "address_line1_check": "pass",
    "cvc_check": "pass",
    "fingerprint": "0i4PDLs7u409euEo"
  },
  "address_object": {
    "region": {
      "region_code": "PA",
      "region": "Pennsylvania",
      "region_id": 51
    },
    "region_id": 51,
    "country_id": "US",
    "street": [
      "123 Test Ln."
    ],
    "company": "",
    "telephone": "111-111-1111",
    "postcode": "17603",
    "city": "Lancaster",
    "firstname": "John",
    "lastname": "Doe"
  },
  "customer_email": "email@example.com",
  "customer_id": 1,
}
```

```

"customer_ip": "127.0.0.1",
"profile_id": "cus_Abcdeg3n490ds8",
"payment_id": "card_1ABcPDLs7udh38sdoivnw98Q",
"method": "paradoxlabs_stripe",
"hash": "f7d085165acdfa0ea6a0b...770111",
"active": "1",
"created_at": "2017-08-03 16:31:54",
"updated_at": "2017-09-20 14:24:14",
"last_use": "2017-08-03 16:31:54",
"expires": "2019-06-30 23:59:59",
"label": "Discover XXXX-0012"
}

```

GET /V1/tokenbase/search (get multiple cards, with searchCriteria)

Example request:

```

GET /rest/v1/tokenbase/search?searchCriteria[pageSize]=1 HTTP/1.1
Host: {host}
Authorization: Bearer {api_key}

```

Example response:

```

{
  "items": [
    {
      "id": 1,
      // ... other card info
    }
  ],
  "search_criteria": {
    "filter_groups": [],
    "page_size": 1
  },
  "total_count": 51
}

```

See also: [Search using REST APIs](#) (Magento DevDocs)

POST /V1/tokenbase (create card)

Example request:

```

POST /rest/v1/tokenbase HTTP/1.1
Host: {host}
Authorization: Bearer {api_key}
Content-Type: application/json

{
  "card": {
    "customer_email": "email@example.com",
    "customer_id": 1,
    "customer_ip": "",
    "profile_id": "cus_Abcdeg3n490ds8",
    "payment_id": "card_1ABcPDLs7udh38sdoivnw98Q",
    "method": "paradoxlabs_stripe",
    "active": "1",
    "created_at": "2017-08-03 16:31:54",
    "last_use": "2017-08-03 16:31:54",
    "expires": "2019-06-30 23:59:59"
  },
  "address": {
    "region": {
      "region_code": "PA",
      "region": "Pennsylvania",
      "region_id": 51
    }
  }
}

```

```

    "region_id": 51,
    "country_id": "US",
    "street": [
      "123 Test Ln."
    ],
    "company": "",
    "telephone": "111-111-1111",
    "postcode": "17603",
    "city": "Lancaster",
    "firstname": "John",
    "lastname": "Doe",
    "vat_id": ""
  },
  "additional": {
    "cc_exp_month": "06",
    "cc_exp_year": "2019",
    "cc_last4": "0012",
    "cc_type": "DI",
    "cc_country": "US",
    "object": "card",
    "funding": "credit",
    "address_line1_check": "pass",
    "cvc_check": "pass",
    "fingerprint": "0i4PDLS7u409euEo"
  }
}

```

Example response:

```

{
  "id": 95,
  "in_use": false,
  "additional_object": {
    "cc_type": "DI",
    "cc_last4": "0012",
    "cc_exp_year": "2019",
    "cc_exp_month": "06",
    "cc_country": "US",
    "object": "card",
    "funding": "credit",
    "address_line1_check": "pass",
    "cvc_check": "pass",
    "fingerprint": "0i4PDLS7u409euEo"
  },
  "address_object": {
    "region": {
      "region_code": "PA",
      "region": "Pennsylvania",
      "region_id": 51
    },
    "region_id": 51,
    "country_id": "US",
    "street": [
      "123 Test Ln."
    ],
    "company": "",
    "telephone": "111-111-1111",
    "postcode": "17603",
    "city": "Lancaster",
    "firstname": "John",
    "lastname": "Doe",
  },
  "customer_email": "email@example.com",
  "customer_id": 1,
  "customer_ip": "127.0.0.1",
  "profile_id": "cus_Abcdeg3n490ds8",
  "payment_id": "card_1ABcPDLS7udh38sdoivnw98Q",
  "method": "paradoxlabs_stripe",
  "hash": "9b83d4683f3d3...2309ccd65b",
  "active": "1",
  "created_at": "2017-09-25 17:41:21",
  "updated_at": "2017-09-25 17:41:21",
  "last_use": "2017-08-03 16:31:54",
  "expires": "2019-06-30 23:59:59",
  "label": "Discover xxxx-0012"
}

```

PUT /V1/tokenbase/:cardId (update card)

Example request:

```
PUT /rest/v1/tokenbase/1 HTTP/1.1
Host: {host}
Authorization: Bearer {api_key}
Content-Type: application/json

{
  "card": {
    "id": 1,
    "customer_email": "email@example.com",
    "customer_id": 1,
    "customer_ip": "127.0.0.1",
    "profile_id": "cus_Abcdeg3n490ds8",
    "payment_id": "card_1ABcPDLs7udh38sdoivnw98Q",
    "method": "paradoxlabs_stripe",
    "hash": "f7d085165acdfa0ea6a0b...770111",
    "active": "1",
    "created_at": "2017-08-03 16:31:54",
    "last_use": "2017-08-03 16:31:54",
    "expires": "2019-06-30 23:59:59"
  },
  "address": {
    "region": {
      "region_code": "PA",
      "region": "Pennsylvania",
      "region_id": 51
    },
    "region_id": 51,
    "country_id": "US",
    "street": [
      "123 Test Ln."
    ],
    "company": "",
    "telephone": "111-111-1111",
    "postcode": "17603",
    "city": "Lancaster",
    "firstname": "John",
    "lastname": "Doe",
    "vat_id": ""
  },
  "additional": {
    "cc_exp_month": "06",
    "cc_exp_year": "2019",
    "cc_last4": "0012",
    "cc_type": "DI",
    "cc_country": "US",
    "object": "card",
    "funding": "credit",
    "address_line1_check": "pass",
    "cvc_check": "pass",
    "fingerprint": "0i4PDLs7u409euEo"
  }
}
```

Example response:

```
{
  "id": 1,
  "in_use": false,
  "additional_object": {
    "cc_type": "DI",
    "cc_last4": "0012",
    "cc_exp_year": "2019",
    "cc_exp_month": "06",
    "cc_country": "US",
    "object": "card",
    "funding": "credit",
    "address_line1_check": "pass",
    "cvc_check": "pass",
    "fingerprint": "0i4PDLs7u409euEo"
  },
}
```

```

"address_object": {
  "region": {
    "region_code": "PA",
    "region": "Pennsylvania",
    "region_id": 51
  },
  "region_id": 51,
  "country_id": "US",
  "street": [
    "123 Test Ln."
  ],
  "company": "",
  "telephone": "111-111-1111",
  "postcode": "17603",
  "city": "Lancaster",
  "firstname": "John",
  "lastname": "Doe",
},
"customer_email": "email@example.com",
"customer_id": 1,
"customer_ip": "127.0.0.1",
"profile_id": "cus_Abcdeg3n490ds8",
"payment_id": "card_1ABcPDLs7udh38sdoivnw98Q",
"method": "paradoxlabs_stripe",
"hash": " f7d085165acdfa0ea6a0b...770111",
"active": "1",
"created_at": "2017-09-25 17:41:21",
"updated_at": "2017-09-25 17:41:21",
"last_use": "2017-08-03 16:31:54",
"expires": "2019-06-30 23:59:59",
"label": "Discover XXXX-0012"
}

```

DELETE /V1/tokenbase/:cardId (delete card by ID)

Example request:

```

DELETE /rest/v1/tokenbase/95 HTTP/1.1
Host: {host}
Authorization: Bearer {api_key}

```

Example response:

```
true
```

Split Database

This module fully supports Magento Enterprise's split database feature. No special setup should be necessary to get it working in a split-database environment. If you encounter any problems, please let us know.

Support

If you have any questions not covered by this document, or something isn't working right, please open a ticket in our support system: support.paradoxlabs.com