First Data / Payeezy: User Manual

Version 2.3 – For Magento® 2.1+ – Updated 2020-02-07

Table of Contents

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

Via Composer
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/firstdata
php bin/magento module:enable -c ParadoxLabs_TokenBase ParadoxLabs_FirstData
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:

```
php bin/magento module:enable -c ParadoxLabs_TokenBase ParadoxLabs_FirstData
php bin/magento setup:upgrade
```

These will enable the module, flush the cache, and trigger the installation process to run.

If your site is not in developer 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 First Data: Payeezy

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:

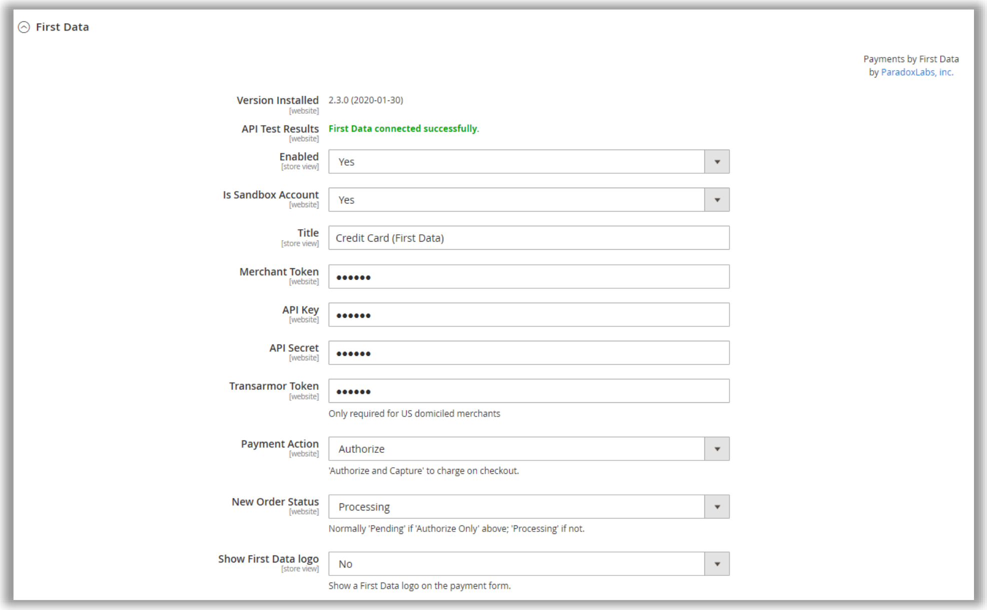
```
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
```

Configuration

Before proceeding: Sign up for a Payeezy developer account and First Data merchant account if you don't have them already. Be sure to request that TransArmor be enabled for your account—this extension won't work without it.

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

General

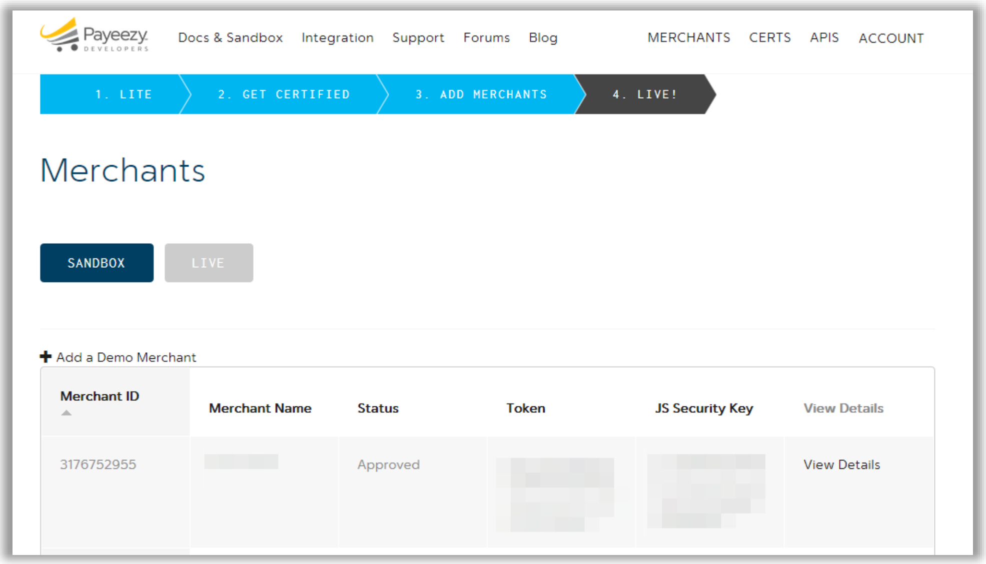


First Data

Payments by First Data
by ParadoxLabs, inc.

Version Installed [website]	2.3.0 (2020-01-30)
API Test Results [website]	First Data connected successfully.
Enabled [store view]	Yes
Is Sandbox Account [website]	Yes
Title [store view]	Credit Card (First Data)
Merchant Token [website]	••••••
API Key [website]	••••••
API Secret [website]	••••••
Transarmor Token [website]	•••••• Only required for US domiciled merchants
Payment Action [website]	Authorize 'Authorize and Capture' to charge on checkout.
New Order Status [website]	Processing Normally 'Pending' if 'Authorize Only' above; 'Processing' if not.
Show First Data logo [store view]	No Show a First Data logo on the payment form.

- **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 a Merchant Token, API Key, API Secret, and Transarmor Token, we will automatically connect to First Data to verify that the API works successfully. If we cannot connect to First Data or your credentials incorrect, this will tell you with a red message. Correct the error, then reload the page and it should show 'First Data connected successfully.'
  - **NOTE:** The API Test currently does not validate the Merchant Token and API Secret. That could prevent checkout even if the API test succeeds.
- **Enabled**: 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.
- **Is Sandbox Account**: Choose 'Yes' if the credentials you entered are for a sandbox account. If this value is not correct, the API Test Results will report *'Your API credentials are invalid.'* To sign up for a sandbox merchant account, navigate to https://provisioning.demo.globalgatewaye4.firstdata.com/signup.
- **Title**: This controls the payment option label on checkout and order status pages.
- **Merchant Token**: This is a secret value from your Payeezy developer account. If you don't know it, log into your First Data developer account, then go to **Merchants** in the navigation. Your Merchant Token will be in the middle of the page. (See Merchants screenshot below for reference.)

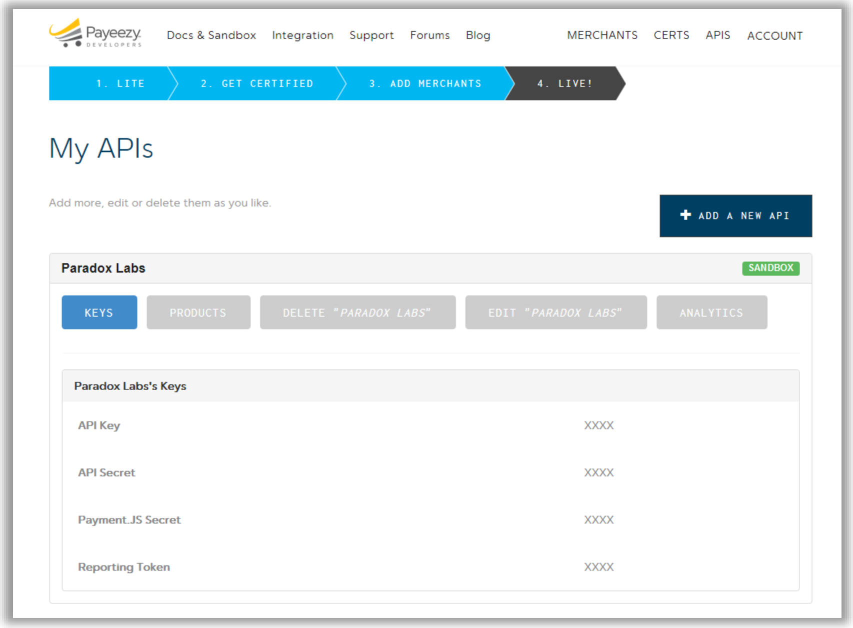


Merchants

Merchant ID	Merchant Name	Status	Token	JS Security Key	View Details
3176752955	[redacted]	Approved	[redacted]	[redacted]	View Details

https://developer.payeezy.com/user/me/merchants

- **API Key**: This is a secret value from your Payeezy developer account. If you don't know it, log into your Payeezy developer account, then go to **APIs** in the navigation. If you do not have an API created, create one to get the API Key **(Make sure you select the appropriate environment)**. If you already have an API, click on it and look under **Keys** to get the API Key. (See My APIs screenshot below for reference)
- **API Secret**: This is a secret value from your Payeezy developer account and is found below the API Key as noted above. (See My APIs screenshot below for reference)



My APIs

Add more, edit or delete them as you like.

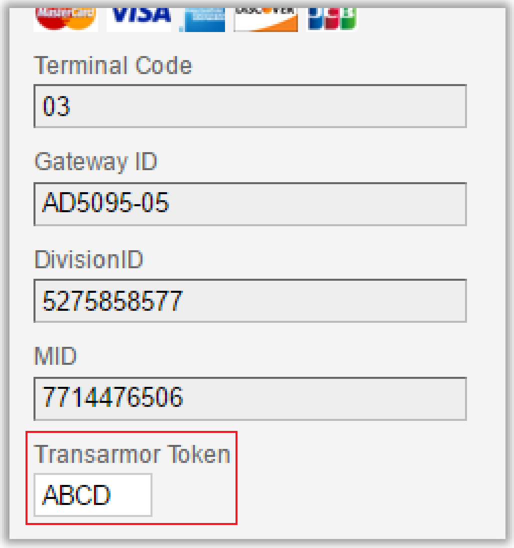
Paradox Labs

Paradox Labs's Keys

API Key	XXXX
API Secret	XXXX
Payment.JS Secret	XXXX
Reporting Token	XXXX

https://developer.payeezy.com/user/me/apps

- **Transarmor Token**: This is a secret value from your First Data account. If you don't know it, log into your First Data merchants account, then go to **Administration > Terminals > Select the appropriate terminal > Details**. WARNING: Your account needs to have TransArmor service enabled. Contact your merchant account representative or contact First Data for TransArmor enablement.

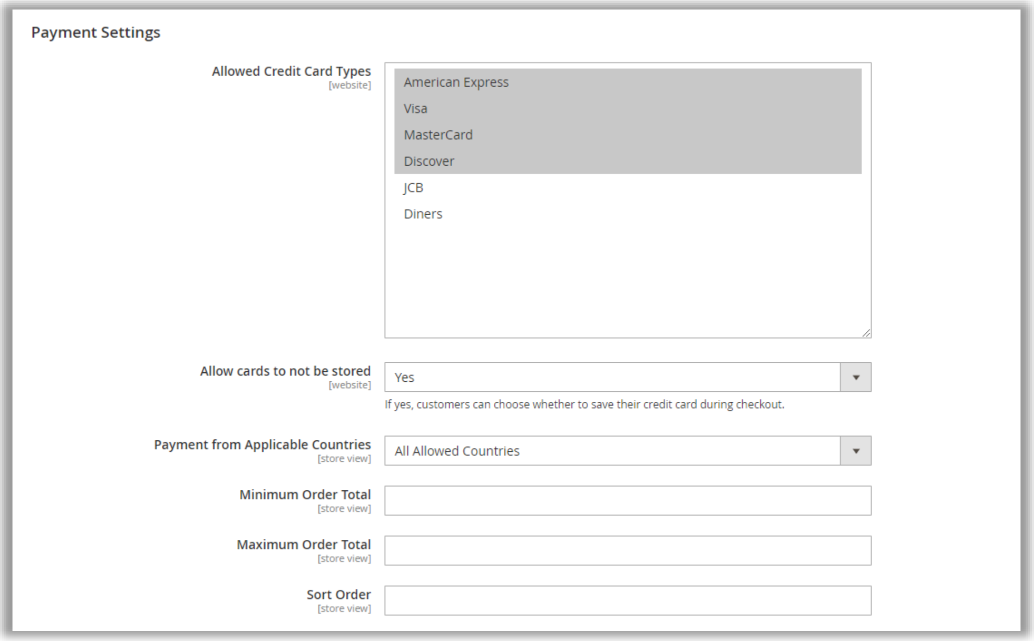


Terminal Code	03
Gateway ID	AD5095-05
DivisionID	5275858577
MID	7714476506
Transarmor Token	ABCD

https://globalgatewaye4.firstdata.com/terminal

- **Payment Action**: Set to 'Authorize and Capture' to capture all funds immediately when an order is placed, or 'Authorize Only' to authorize upon checkout, then manually invoice and capture later. Payment processors strongly recommend not capturing funds unless/until you are within three days of fulfilling (shipping) the purchase.
- **New Order Status**: Set this to your desired initial status for orders paid via First Data. Default Magento behavior is 'Pending' for Authorize Only, and 'Processing' for Authorize and Capture.
- **Show First Data Logo**: If yes, checkout will display an 'First Data' logo next to the payment form.

Payment Settings

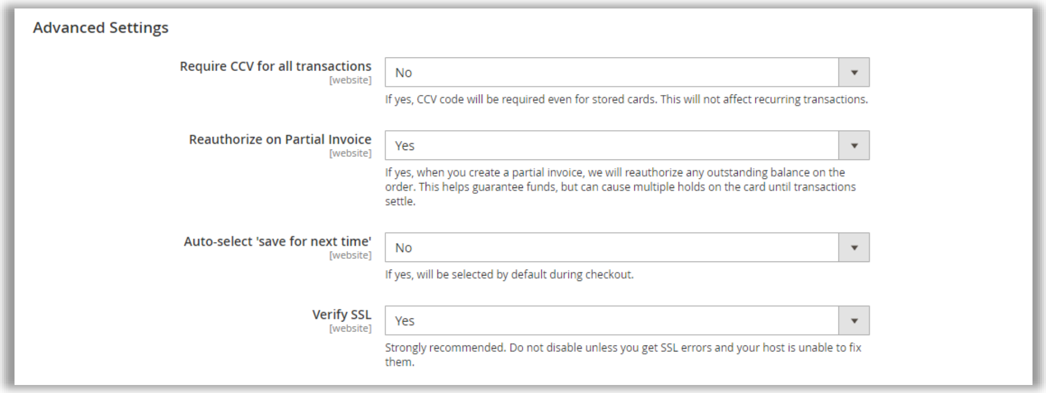


Payment Settings

Allowed Credit Card Types [website]	American Express (selected), Visa (selected), MasterCard (selected), Discover (selected), JCB, Diners
Allow cards to not be stored [website]	Yes If yes, customers can choose whether to save their credit card during checkout.
Payment from Applicable Countries [store view]	All Allowed Countries
Minimum Order Total [store view]	
Maximum Order Total [store view]	
Sort Order [store view]	

- **Allow Credit Card Types**: Choose the CC types you want to allow on checkout.
- **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 First Data, 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' 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 can select it as a payment option.
- **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.
- **Set 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

Require CCV for all transactions [website]: No
If yes, CCV code will be required even for stored cards. This will not affect recurring transactions.

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]: No
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.

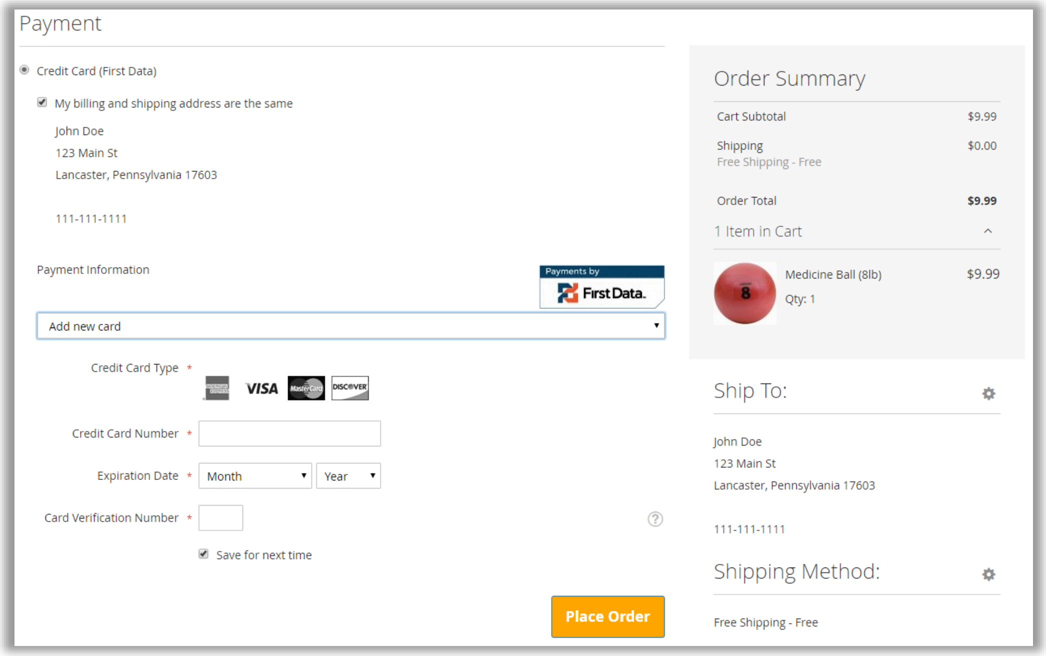
- **Require CCV for all transactions**: If Yes, customers and admins will be required to enter the credit card CCV for all transactions, even with previously stored cards.
- **Reauthorize on Partial Invoice**: If Yes, and you invoice part of an order, a new authorization will be created for the outstanding order balance (if any). This helps guarantee funds but can cause multiple holds on the card until transactions settle. 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 First Data 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.

Usage

There isn't much to using First Data in practice: It's a standard Magento payment method, and all interfaces should be pretty self-explanatory. That being said, here's 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. Credit card type is detected automatically.



Payment

● Credit Card (First Data)

✓ My billing and shipping address are the same

John Doe
123 Main St
Lancaster, Pennsylvania 17603

111-111-1111

Payment Information

Add new card

Credit Card Type *

Credit Card Number *

Expiration Date * Month Year

Card Verification Number *

✓ Save for next time

Place Order

Order Summary

Cart Subtotal	$9.99
Shipping Free Shipping - Free	$0.00
Order Total	$9.99

1 Item in Cart

Medicine Ball (8lb) $9.99
Qty: 1

Ship To:

John Doe
123 Main St
Lancaster, Pennsylvania 17603

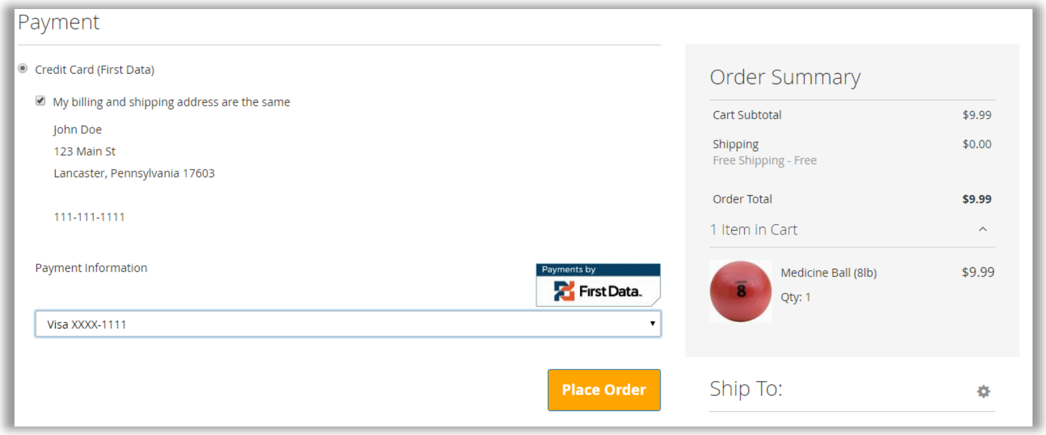
111-111-1111

Shipping Method:

Free Shipping - Free

If a customer re-enters a card they've used before, the existing card will be detected, and the new information (expiration date, billing address, etc.) will be saved on top of it. This happens seamlessly behind the scenes.

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



Payment

● Credit Card (First Data)

✓ My billing and shipping address are the same

John Doe
123 Main St
Lancaster, Pennsylvania 17603

111-111-1111

Payment Information

Visa XXXX-1111

Place Order

Order Summary

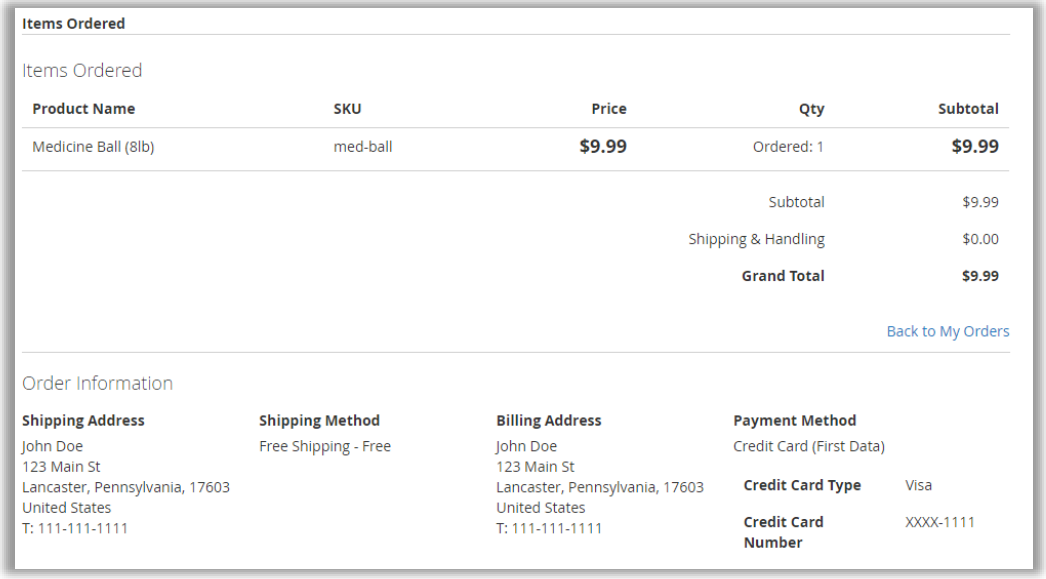
Cart Subtotal	$9.99
Shipping Free Shipping - Free	$0.00
Order Total	$9.99

1 Item in Cart

Medicine Ball (8lb) $9.99
Qty: 1

Ship To:

Order status page



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

Shipping Address
John Doe
123 Main St
Lancaster, Pennsylvania, 17603
United States
T: 111-111-1111

Shipping Method
Free Shipping - Free

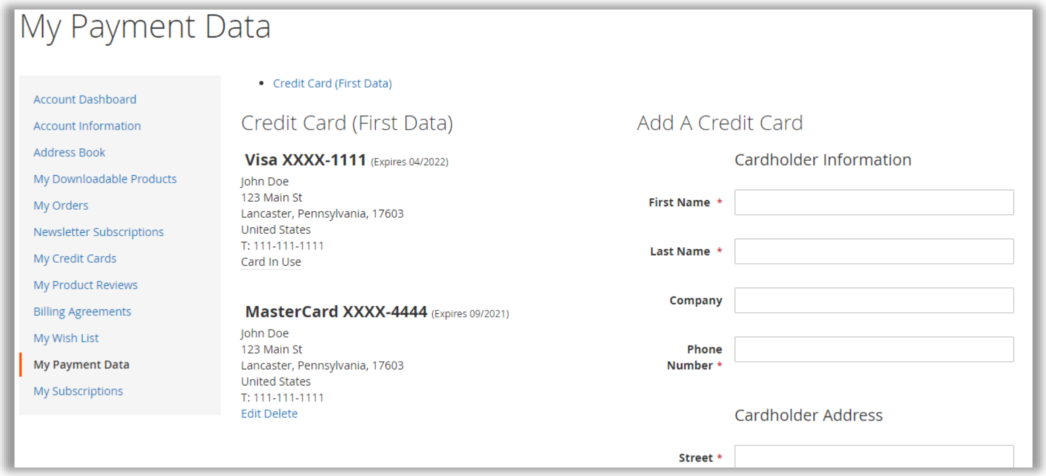
Billing Address
John Doe
123 Main St
Lancaster, Pennsylvania, 17603
United States
T: 111-111-1111

Payment Method
Credit Card (First Data)

Credit Card Type	Visa
Credit Card Number	XXXX-1111

Customer 'My Payment Data' account area

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



My Payment Data

- Credit Card (First Data)

Credit Card (First Data)

Visa XXXX-1111 (Expires 04/2022)
John Doe
123 Main St
Lancaster, Pennsylvania, 17603
United States
T: 111-111-1111
Card In Use

MasterCard XXXX-4444 (Expires 09/2021)
John Doe
123 Main St
Lancaster, Pennsylvania, 17603
United States
T: 111-111-1111
Edit Delete

Add A Credit Card

Cardholder Information

First Name *
Last Name *
Company
Phone Number *

Cardholder Address

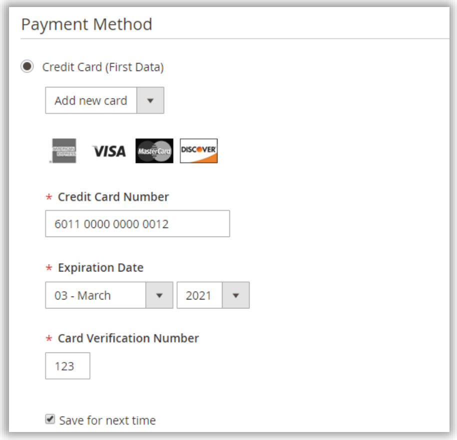
Street *

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, in slightly different format.



Payment Method

● Credit Card (First Data)

Add new card ▾

* Credit Card Number

6011 0000 0000 0012

* Expiration Date

03 - March ▾ 2021 ▾

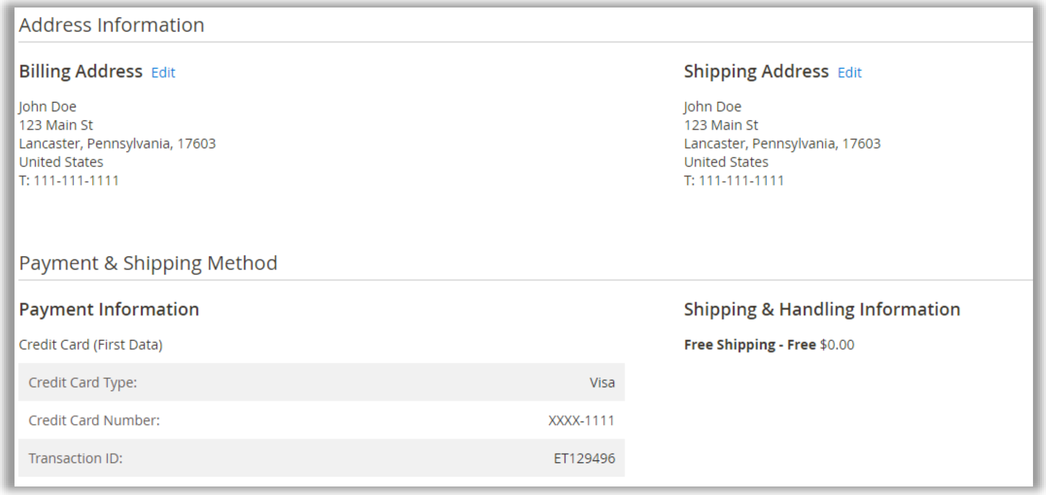
* Card Verification Number

123

☑ 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

Billing Address Edit

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

Shipping Address Edit

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

Payment & Shipping Method

Payment Information

Credit Card (First Data)

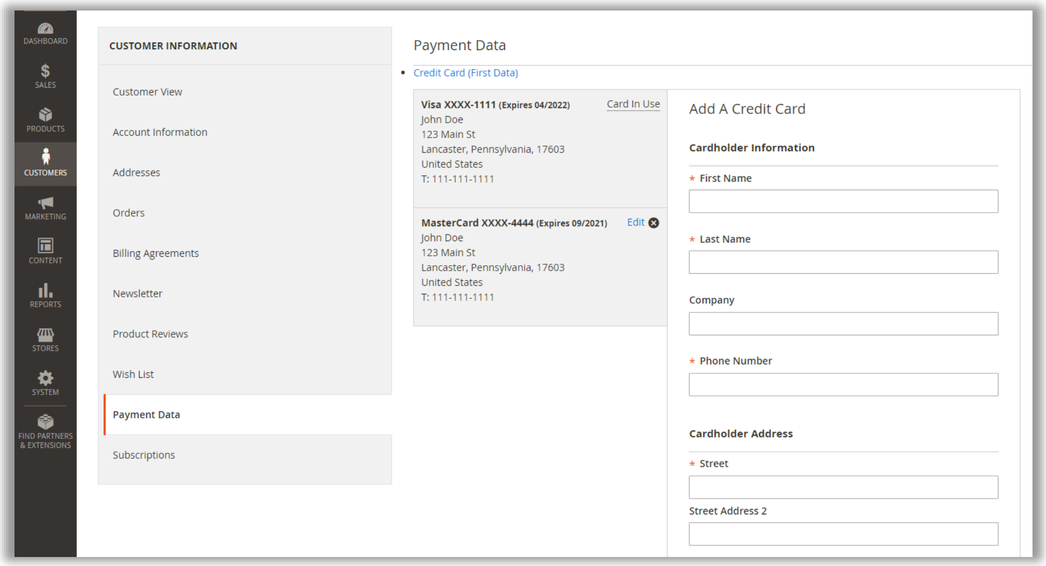
Credit Card Type:	Visa
Credit Card Number:	XXXX-1111
Transaction ID:	ET129496

Shipping & Handling Information

Free Shipping - Free $0.00

Admin customer 'Payment Data' account area

Viewing a customer, you will see an added 'Payment Data' tab. This shows all the same information with all of the same functionality as the equivalent frontend section.



Payment Data

- Credit Card (First Data)

Visa XXXX-1111 (Expires 04/2022) Card In Use
John Doe
123 Main St
Lancaster, Pennsylvania, 17603
United States
T: 111-111-1111

MasterCard XXXX-4444 (Expires 09/2021) Edit
John Doe
123 Main St
Lancaster, Pennsylvania, 17603
United States
T: 111-111-1111

Add A Credit Card

Cardholder Information

* First Name

* Last Name

Company

* Phone Number

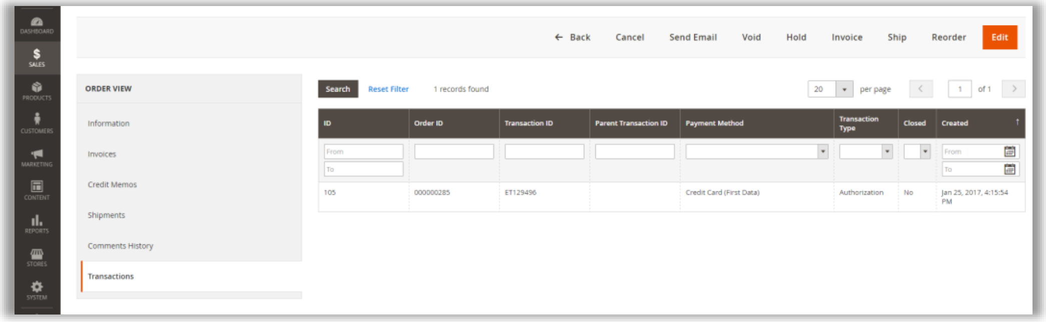
Cardholder Address

* Street

Street Address 2

Admin transaction info

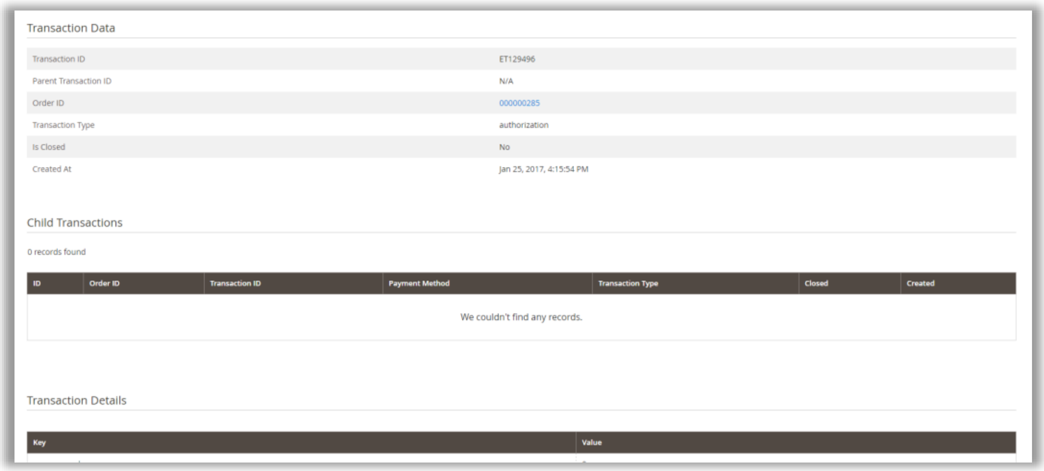
Viewing an order, you can also see full transaction info from the 'Transactions' tab.



1 records found

ID	Order ID	Transaction ID	Parent Transaction ID	Payment Method	Transaction Type	Closed	Created
105	000000285	ET129496		Credit Card (First Data)	Authorization	No	Jan 25, 2017, 4:15:54 PM

Click into a transaction, and you'll see all the raw transaction data from First Data.



Transaction Data

Transaction ID	ET129496
Parent Transaction ID	N/A
Order ID	000000285
Transaction Type	authorization
Is Closed	No
Created At	Jan 25, 2017, 4:15:54 PM

Child Transactions

0 records found

ID	Order ID	Transaction ID	Payment Method	Transaction Type	Closed	Created
We couldn't find any records.						

Transaction Details

Key	Value

Frequently Asked Questions

How do I do an online refund from Magento?

In order to process an 'online' refund through First Data, 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.

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.

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

There are at least two situations that can cause generic error messages like this, depending on your Magento version.

On Magento 2.1:

Magento made a change in 2.1.x that means no payment error messages 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.

On Magento 2.1.13+ / 2.2.4+ / 2.3.0+:

A change to Magento's guest checkout processing made in these versions breaks error handling around data saving during the order process. The net result is that any failure of that type (such as a new credit card failing to validate) will cause a second error “Rolled back transaction has not been completed correctly”. That second error causes the generic error response “an error occurred” to be sent to the user instead of information about the actual payment failure.

This issue is being tracked as a Magento bug, but as of yet the issue is unresolved. One workaround is to remove the `$salesConnection` and `$checkoutConnection` code that is causing the problem from core Magento file `vendor/magento/module-checkout/Model/GuestPaymentInformationManagement.php`, as described in a response comment.

Technical / Integration Details

Architecture

The payment method code for CIM is `paradoxlabs_firstdata`.

`ParadoxLabs_FirstData` 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\FirstData\Model\Method`. This talks to First Data through `\ParadoxLabs\FirstData\Model\Gateway`, and stores card information in instances of `\ParadoxLabs\FirstData\Model\Card`. Each of these extends an equivalent abstract class in TokenBase and implements only the details specific to the First Data 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 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: REST and SOAP

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_firstdata`), and information for a new credit card. To place an order with a stored card, pass that card's hash in as `additional_data` -> `card_id`.

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.

Integration / Admin-Authenticated API Endpoints

These API requests allow solutions acting with an admin user login, OAUTH authentication, or token-based authentication to take action on any card in the system. Data and behavior are not limited.

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"
    },
    "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": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "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": "1234567890",
        "payment_id": "0987654321",
        "method": "paradoxlabs_firstdata",
        "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"
    }
}
```

Example response:

```
{
    "id": 95,
    "in_use": false,
    "additional_object": {
        "cc_type": "DI",
        "cc_last4": "0012",
        "cc_exp_year": "2019",
        "cc_exp_month": "06"
    },
    "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": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "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": "1234567890",
        "payment_id": "0987654321",
        "method": "paradoxlabs_firstdata",
        "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"
    }
}
```

Example response:

```
{
    "id": 1,
    "in_use": false,
    "additional_object": {
        "cc_type": "DI",
        "cc_last4": "0012",
        "cc_exp_year": "2019",
        "cc_exp_month": "06"
    },
    "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": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "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
```

Customer Authenticated API Endpoints

These API requests allow authenticated frontend customers to manage their stored cards. This is intended for headless implementations or app integration where card management needs to be exposed outside of Magento's standard frontend.

Customers will only be able to access and manipulate active cards assigned to their specific customer ID.

Note: These requests are disabled by default. You can enable them at **Admin > Stores > Configuration > Sales > Checkout > ParadoxLabs Payment Module Settings > Enable public API**. Only enable this if you use them.

GET /V1/tokenbase/mine/:cardHash (get one card by hash)

Example request:

```
GET /rest/V1/tokenbase/mine/50b8e326b012e793957215c0361afc4b52434b26 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"
    },
    "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": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "hash": "50b8e326b012e793957215c0361afc4b52434b26",
    "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/mine/search (get multiple cards, with searchCriteria)

Example request:

```
GET /rest/V1/tokenbase/mine/search?searchCriteria[pageSize]=3 HTTP/1.1
Host: {host}
Authorization: Bearer {api_key}
```

Example response:

```
{
    "items": [
        {
            "id": 1,
            // ... other card info
        }
    ],
    "search_criteria": {
        "filter_groups": [],
        "page_size": 3
    },
    "total_count": 5
}
```

See also: Search using REST APIs (Magento DevDocs)

POST /V1/tokenbase/mine (create card)

Example request:

```
POST /rest/V1/tokenbase/mine HTTP/1.1
Host: {host}
Content-Type: application/json
Authorization: Bearer {api_key}

{
    "card": {
        "customer_email": "email@example.com",
        "customer_id": 1,
        "customer_ip": "127.0.0.1",
        "profile_id": "1234567890",
        "payment_id": "0987654321",
        "method": "paradoxlabs_firstdata",
        "active": "1"
    },
    "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"
    }
}
```

Example response:

```
{
    "id": 95,
    "in_use": false,
    "additional_object": {
        "cc_type": "DI",
        "cc_last4": "0012",
        "cc_exp_year": "2019",
        "cc_exp_month": "06"
    },
    "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": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "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/mine/:cardHash (update card by hash)

Example request:

```
PUT /rest/V1/tokenbase/mine/50b8e326b012e793957215c0361afc4b52434b26 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": "1234567890",
        "payment_id": "0987654321",
        "method": "paradoxlabs_firstdata",
        "hash": "50b8e326b012e793957215c0361afc4b52434b26",
        "active": "1",
        "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"
    }
}
```

Example response:

```
{
    "id": 1,
    "in_use": false,
    "additional_object": {
        "cc_type": "DI",
        "cc_last4": "0012",
        "cc_exp_year": "2019",
        "cc_exp_month": "06"
    },
    "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": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "hash": "50b8e326b012e793957215c0361afc4b52434b26",
    "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/mine/:cardHash (delete card by hash)
Example request:

```
DELETE /rest/V1/tokenbase/mine/50b8e326b012e793957215c0361afc4b52434b26 HTTP/1.1
Host: {host}
Authorization: Bearer {api_key}
```

Example response:

```
true
```

Guest API Endpoints

These API requests allow unauthenticated frontend guest users to add and fetch an individual stored card. This is intended for headless implementations or app integration where card management needs to be exposed outside of Magento’s standard frontend. Guests are not able to list, edit, delete, or reuse stored cards, so no API requests are exposed for those actions.

Guests will only be able to access and manipulate active cards, by hash, not assigned to any customer ID.

Note: These requests are disabled by default. You can enable them at **Admin > Stores > Configuration > Sales > Checkout > ParadoxLabs Payment Module Settings > Enable public API**. Only enable this if you use them.

GET /V1/tokenbase/guest/:cardHash (get one card by hash)

Example request:

```
GET /rest/V1/tokenbase/guest/50b8e326b012e793957215c0361afc4b52434b26 HTTP/1.1
Host: {host}
```

Example response:

```
{
    "id": 1,
    "in_use": true,
    "additional_object": {
        "cc_type": "DI",
        "cc_last4": "0012",
        "cc_exp_year": "2019",
        "cc_exp_month": "6"
    },
    "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": 0,
    "customer_ip": "127.0.0.1",
    "profile_id": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "hash": "50b8e326b012e793957215c0361afc4b52434b26",
    "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"
}
```

POST /V1/tokenbase/guest (create card)

Example request:

```
POST /rest/V1/tokenbase/guest HTTP/1.1
Host: {host}
Content-Type: application/json

{
    "card": {
        "customer_email": "email@example.com",
        "customer_id": 0,
        "customer_ip": "127.0.0.1",
        "profile_id": "1234567890",
        "payment_id": "0987654321",
        "method": "paradoxlabs_firstdata",
        "active": "1"
    },
    "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"
    }
}
```

Example response:

```
{
    "id": 95,
    "in_use": false,
    "additional_object": {
        "cc_type": "DI",
        "cc_last4": "0012",
        "cc_exp_year": "2019",
        "cc_exp_month": "06"
    },
    "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": 0,
    "customer_ip": "127.0.0.1",
    "profile_id": "1234567890",
    "payment_id": "0987654321",
    "method": "paradoxlabs_firstdata",
    "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"
}
```

Magento API: GraphQL

For Magento 2.3.1+, this extension supports the GraphQL API for all customer card management. This is intended for PWA and headless implementations where card management needs to be exposed outside of Magento's standard frontend.

Customers will only be able to access and manipulate active cards assigned to their specific customer ID.

Guests will only be able to access and manipulate active cards, by hash, not assigned to any customer ID.

Note: These requests are disabled by default. You can enable them at **Admin > Stores > Configuration > Sales > Checkout > ParadoxLabs Payment Module Settings > Enable public API**. Only enable this if you use them.

We recommend using GraphiQL or the Chrome ChromeiQL browser extension for browsing your store's GraphQL schema and testing API requests.

Queries

tokenBaseCards(hash: String): [TokenBaseCard]

Get the current customer's stored card(s), if any. Takes a card hash (optional); returns one or more `TokenBaseCard` records. If no hash is given, will return all active cards belonging to the customer.

tokenBaseCheckoutConfig(method: String!): TokenBaseCheckoutConfig

Get checkout configuration for the given TokenBase payment method. Takes a TokenBase payment method code, such as `paradoxlabs_firstdata`; returns a `TokenBaseCheckoutConfig`. This returns all data necessary to render and handle the client-side checkout form. Values mirror what is passed to Magento's standard frontend checkout.

Mutations

createTokenBaseCard(input: TokenBaseCardCreateInput!): TokenBaseCard

Create a new stored card. Takes `TokenBaseCardCreateInput`, returns the new stored `TokenBaseCard` if successful.

deleteTokenBaseCard(hash: String!): Boolean

Delete a stored card. Takes a card hash; returns true if successful.

updateTokenBaseCard(input: TokenBaseCardUpdateInput!): TokenBaseCard

Update an existing stored card. Takes `TokenBaseCardUpdateInput`; returns the updated `TokenBaseCard` if successful.

Data Types

TokenBaseCard

A stored payment account/credit card.

```
type TokenBaseCard {
    hash: String                         Card identifier hash
    address: CustomerAddress             Card billing address
    customer_email: String               Customer email
    customer_id: Int                     Customer ID
    customer_ip: String                  Created-by IP
    profile_id: String                   Card gateway profile ID
    payment_id: String                   Card gateway payment ID
    method: String                       Payment method code
    active: Boolean                      Is card active
    created_at: String                   Created-at date
    updated_at: String                   Last updated date
    last_use: String                     Last used date
    expires: String                      Expiration date
    label: String                        Card label
    additional: TokenBaseCardAdditional  Card payment data
}
```

TokenBaseCardAdditional

Details and metadata for a stored CC/ACH.

```
type TokenBaseCardAdditional {
        cc_type: String                                    CC Type
        cc_owner: String                                    CC Owner
        cc_bin: String                                      CC Bin (CC First-6)
        cc_last4: String                                    CC Last-4
        cc_exp_year: String                                 CC Expiration Year
        cc_exp_month: String                                CC Expiration Month
        echeck_account_name: String                         ACH Account Name
        echeck_bank_name: String                            ACH Bank Name
        echeck_account_type: TokenBaseEcheckAccountType     ACH Account type
        echeck_routing_number_last4: String                 ACH Routing Number Last-4
        echeck_account_number_last4: String                 ACH Account Number Last-4
}
```

TokenBaseCheckoutConfig

Checkout configuration for a TokenBase payment method.

```
type TokenBaseCheckoutConfig {
        method: String                                      Payment method code
        useVault: Boolean                                   Are stored cards enabled?
        canSaveCard: Boolean                                Can cards be saved?
        forceSaveCard: Boolean                              Is card saving forced?
        defaultSaveCard: Boolean                            Hash of the default card to select
        isCcDetectionEnabled: Boolean                       Is CC type detection enabled?
        logoImage: String                                   Payment logo image URL (if enabled)
        requireCcv: Boolean                                 Is CVV required for stored cards?
        sandbox: Boolean                                    Is the payment gateway in sandbox mode?
        canStoreBin: Boolean                                Is CC BIN (CC first6) storage enabled?
        availableTypes: [TokenBaseKeyValue]                 Available CC types
        months: [TokenBaseKeyValue]                         Available CC Exp Months
        years: [TokenBaseKeyValue]                          Available CC Exp Years
        hasVerification: Boolean                            Is CVV enabled?
        cvvImageUrl: String                                 CVV helper image URL
}
```

TokenBaseKeyValue

Container for generic key/value data.

```
type TokenBaseKeyValue {
    key: String                                             Generic key
    value: String                                           Generic value
}
```

TokenBaseCardUpdateInput

Input for updating a stored card.

```
input TokenBaseCardUpdateInput {
        hash: String!                                       Card identifier hash to update (required)
        address: CustomerAddressInput                       Card billing address
        customer_email: String                              Customer email
        customer_ip: String                                 Created-by IP
        method: String                                      Payment method code
        active: Boolean                                     Is card active
        expires: String                                     Card expiration date (YYYY-MM-DD 23:59:59)
        additional: TokenBaseCardPaymentInput               Card payment data
}
```

TokenBaseCardCreateInput

Input for creating a stored card.

```
input TokenBaseCardCreateInput {
        address: CustomerAddressInput                       Card billing address
        customer_email: String!                             Customer email (required)
        customer_ip: String                                 Created-by IP
        method: String!                                     Payment method code (required)
        active: Boolean                                     Is card active
        expires: String                                     Card expiration date (YYYY-MM-DD 23:59:59)
        additional: TokenBaseCardPaymentInput               Card payment data
}
```

TokenBaseCardPaymentInput

Payment data for a stored card. Note, the specific fields that are relevant depend on the payment method. This data structure is also used for adding payment data to the cart during checkout.

```
input TokenBaseCardPaymentInput {
        cc_type: String                                    CC Type
        cc_owner: String                                   CC Owner
        cc_bin: String                                     CC Bin (CC First-6)
        cc_last4: String                                   CC Last-4
        cc_number: String                                  CC Number
        cc_cid: String                                     CC CVV
        cc_exp_year: String                                CC Expiration Year
        cc_exp_month: String                               CC Expiration Month
        echeck_account_name: String                        ACH Account Name
        echeck_bank_name: String                           ACH Bank Name
        echeck_account_type: TokenBaseEcheckAccountType    ACH Account Type
        echeck_routing_number: String                      ACH Routing Number
        echeck_account_number: String                      ACH Account Number
        save: Boolean                                      Save the card for later use? (checkout only)
        card_id: String                                    Card identifier hash to use (checkout only)
}
```

GraphQL Query Examples

Some response data has been omitted for brevity.

Fetch card by ID

Example request:

```
{
  tokenBaseCards(hash:"88bb7dc06faad55c77177446ed83047811234008") {
    label,
    expires,
    hash,
    customer_email,
    customer_id,
    profile_id,
    payment_id,
    method,
    active,
    created_at,
    updated_at,
    last_use,
    address {
      region {
        region_code,
        region,
        region_id
      },
      region_id,
      country_id,
      street,
      company,
      telephone,
      postcode,
      city,
      firstname,
      lastname
    },
    additional {
      cc_type,
      cc_last4,
      cc_exp_year,
      cc_exp_month
    }
  }
}
```

Example response:

```
{
  "data": {
    "tokenBaseCards": [
      {
        "label": "Discover XXXX-0012",
        "expires": "2021-03-31 23:59:59",
        "hash": "88bb7dc06faad55c77177446ed83047811234008",
        "customer_email": "roni_cost@example.com",
        "customer_id": 1,
        "profile_id": "1200144368",
        "payment_id": "1506360102",
        "method": "paradoxlabs_firstdata",
        "active": true,
        "created_at": "2019-03-11 16:12:52",
        "updated_at": "2019-04-04 18:01:39",
        "last_use": "2019-04-04 18:01:39",
        "address": {
          "region": {
            "region_code": "MI",
            "region": "Michigan",
            "region_id": 33
          },
          "region_id": 33,
          "country_id": "US",
          "street": [
            "6146 Honey Bluff Parkway"
          ],
          "company": "",
          "telephone": "(555) 229-3326",
          "postcode": "49628-7978",
          "city": "Calder",
          "firstname": "Veronica",
          "lastname": "Costello"
        },
        "additional": {
          "cc_type": "DI",
          "cc_last4": "0012",
          "cc_exp_year": "2021",
          "cc_exp_month": "3"
        }
      }
    ]
  }
}
```

Fetch checkout config

Example request:

```
{
  tokenBaseCheckoutConfig(method:"paradoxlabs_firstdata") {
    method,
    useVault,
    canSaveCard,
    forceSaveCard,
    defaultSaveCard,
    isCcDetectionEnabled,
    logoImage,
    requireCcv,
    sandbox,
    canStoreBin,
    availableTypes {key, value},
    months {key, value},
    years {key, value},
    hasVerification,
    cvvImageUrl
  }
}
```

Example response:

```
{
  "data": {
    "tokenBaseCheckoutConfig": {
      "method": "paradoxlabs_firstdata",
      "useVault": true,
      "canSaveCard": true,
      "forceSaveCard": false,
      "defaultSaveCard": true,
      "isCcDetectionEnabled": true,
      "logoImage": "https://.../images/logo.png",
      "requireCcv": false,
      "sandbox": true,
      "canStoreBin": false,
      "availableTypes": [
        {
          "key": "AE",
          "value": "American Express"
        },
        ...
      ],
      "months": [
        {
          "key": "1",
          "value": "01 - January"
        },
        ...
      ],
      "years": [
        {
          "key": "2019",
          "value": "2019"
        },
        ...
      ],
      "hasVerification": true,
      "cvvImageUrl": "https://.../Magento_Checkout/cvv.png"
    }
  }
}
```

Create card

Example request:

```
mutation {
  createTokenBaseCard(
    input: {
      expires: "2022-12-31 23:59:59",
      customer_ip: "127.0.0.1",
      customer_email: "roni_cost@example.com",
      method: "paradoxlabs_firstdata",
      active: true,
      address: {
        region: {
          region_code: "PA",
          region: "Pennsylvania",
          region_id: 51
        },
        country_id: US,
        street: [
          "123 Test St.",
          "Apt 9"
        ],
        company: "",
        telephone: "111-111-1111",
        postcode: "12345",
        city: "Testcity",
        firstname: "John",
        lastname: "Doe"
      },
      additional: {
        cc_type: "VI",
        cc_number: "4111111111111111",
        cc_exp_year: "2022",
        cc_exp_month: "12",
        cc_cid: "123",
      }
    }
  ) {
    label,
    expires,
    hash,
    customer_email,
    customer_id,
    customer_ip,
    profile_id,
    payment_id,
    method,
    active,
    created_at,
    updated_at,
    last_use,
    address {
      region {
        region_code,
        region,
        region_id
      },
      region_id,
      country_id,
      street,
      company,
      telephone,
      postcode,
      city,
      firstname,
      lastname
    },
    additional {
      cc_type,
      cc_last4,
      cc_exp_year,
      cc_exp_month
    }
  }
}
```

Example response:

```
{
  "data": {
    "createTokenBaseCard": {
      "label": "Visa XXXX-0027",
      "expires": "2021-03-31 23:59:59",
      "hash": "88bb7dc06faad55c77177446ed83047811234008",
      "customer_email": "roni_cost@example.com",
      "customer_id": 1,
      "profile_id": "1200144368",
      "payment_id": "1506360102",
      "method": "paradoxlabs_firstdata",
      "active": true,
      "created_at": "2019-03-11 16:12:52",
      "updated_at": "2019-04-04 18:01:39",
      "last_use": "2019-04-04 18:01:39",
      "address": {
        "region": {
          "region_code": "MI",
          "region": "Michigan",
          "region_id": 33
        },
        "region_id": 33,
        "country_id": "US",
        "street": [
          "6146 Honey Bluff Parkway"
        ],
        "company": "",
        "telephone": "(555) 229-3326",
        "postcode": "49628-7978",
        "city": "Calder",
        "firstname": "Veronica",
        "lastname": "Costello"
      },
      "additional": {
        "cc_type": "VI",
        "cc_last4": "1111",
        "cc_exp_year": "2022",
        "cc_exp_month": "12"
      }
    }
  }
}
```

Delete card

Example request:

```
mutation {
  deleteTokenBaseCard(hash:"88bb7dc06faad55c77177446ed83047811234008")
}
```

Example response:

```
{
  "data": {
    "deleteTokenBaseCard": true
  }
}
```

Place an order

Example request:

```
mutation {
  setPaymentMethodOnCart(
    input: {
      cart_id: "kDy0EKkJmIOxa6H3ceus6MjMaSF9lao1"
      payment_method: {
        code: "paradoxlabs_firstdata",
        tokenbase_data: {
          cc_type: "VI",
          cc_last4: "1111",
          cc_exp_year: "2022",
          cc_exp_month: "08",
          cc_cid: "123",
          cc_number: “4111111111111111",
          save: true
        }
      }
    }
  ) {
    cart {
      selected_payment_method {
        code
      }
    }
  }
  placeOrder(
    input: {
      cart_id: "kDy0EKkJmIOxa6H3ceus6MjMaSF9lao1"
    }
  ) {
    order {
      order_id
    }
  }
}
```

Example response:

```
{
  "data": {
    "setPaymentMethodOnCart": {
      "cart": {
        "selected_payment_method": {
          "code": "paradoxlabs_firstdata"
        }
      }
    },
    "placeOrder": {
      "order": {
        "order_id": "000000255"
      }
    }
  }
}
```

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

Support Policy: https://store.paradoxlabs.com/support.html

License and Terms of Use: https://store.paradoxlabs.com/license.html