

First Data / Payeezy: 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
Configuration	4
General.....	5
Payment Settings	8
Advanced Settings.....	9
Usage	10
Checkout Payment Form.....	10
Order status page	11
Customer ‘My Payment Data’ account area	11
Admin order form	12
Admin order status page.....	12
Admin customer ‘Payment Data’ account area	13
Admin transaction info	14
Frequently Asked Questions.....	15
How do I do an online refund from Magento?	15
How does this payment method handle currency?	15
Error on checkout: “An error occurred on the server. Please try to place the order again.”	15
Technical / Integration Details.....	16
Architecture	16
Custom database schema	16
Events.....	16
Magento API	16
Split Database	21
Support	21

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:

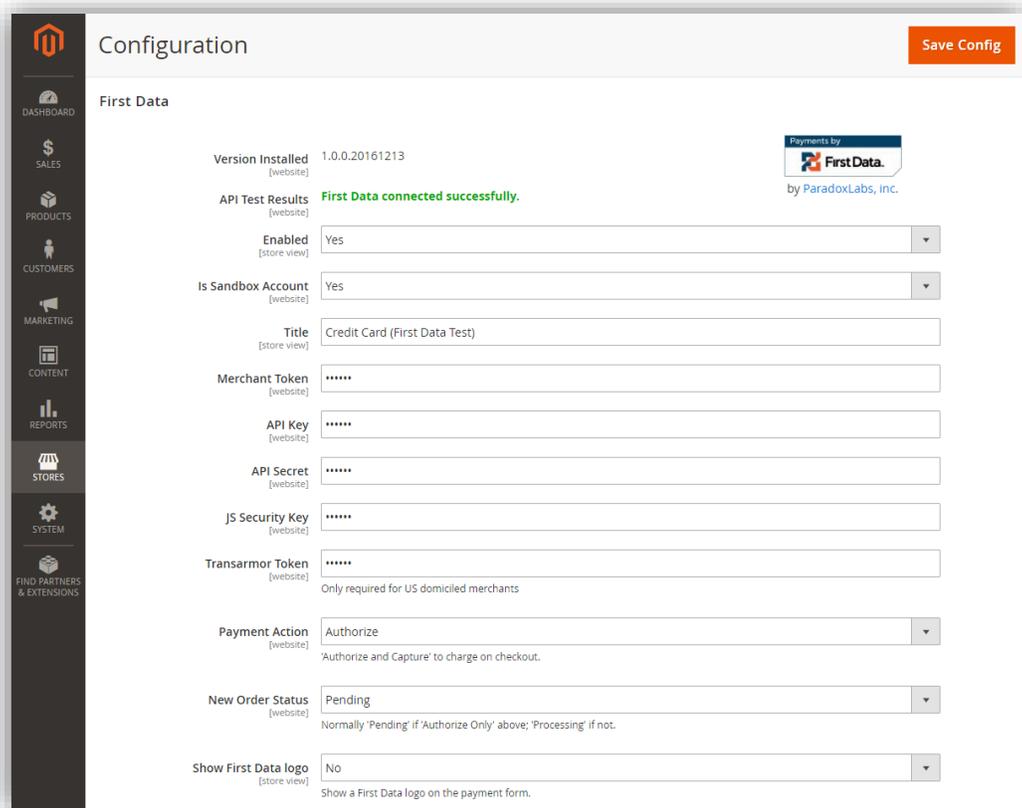
```
php bin/magento setup:di:compile
php bin/magento setup:static-content:deploy
```

Configuration

Before proceeding: Sign up for a [First Data 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



Configuration Save Config

First Data

Version Installed [website] 1.0.0.20161213

API Test Results [website] **First Data connected successfully.**

Enabled [store view] Yes

Is Sandbox Account [website] Yes

Title [store view] Credit Card (First Data Test)

Merchant Token [website] *****

API Key [website] *****

API Secret [website] *****

JS Security Key [website] *****

Transarmor Token [website] *****
Only required for US domiciled merchants

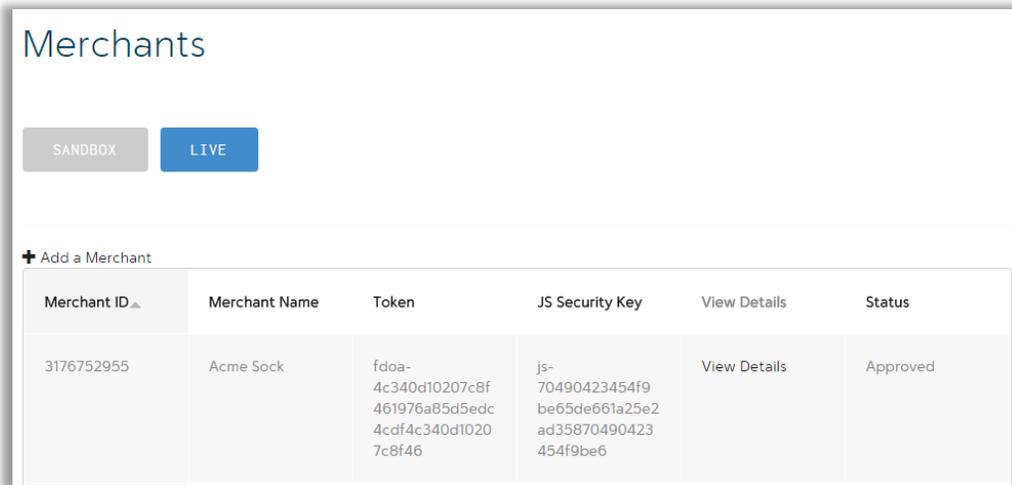
Payment Action [website] Authorize
Authorize and Capture to charge on checkout.

New Order Status [website] Pending
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, JS Security Key, 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.
- **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.
- **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.'** If you want to test, you must

have a sandbox account (separate from your production First Data account). You can create one here: <https://provisioning.demo.globalgatewaye4.firstdata.com/signup>

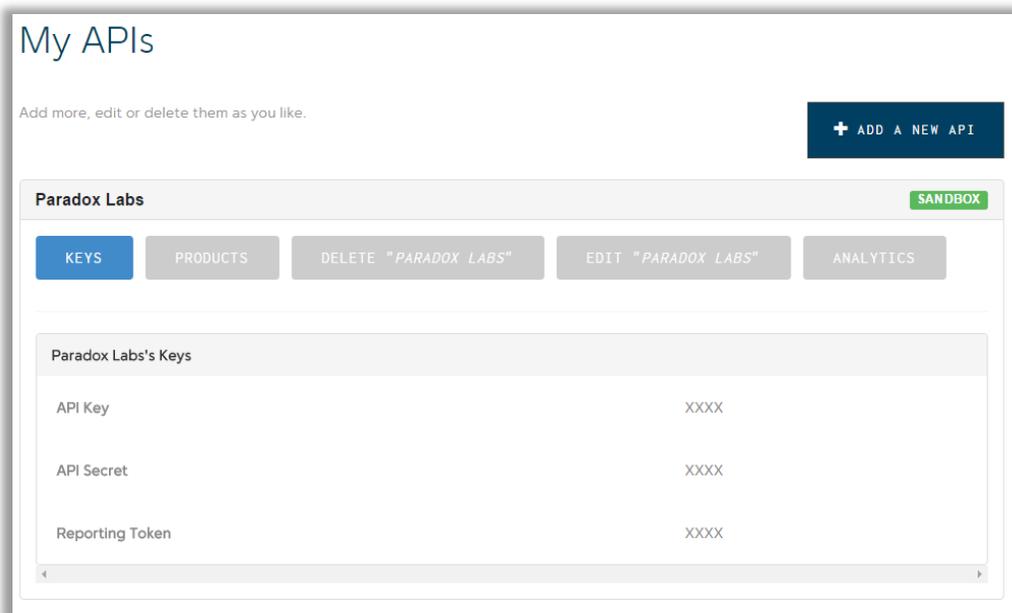


Merchants

SANDBOX LIVE

+ Add a Merchant

Merchant ID ▲	Merchant Name	Token	JS Security Key	View Details	Status
3176752955	Acme Sock	fdoa-4c340d10207c8f461976a85d5edc4cdf4c340d10207c8f46	js-70490423454f9be65de661a25e2ad35870490423454f9be6	View Details	Approved



My APIs

Add more, edit or delete them as you like.

+ ADD A NEW API

Paradox Labs SANDBOX

KEYS PRODUCTS DELETE "PARADOX LABS" EDIT "PARADOX LABS" ANALYTICS

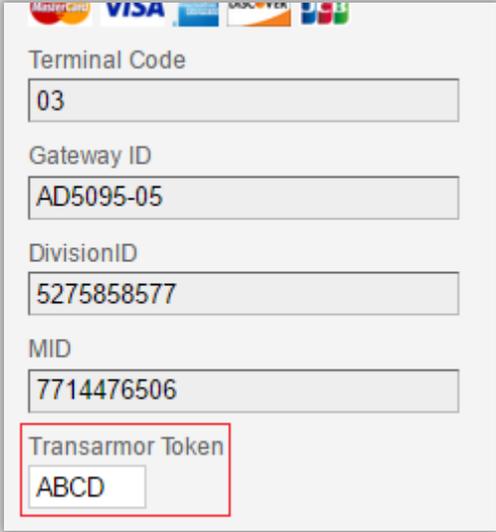
Paradox Labs's Keys

API Key	XXXX
API Secret	XXXX
Reporting Token	XXXX

- **Title:** This controls the payment option label on checkout and order status pages.
- **Merchant Token:** This is a secret value from your First Data 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 above for reference)
- **API Key:** This is a secret value from your First Data developer account. If you don't know it, log into your First Data 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 development**). If you already have an API, click on it and look under **Keys** to get the API Key. (See My APIs screenshot above for reference)
- **API Secret:** This is a secret value from your First Data developer account. If you don't know it, log into your First Data developer account, then go to **APIs** in the navigation. If you do not have an API created,

create one to get the API Secret (**Make sure you select the appropriate development**). If you already have an API, click on it and look under **Keys** to get the API Secret. (See My APIs screenshot above for reference)

- **JS Security Key:** This is a secret value from your First Data developer account. If you don't know it, log into your First Data developer account, then go to **Merchants** in the navigation. Your JS Security Token will be in the middle of the page. (See Merchants screenshot above for reference)
- **Transarmor Token:** This is a secret value from your First Data account. If you don't know it, log into your First Data account, then go to **Administration > Terminal > 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.



The screenshot shows a configuration form with the following fields:

- Terminal Code: 03
- Gateway ID: AD5095-05
- DivisionID: 5275858577
- MID: 7714476506
- Transarmor Token: ABCD (highlighted with a red box)

- **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

Visa

MasterCard

Discover

Other

Allow cards to not be stored [website] ▼

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

Accepted Currency [website] ▼

Payment from Applicable Countries [store view] ▼

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'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 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

Reauthorize on Partial Invoice [website]

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]

If yes, will be selected by default during checkout.

Verify SSL [website]

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

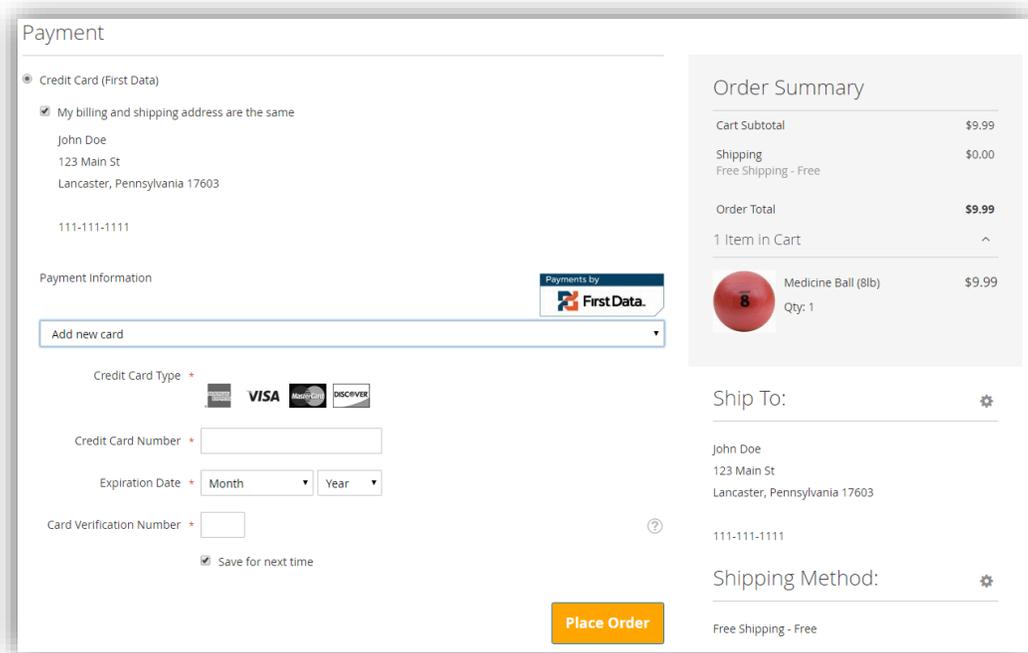
- **Require CCV for all transactions:** If yes, customers and admins will be required to enter the credit card CCV for all transactions, even with previously-stored cards.
- **Reauthorize on Partial Invoice:** If yes, and you invoice part of an order, a new authorization will be created for the outstanding 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

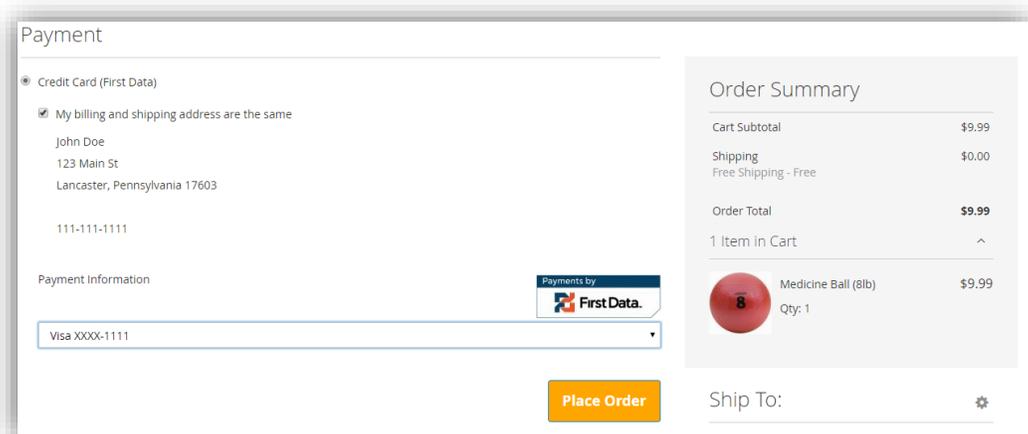
The frontend payment form lets you choose/enter billing address and credit card. You can choose an existing card (if any) from the dropdown, or to add a new one. Credit card type is detected automatically.



The screenshot shows the 'Payment' section of a Magento checkout page. It includes a 'Credit Card (First Data)' section with a checked option 'My billing and shipping address are the same'. The billing address is John Doe, 123 Main St, Lancaster, Pennsylvania 17603, with phone number 111-111-1111. The 'Payment Information' section has a dropdown menu set to 'Add new card' and a 'First Data' logo. Below this are fields for 'Credit Card Type' (with icons for American Express, VISA, MasterCard, and DISCOVER), 'Credit Card Number', 'Expiration Date' (Month and Year dropdowns), and 'Card Verification Number'. A 'Save for next time' checkbox is checked. A 'Place Order' button is at the bottom right. The 'Order Summary' sidebar on the right shows a cart subtotal of \$9.99, shipping of \$0.00, and an order total of \$9.99 for one item: 'Medicine Ball (8lb) Qty: 1'. Shipping and shipping method are also visible.

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:



This screenshot is identical to the one above, but the dropdown menu in the 'Payment Information' section now displays 'Visa XXXX-1111' instead of 'Add new card'. This indicates that a previously used card has been detected and selected by default.

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	Shipping Method	Billing Address	Payment Method	
John Doe 123 Main St Lancaster, Pennsylvania, 17603 United States T: 111-111-1111	Free Shipping - Free	John Doe 123 Main St Lancaster, Pennsylvania, 17603 United States T: 111-111-1111	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 *

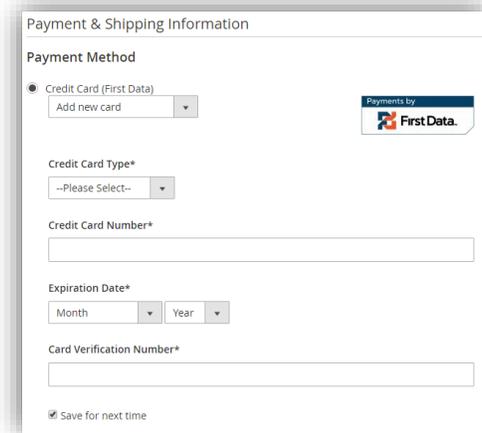
Note that cards associated with an open (uncaptured) order cannot be edited or deleted. They will display a 'Card In Use' message in place of the buttons. As soon as all orders paid by the card are completed, the 'Edit' and 'Delete' buttons will appear.

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

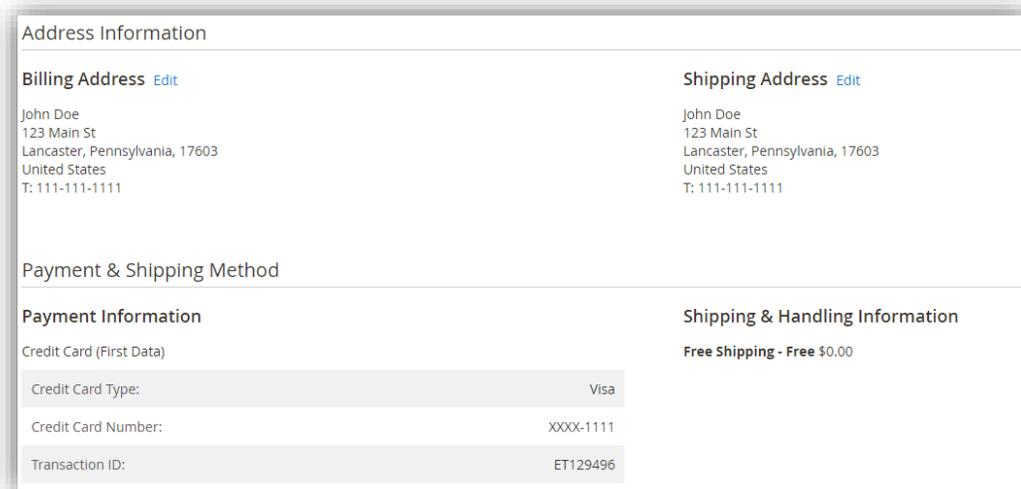
The admin form displays the same options as frontend checkout, in slightly different format.



The screenshot shows a form titled "Payment & Shipping Information". Under the "Payment Method" section, "Credit Card (First Data)" is selected. There is a dropdown menu for "Add new card" and a "Payments by FirstData" logo. Below this, there are fields for "Credit Card Type*" (with a "--Please Select--" dropdown), "Credit Card Number*" (a text input field), "Expiration Date*" (with "Month" and "Year" dropdowns), and "Card Verification Number*" (a text input field). At the bottom, there is a checkbox labeled "Save for next time" which is checked.

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.



The screenshot shows the "Admin order status page" with two main sections: "Address Information" and "Payment & Shipping Method".

Address Information

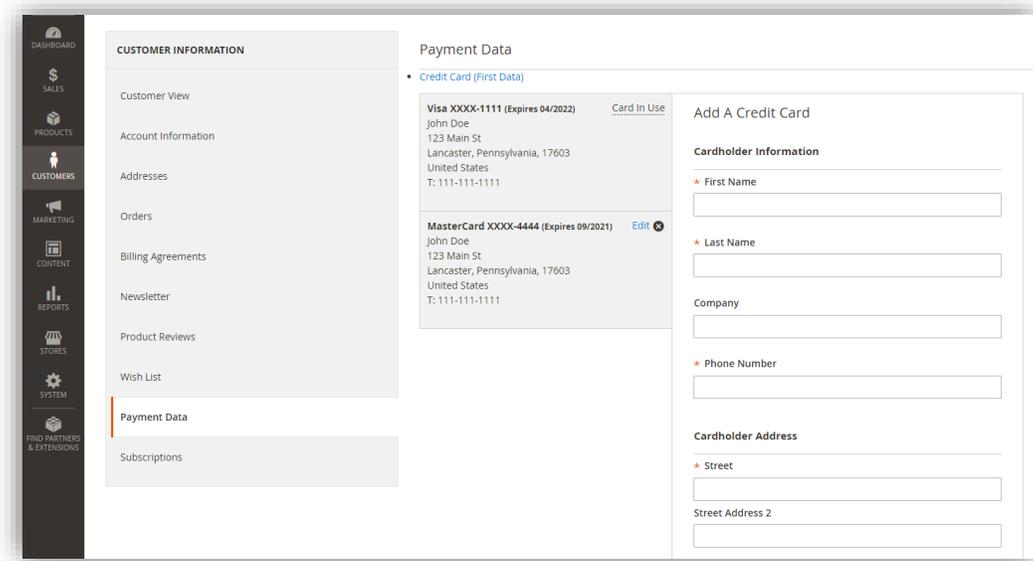
Billing Address Edit	Shipping Address Edit
John Doe 123 Main St Lancaster, Pennsylvania, 17603 United States T: 111-111-1111	John Doe 123 Main St Lancaster, Pennsylvania, 17603 United States T: 111-111-1111

Payment & Shipping Method

Payment Information	Shipping & Handling Information
Credit Card (First Data)	Free Shipping - Free \$0.00
Credit Card Type: Visa	
Credit Card Number: XXXX-1111	
Transaction ID: ET129496	

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.



CUSTOMER INFORMATION

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

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

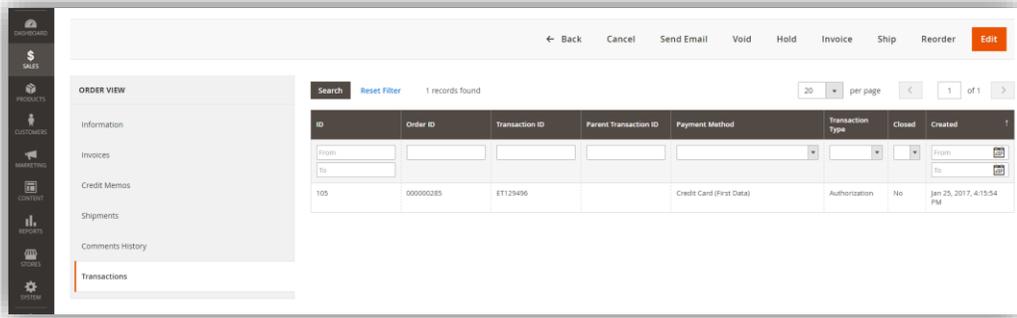
Cardholder Address

* Street

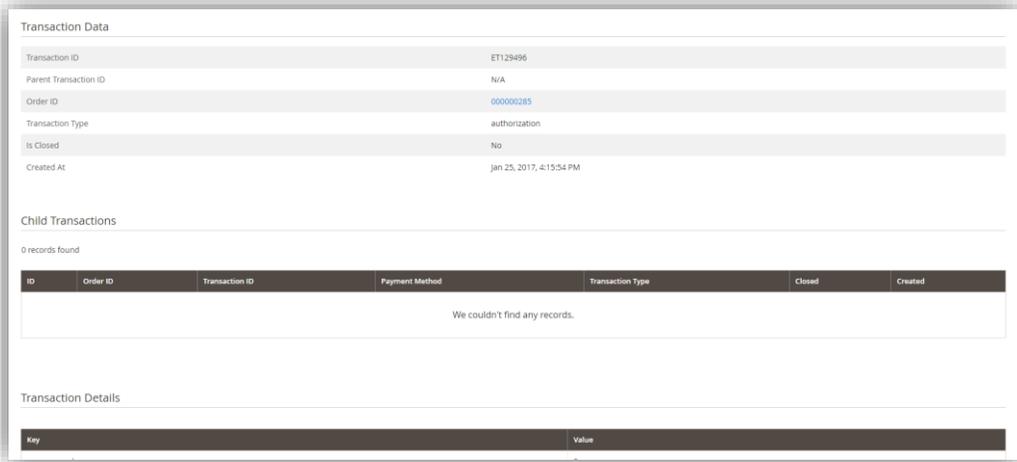
Street Address 2

Admin transaction info

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



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



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."

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.

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

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.

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": "f7d085165acdffa0ea6a0b...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": "f7d085165acdffa0ea6a0b...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
```

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