# BTCPay Greenfield API (v1)

Download OpenAPI specification:

Download

URL: https://btcpayserver.org | License: MIT

BTCPay Greenfield Plugins API

## Introduction

The BTCPay Server Greenfield API is a REST API. Our API has predictable resource-oriented URLs, accepts form-encoded request bodies, returns JSON-encoded responses, and uses standard HTTP response codes, authentication, and verbs.

## Authentication

You can authenticate either via Basic Auth or an API key. It's recommended to use an API key for better security. You can create an API key in the BTCPay Server UI under Account -> Manage Account -> API keys. You can restrict the API key for one or multiple stores and for specific permissions. For testing purposes, you can give it the 'Unrestricted access' permission. On production you should limit the permissions to the actual endpoints you use, you can see the required permission on the API docs at the top of each endpoint under AUTHORIZATIONS.

If you want to simplify the process of creating API keys for your users, you can use the Authorization endpoint to predefine permissions and redirect your users to the BTCPay Server Authorization UI. You can find more information about this on the API Authorization Flow docs page.

# API\_Key

BTCPay Server supports authenticating and authorizing users through an API Key that is generated by them. Send the API Key as a header value to Authorization with the format: <a href="token">token {token}</a>. For a smoother experience, you can generate a url that redirects users to an API key creation screen.

The following permissions are available to the context of the user creating the API Key:

- unrestricted : Unrestricted access
- btcpay.user.candeleteuser : Delete user
- btcpay.user.canviewprofile : View your profile
- btcpay.user.canmodifyprofile : Manage your profile
- btcpay.user.canmanagenotificationsforuser: Manage your notifications
- btcpay.user.canviewnotificationsforuser : View your notifications

The following permissions are available if the user is an administrator:

- btcpay.server.canviewusers: View users
- btcpay.server.cancreateuser : Create new users
- btcpay.server.canmanageusers : Manage users
- btcpay.server.canmodifyserversettings : Manage your server
- btcpay.server.canuseinternallightningnode
   Use the internal lightning node
- <a href="btcpay.server.canviewlightninginvoiceinternalnode">btcpay.server.canviewlightninginvoiceinternalnode</a> : View invoices from internal lightning node
- <a href="btcpay.server.cancreatelightninginvoiceinternalnode">btcpay.server.cancreatelightninginvoiceinternalnode</a> : Create invoices with internal lightning node

The following permissions applies to all stores of the user, you can limit to a specific store with the following format:

btcpay.store.cancreateinvoice:6HSHAEU4iYWtjxtyRs9KyPjM9GAQp8kw2T9VWbGG1FnZ:

- btcpay.store.canmodifystoresettings: Modify your stores
- btcpay.store.webhooks.canmodifywebhooks : Modify stores webhooks
- btcpay.store.canviewstoresettings : View your stores
- btcpay.store.canviewreports: View your reports
- btcpay.store.cancreateinvoice : Create an invoice
- btcpay.store.canviewinvoices : View invoices
- btcpay.store.canmodifyinvoices : Modify invoices
- btcpay.store.canmodifypaymentrequests : Modify your payment requests

- btcpay.store.canviewpaymentrequests : View your payment requests
- btcpay.store.canviewpullpayments : View your pull payments
- btcpay.store.canmanagepullpayments : Manage your pull payments
- btcpay.store.canarchivepullpayments : Archive your pull payments
- btcpay.store.cancreatepullpayments : Create pull payments
- btcpay.store.canmanagepayouts : Manage payouts
- btcpay.store.canviewpayouts: View payouts
- btcpay.store.cancreatenonapprovedpullpayments: Create non-approved pull payments
- <a href="btcpay.store.canuselightningnode">btcpay.store.canuselightningnode</a> : Use the lightning nodes associated with your stores
- <a href="btcpay.store.canviewlightninginvoice">btcpay.store.canviewlightninginvoice</a> : View the lightning invoices associated with your stores
- <a href="btcpay.store.cancreatelightninginvoice">btcpay.store.cancreatelightninginvoice</a> : Create invoices from the lightning nodes associated with your stores

Note that API Keys only limits permission of a user and can never expand it. If an API Key has the permission <a href="btcpay.server.canmodifyserversettings">btcpay.server.canmodifyserversettings</a> but that the user account creating this API Key is not administrator, the API Key will not be able to modify the server settings. Some permissions may include other permissions, see this operation.

Security Scheme Type	API Key
Header parameter name:	Authorization

## Basic

BTCPay Server supports authenticating and authorizing users through the Basic HTTP authentication scheme. Send the user and password encoded in base64 with the format Basic

{base64(username:password)} . Using this authentication method implicitly provides you with the unrestricted permission

Security Scheme Type	HTTP
HTTP Authorization Scheme	Basic

# Usage examples

Use **Basic Auth** to read store information with cURL:

Use an API key to read store information with cURL:

```
STORE_ID="yourStoreId"

curl -s \
    -H "Content-Type: application/json" \
    -H "Authorization: token $API_KEY" \
    -X GET \
    "$BTCPAY_INSTANCE/api/v1/stores/$STORE_ID"
```

You can find more examples on our docs for different programming languages:

- cURL
- Javascript/Node.Js
- PHP

# **API Keys**

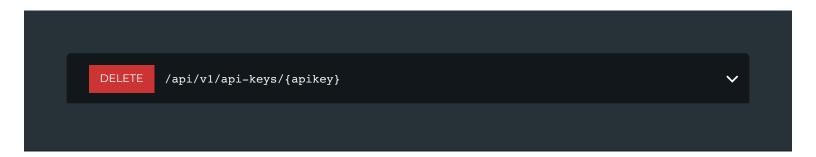
API Key operations

# Revoke an API Key

Revoke the current API key so that it cannot be used anymore

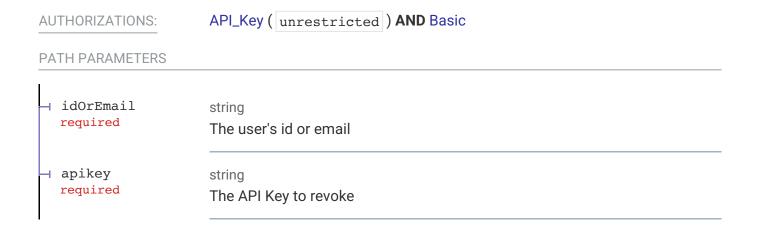
## Responses

- 200 The key has been deleted
- 404 The key is not found for this user



# Revoke an API Key of target user

Revoke the API key of a target user so that it cannot be used anymore



## Responses

```
200 The key has been deleted
404 The key is not found for this user
```



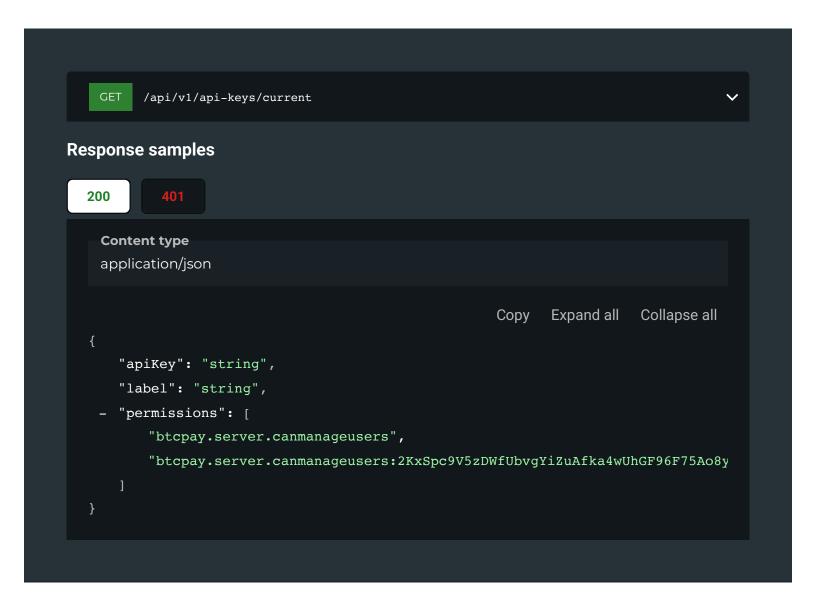
# Get the current API Key information

View information about the current API key

AUTHORIZATIONS: API\_Key ( btcpay.server.canmanageusers ) AND Basic

## Responses

```
> 200 Information about the current api key
> 401 Missing authorization
```



# Revoke the current API Key

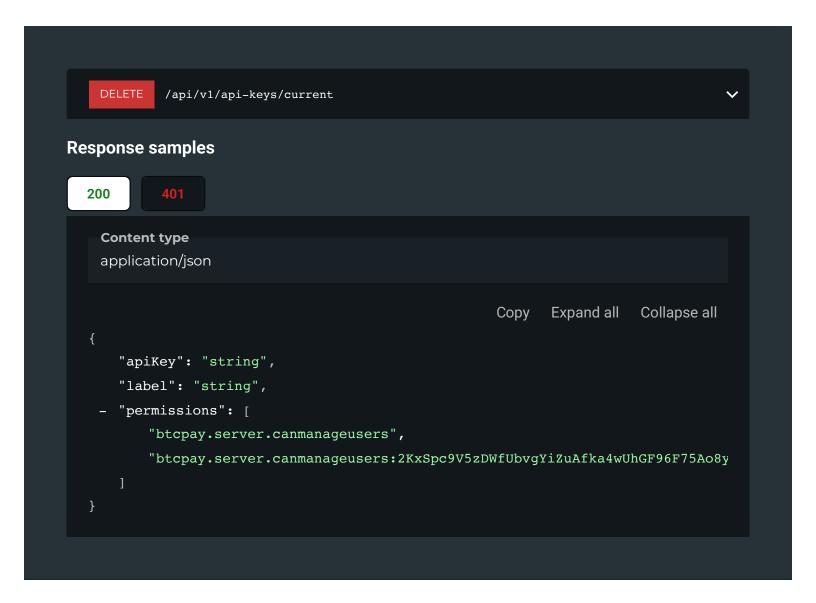
Revoke the current API key so that it cannot be used anymore

AUTHORIZATIONS: API\_Key

## Responses

> 200 The key was revoked and is no longer usable

> 401 Missing authorization



# Create a new API Key

#### Create a new API Key

API\_Key (unrestricted) AND Basic

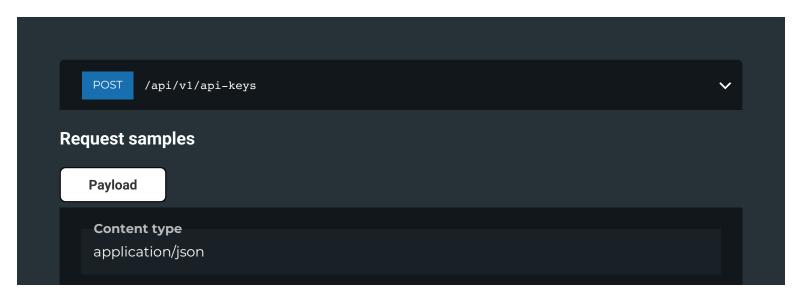
REQUEST BODY SCHEMA: application/json

I label string Nullable
The label of the new API Key

I permissions Array of strings Nullable
The permissions granted to this API Key (See API Key Authentication)

## Responses

200 Information about the new api key
 401 Missing authorization
 default Unexpected error



```
Copy
                                                              Expand all
                                                                          Collapse all
      "label": "string",
     "permissions": [
          "string"
Response samples
  200
    Content type
    application/json
                                                                         Collapse all
                                                       Copy
                                                              Expand all
      "apiKey": "string",
      "label": "string",
    - "permissions": [
          "btcpay.server.canmanageusers",
          "btcpay.server.canmanageusers:2KxSpc9V5zDWfUbvgYiZuAfka4wUhGF96F75Ao8y
```

# Create a new API Key for a user

Create a new API Key for a user

```
AUTHORIZATIONS: API_Key ( btcpay.server.canmanageusers ) AND Basic

PATH PARAMETERS
```

The user's id or email

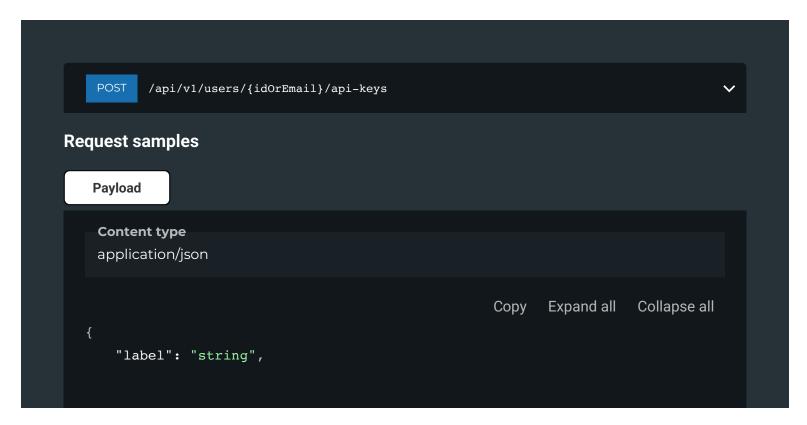
REQUEST BODY SCHEMA: application/json

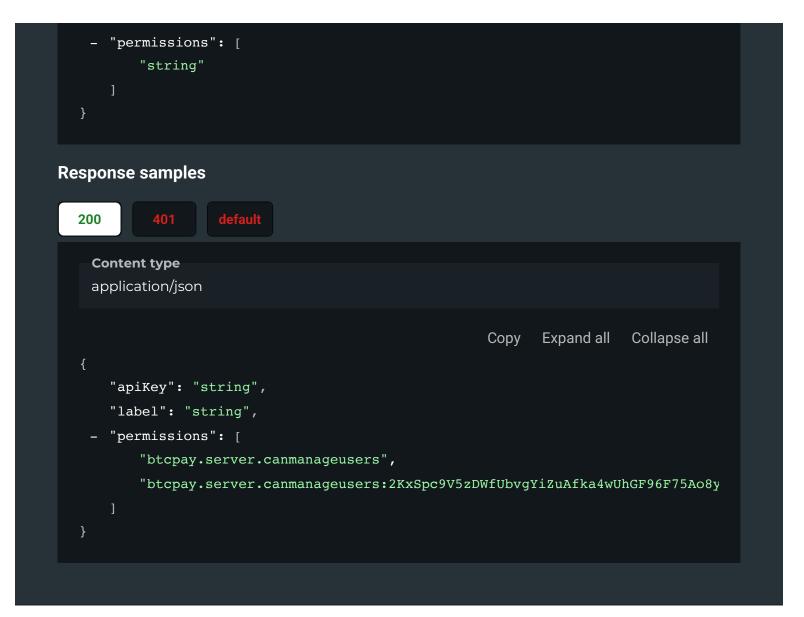
I label string Nullable
The label of the new API Key

Permissions Array of strings Nullable
The permissions granted to this API Key (See API Key Authentication)

## Responses

200 Information about the new api key
 401 Missing authorization
 default Unexpected error





# Apps

App operations

# Create a new Point of Sale app

#### Point of Sale app allows accepting payments for items in a virtual store

API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic **AUTHORIZATIONS:** PATH PARAMETERS → storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json ⊣ id string Id of the app → appName string Name given to the app when it was created ⊣ storeId string Id of the store to which the app belongs → created integer UNIX timestamp for when the app was created → appType string Type of the app which was created → archived boolean Nullable Default: false If true, the app does not appear in the apps list by default. → title string Nullable Display title of the app description string Nullable App description → defaultView string Nullable Enum: "Static" "Cart" "Light" "Print" App view type (e.g., static, cart, etc...)

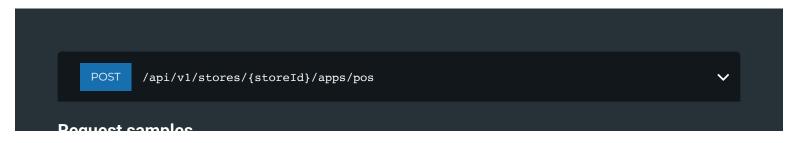
- 1		
L	showItems	boolean Nullable
		Default: false
		Display item selection for keypad
$\vdash$	showCustomAmount	boolean Nullable
		Whether the option to enter a custom amount is shown
L	showDiscount	boolean Nullable
		Default: false
		Whether the option to enter a discount is shown
		whether the option to enter a discount is shown
H	showSearch	boolean Nullable
		Default: true
		Display the search bar
	ah angal a mani a	
Γ	showCategories	boolean Nullable
		Default: true
		Display the list of categories
H	enableTips	boolean Nullable
		Default: false
		Whether the option to enter a tip is shown
١.	aurronau	atring Mullahla
Γ	currency	string Nullable
		Currency used for the app
L	fixedAmountPayButtonText	string Nullable
		Payment button text template for items with a set price
H	customAmountPayButtonText	string Nullable
		Payment button text which appears for items which allow user to
		input a custom amount
L	tipText	string Nullable
	-	Prompt which appears next to the tip amount field if tipping is
		enabled
L	customTipPercentages	Array of numbers Nullable
1.	1	Default: [15,18,20]
		Delault. [13,10,20]

Array of predefined tip percentage amounts → notificationUrl string Nullable Callback notification url to POST to once when invoice is paid for and once when there are enough blockchain confirmations → redirectUrl string Nullable URL user is redirected to once invoice is paid → redirectAutomatically boolean Nullable Whether user is redirected to specified redirect URL automatically after the invoice is paid htmlLang string Nullable Used for SEO, the HTML Lang of the page htmlMetaTags string Nullable Used for SEO, the Meta tags of the page → formId string Nullable Form ID to request customer data template string JSON of item available in the app

## Responses

> 200 Created app details

> 422 Unable to validate the request



#### request samples

#### **Payload**

```
Content type
application/json
                                                 Copy
                                                        Expand all Collapse all
  "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s",
  "appName": "my test app",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "created": 1651554744,
  "appType": "PointOfSale",
  "archived": false,
  "title": "My PoS app",
  "description": "This is my amazing PoS app",
  "defaultView": "Cart",
  "showItems": true,
  "showCustomAmount": true,
  "showDiscount": false,
  "showSearch": false,
  "showCategories": false,
  "enableTips": true,
  "currency": "BTC",
  "fixedAmountPayButtonText": "Buy for {0}",
  "customAmountPayButtonText": "Pay",
  "tipText": "Do you want to leave a tip?",
- "customTipPercentages": [
     15,
     18,
  ],
  "notificationUrl": "string",
  "redirectUrl": "string",
  "redirectAutomatically": true,
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string"
```

```
"template": "string"
}
```

### **Response samples**

200

422

```
Content type
application/json
                                                 Copy Expand all Collapse all
  "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s",
  "appName": "my test app",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "created": 1651554744,
  "appType": "PointOfSale",
  "archived": false,
  "title": "My PoS app",
  "description": "This is my amazing PoS app",
  "defaultView": "Cart",
  "showItems": true,
  "showCustomAmount": true,
  "showDiscount": false,
  "showSearch": false,
  "showCategories": false,
  "enableTips": true,
  "currency": "BTC",
  "fixedAmountPayButtonText": "Buy for \{0\}",
  "customAmountPayButtonText": "Pay",
  "tipText": "Do you want to leave a tip?",
- "customTipPercentages": [
      15,
      18,
  ],
```

# Update a Point of Sale app

Use this endpoint for updating the properties of a POS app

```
AUTHORIZATIONS:
                       API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS
→ appId
                       string
   required
                       App ID
REQUEST BODY SCHEMA: application/json
⊣ id
                                 string
                                 Id of the app
→ appName
                                 Name given to the app when it was created
⊣ storeId
                                 string
                                 Id of the store to which the app belongs
```

⊣ created	integer UNIX timestamp for when the app was created
⊣ аррТуре	string Type of the app which was created
⊣ archived	boolean Nullable Default: false If true, the app does not appear in the apps list by default.
⊣ title	string Nullable Display title of the app
→ description	string Nullable App description
⊣ defaultView	string Nullable Enum: "Static" "Cart" "Light" "Print" App view type (e.g., static, cart, etc)
⊣ showItems	boolean Nullable Default: false Display item selection for keypad
⊣ showCustomAmount	boolean Nullable Whether the option to enter a custom amount is shown
⊣ showDiscount	boolean Nullable  Default: false  Whether the option to enter a discount is shown
⊣ showSearch	boolean Nullable  Default: true  Display the search bar
— showCategories	boolean Nullable Default: true Display the list of categories
⊣ enableTips	boolean Nullable

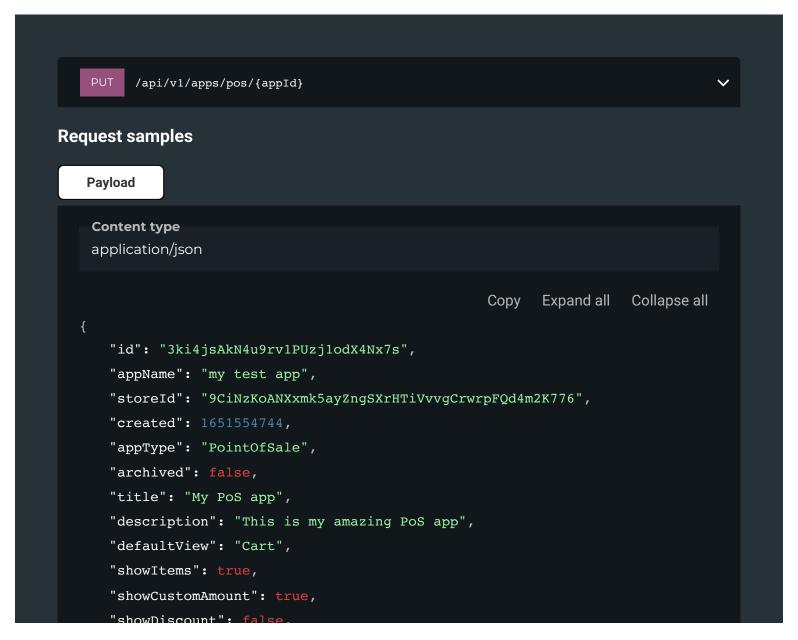
	Default: false Whether the option to enter a tip is shown
→ currency	string Nullable Currency used for the app
→ fixedAmountPayButtonText	string Nullable Payment button text template for items with a set price
→ customAmountPayButtonText	string Nullable Payment button text which appears for items which allow user to input a custom amount
⊣ tipText	string Nullable Prompt which appears next to the tip amount field if tipping is enabled
→ customTipPercentages	Array of numbers Nullable  Default: [15,18,20]  Array of predefined tip percentage amounts
→ notificationUrl	string Nullable Callback notification url to POST to once when invoice is paid for and once when there are enough blockchain confirmations
⊣ redirectUrl	string Nullable URL user is redirected to once invoice is paid
→ redirectAutomatically	boolean Nullable Whether user is redirected to specified redirect URL automatically after the invoice is paid
⊣ htmlLang	string Nullable Used for SEO, the HTML Lang of the page
⊣ htmlMetaTags	string Nullable Used for SEO, the Meta tags of the page
⊣ formId	string Nullable Form ID to request customer data

→ template string

JSON of item available in the app

## Responses

> 200 App details
> 422 Unable to validate the request



```
"showSearch": false,
  "showCategories": false,
  "enableTips": true,
  "currency": "BTC",
  "fixedAmountPayButtonText": "Buy for {0}",
  "customAmountPayButtonText": "Pay",
  "tipText": "Do you want to leave a tip?",
- "customTipPercentages": [
     15,
     18,
  ],
  "notificationUrl": "string",
  "redirectUrl": "string",
  "redirectAutomatically": true,
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string",
  "template": "string"
```

### Response samples

200

422

```
Content type
application/json

Copy Expand all Collapse all

"id": "3ki4jsAkN4u9rv1PUzjlodX4Nx7s",

"appName": "my test app",

"storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",

"created": 1651554744,

"appType": "PointOfSale",

"archived": false,

"title": "My Pos app",
```

```
"description": "This is my amazing PoS app",
  "defaultView": "Cart",
  "showItems": true,
  "showCustomAmount": true,
  "showDiscount": false,
  "showSearch": false,
  "showCategories": false,
  "enableTips": true,
  "currency": "BTC",
  "fixedAmountPayButtonText": "Buy for \{0\}",
  "customAmountPayButtonText": "Pay",
  "tipText": "Do you want to leave a tip?",
- "customTipPercentages": [
     15,
     18,
  ],
  "notificationUrl": "string",
  "redirectUrl": "string",
  "redirectAutomatically": true,
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string",
- "items": [
   + { ... },
```

# Get Point of Sale app data

Returns POS app data

AVITHORIZATIONS:

PATH PARAMETERS

API\_Key AND Basic

PATH PARAMETERS

string
required
App ID

## Responses

```
> 200 POS app data

— 404 POS app with specified ID was not found
```

```
/api/v1/apps/pos/{appId}
Response samples
     200
    Content type
    application/json
                                                                         Collapse all
                                                      Copy
                                                             Expand all
      "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s",
       "appName": "my test app",
      "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
      "created": 1651554744,
       "appType": "PointOfSale",
       "archived": false,
       "title": "My PoS app",
      "description": "This is my amazing PoS app",
```

```
"showItems": true,
  "showCustomAmount": true,
  "showDiscount": false,
  "showSearch": false,
  "showCategories": false,
  "enableTips": true,
  "currency": "BTC",
  "fixedAmountPayButtonText": "Buy for {0}",
  "customAmountPayButtonText": "Pay",
  "tipText": "Do you want to leave a tip?",
- "customTipPercentages": [
     15,
     18,
  ],
  "notificationUrl": "string",
  "redirectUrl": "string",
  "redirectAutomatically": true,
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string",
- "items": [
   + { ... },
   + { ... }
```

# Get crowdfund app data

Returns crowdfund app data

**AUTHORIZATIONS:** 

API\_Key AND Basic

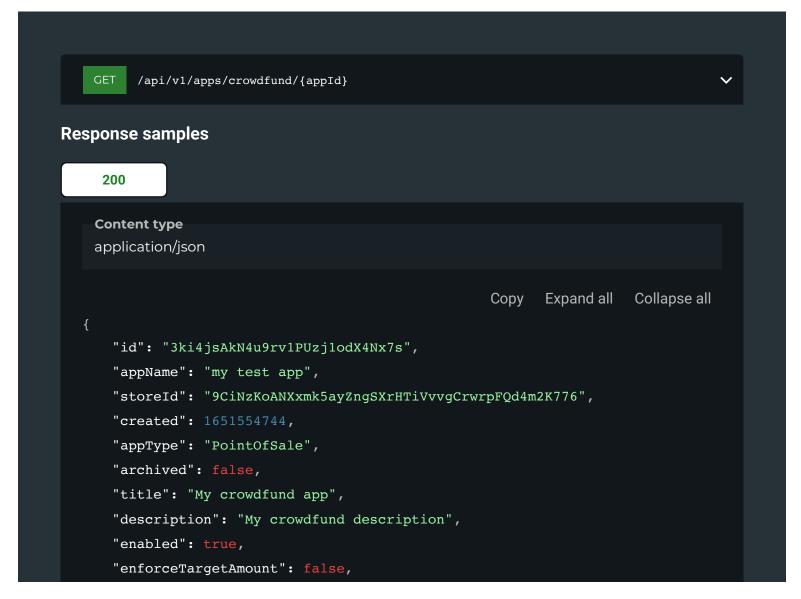
#### PATH PARAMETERS

→ appId string required App ID

### Responses

> 200 Crowdfund app data

— 404 Crowdfund app with specified ID was not found



```
"startDate": 768658369,
  "endDate": 771336769,
  "targetCurrency": "BTC",
  "targetAmount": 420.69,
  "mainImageUrl": "string",
  "notificationUrl": "string",
  "tagline": "I can't believe it's not butter",
  "disqusEnabled": true,
  "disqusShortname": "string",
  "soundsEnabled": false,
  "animationsEnabled": true,
  "resetEveryAmount": 1,
  "resetEvery": "Day",
  "displayPerksValue": false,
  "sortPerksByPopularity": true,
- "sounds": [
      "https://github.com/ClaudiuHKS/AdvancedQuakeSounds/raw/master/sound/AC
  ],
- "animationColors": [
      "#FF0000",
      "#00FF00",
      "#0000FF"
  ],
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string",
- "perks": [
   + { ... },
   + { ... },
```

https://docs.btcpayserver.org/API/Greenfield/v1/

## Create a new Crowdfund app

API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic **AUTHORIZATIONS:** PATH PARAMETERS → storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json ⊣ id string Id of the app → appName string Name given to the app when it was created → storeId string Id of the store to which the app belongs → created integer UNIX timestamp for when the app was created → appType string Type of the app which was created → archived boolean Nullable Default: false If true, the app does not appear in the apps list by default. ⊣ title string Nullable Display title of the app → description string Nullable App description ⊣ enabled boolean Nullable Whether the app is enabled to be viewed by everyone

- enforceTargetAmount	boolean Nullable
	Whether contributions over the set target amount are allowed
⊣ startDate	number <int32> Nullable</int32>
	UNIX timestamp for crowdfund start time
	(https://www.unixtimestamp.com/)
→ endDate	number <int32> Nullable</int32>
	UNIX timestamp for crowdfund end time
	(https://www.unixtimestamp.com/)
→ targetCurrency	string Nullable
	Target currency for the crowdfund
→ targetAmount	number Nullable
	Target amount for the crowdfund
⊣ mainImageUrl	string Nullable
	URL for image used as a cover image for the app
→ notificationUrl	string Nullable
	Callback notification url to POST to once when invoice is paid for and
	once when there are enough blockchain confirmations
→ tagline	string Nullable
	Tagline for the app displayed to user
⊣ disqusEnabled	boolean Nullable
	Whether Disqus is enabled for the app
→ disqusShortname	string Nullable
	Disqus shortname to used for the app
→ soundsEnabled	boolean Nullable
	Whether sounds on new contributions are enabled
→ animationsEnabled	boolean Nullable
	Whether background animations on new contributions are enabled
⊢ resetEveryAmount	number Nullable

Contribution goal reset frequency amount resetEvery string Nullable Contribution goal reset frequency → displayPerksValue boolean Nullable Whether perk values are displayed → sortPerksByPopularity boolean Nullable Default: true Whether perks are sorted by popularity ⊣ sounds Array of strings Nullable Array of custom sounds which can be used on new contributions → animationColors Array of strings Nullable Array of custom HEX colors which can be used for background animations on new contributions htmlLang string Nullable Used for SEO, the HTML Lang of the page htmlMetaTags string Nullable Used for SEO, the Meta tags of the page ⊣ formId string Nullable Form ID to request customer data → perksTemplate JSON of perks available in the app

## Responses

> 200 Created app details

> 422 Unable to validate the request

POST /api/v1/stores/{storeId}/apps/crowdfund

### **Request samples**

#### **Payload**

```
Content type
application/json
                                                        Expand all Collapse all
                                                 Copy
  "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s",
  "appName": "my test app",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "created": 1651554744,
  "appType": "PointOfSale",
  "archived": false,
  "title": "My crowdfund app",
  "description": "My crowdfund description",
  "enabled": true,
  "enforceTargetAmount": false,
  "startDate": 768658369,
  "endDate": 771336769,
  "targetCurrency": "BTC",
  "targetAmount": 420.69,
  "mainImageUrl": "string",
  "notificationUrl": "string",
  "tagline": "I can't believe it's not butter",
  "disqusEnabled": true,
  "disqusShortname": "string",
  "soundsEnabled": false,
  "animationsEnabled": true,
  "resetEveryAmount": 1,
  "resetEvery": "Day",
```

```
"displayPerksValue": false,
    "sortPerksByPopularity": true,

- "sounds": [
        "https://github.com/ClaudiuHKS/AdvancedQuakeSounds/raw/master/sound/AC
],

- "animationColors": [
        "#FF0000",
        "#00FF00",
        "#000FF"

],
    "htmlLang": "en",
    "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">'
    "formId": "string",
    "perksTemplate": "string"
}
```

### Response samples

200

422

```
Content type
application/json
                                                        Expand all Collapse all
                                                  Copy
  "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s",
  "appName": "my test app",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "created": 1651554744,
  "appType": "PointOfSale",
  "archived": false,
  "title": "My crowdfund app",
  "description": "My crowdfund description",
  "enabled": true,
  "enforceTargetAmount": false,
  "startDate": 768658369,
  "endDate": 771336769,
```

```
"targetCurrency": "BTC",
  "targetAmount": 420.69,
  "mainImageUrl": "string",
  "notificationUrl": "string",
  "tagline": "I can't believe it's not butter",
  "disqusEnabled": true,
  "disqusShortname": "string",
  "soundsEnabled": false,
  "animationsEnabled": true,
  "resetEveryAmount": 1,
  "resetEvery": "Day",
  "displayPerksValue": false,
  "sortPerksByPopularity": true,
- "sounds": [
      "https://github.com/ClaudiuHKS/AdvancedQuakeSounds/raw/master/sound/AC
  ],
- "animationColors": [
      "#FF0000",
      "#00FF00",
      "#0000FF"
  ],
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string",
- "perks": [
  + { ... },
   + { ... },
```

## Get basic app data

### Returns basic app data shared between all types of apps

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS	
→ appId required	string App ID

## Responses

```
> 200 Basic app data

— 404 App with specified ID was not found
```

```
Response samples

Content type
application/json

Copy Expand all Collapse all

{
    "id": "3ki4jsAkN4u9rv1PUzjlodX4Nx7s",
    "appName": "my test app",
    "storeId": "9ciNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "created": 1651554744,
    "appType": "PointOfSale",
    "archived": false
```



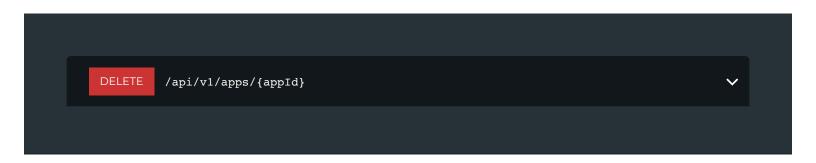
# Delete app

Deletes apps with specified ID

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic	
PATH PARAMETERS		
→ appId required	string App ID	

## Responses

200 App was deleted404 App with specified ID was not found



# Uploads an image for a app item

#### Uploads an image for a app item

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS	
→ appId required	string App ID
REQUEST BODY SCHEMA:	multipart/form-data
⊣ file	string binary> The image

- > 200 Uploads an image for a app item
- 404 The app could not be found



```
"uri": "string",

"url": "string",

"originalName": "string",

"storageName": "string",

"created": 1592312018
}
```

# Deletes the app item image

Deletes the app item image

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings) AND Basic
PATH PARAMETERS	
⊢ appId required	string App ID
⊣ fileId required	string The file ID

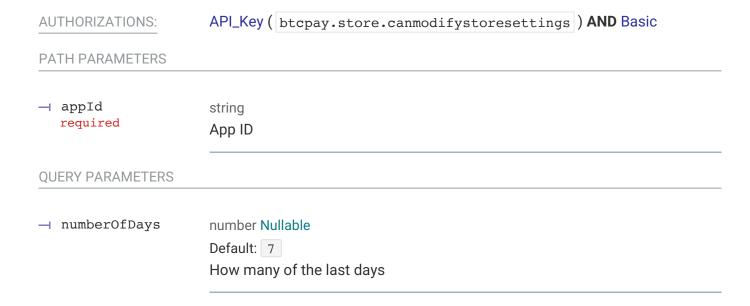
### Responses

```
200 App item image deleted successfully
404 The app could not be found
```



# Get app sales statistics

Returns sales statistics for the app



#### Responses

> 200 App sales statistics

— 404 App with specified ID was not found



```
+ { ... }
]
}
```

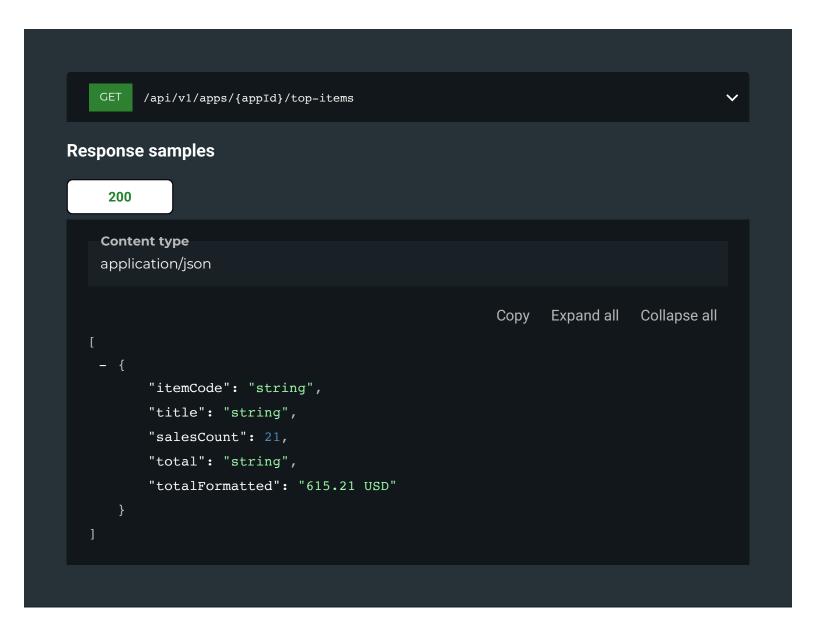
# Get app top items statistics

Returns top items statistics for the app

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS	
→ appId required	string App ID
QUERY PARAMETERS	
- count	number Nullable Default: 5 How many of the items
⊣ offset	number Nullable Default: 0 Offset for paging

```
> 200 App top items statistics

— 404 App with specified ID was not found
```



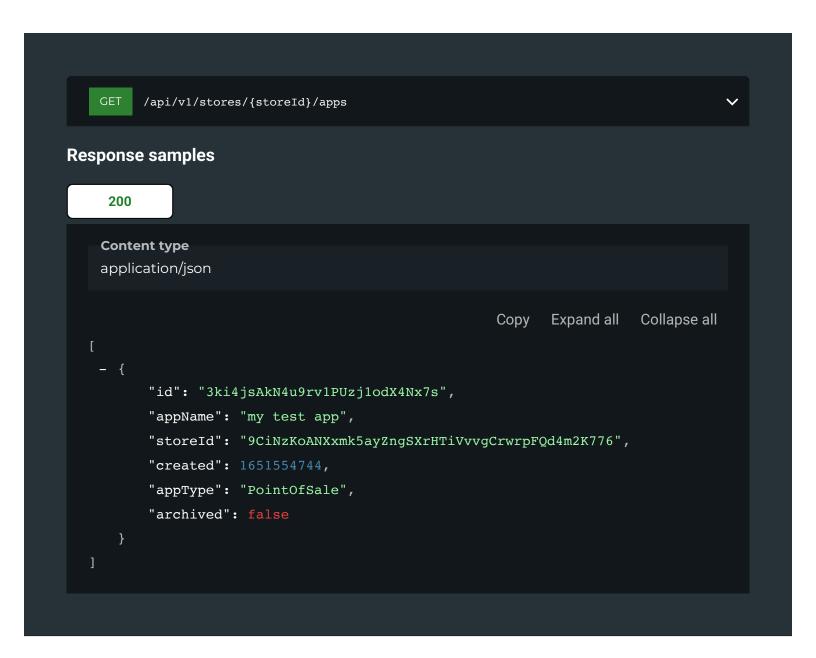
# Get basic app data for all apps for a store

Returns basic app data for all apps for a store

#### Responses

> 200 Array of basic app data object

- 403 If you are authenticated but forbidden to get the data



## Get basic app data for all apps for all stores for a user

Returns basic app data for all apps for all stores

```
AUTHORIZATIONS: API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
```

```
> 200 Array of basic app data object

— 403 If you are authenticated but forbidden to get the data
```

```
/api/v1/apps
Response samples
     200
    Content type
    application/json
                                                                         Collapse all
                                                       Copy
                                                              Expand all
          "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s",
          "appName": "my test app",
          "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
          "created": 1651554744,
          "appType": "PointOfSale",
          "archived": false
```

# Authorization

Authorization operations

### Authorize User

Redirect the browser to this endpoint to request the user to generate an api-key with specific permissions

#### **QUERY PARAMETERS**

H	permissions	Array of strings Nullable The permissions to request. (See API Key authentication)
	strict	boolean Nullable  Default: true  If permissions are specified, and strict is set to false, it will allow the user to reject some of permissions the application is requesting.
	applicationIdentifier	string Nullable If specified, BTCPay Server will check if there is an existing API key associated with the user that also has this application identifier, redirect host AND the permissions required match(takes selectiveStores and strict into account). applicationIdentifier is ignored if redirect is not specified.
H	selectiveStores	boolean Nullable Default: false

If the application is requesting the CanModifyStoreSettings permission and selectiveStores is set to true, this allows the user to only grant permissions to selected stores under the user's control.

I applicationName string Nullable
The name of your application

I redirect string <url>
Nullable
The url to redirect to after the user consents, with the query parameters appended to it: permissions, user-id, api-key. If not specified, user is redirected to their API Key list.

#### Responses

200 A HTML form that a user can use to confirm permissions to grant
 307
 Makes browser do an HTTP POST request to the specified url in redirect with a JSON body consisting of apiKey (the api key created or matched), permissions (the permissions the user consented to), and userId (the id of the user that consented) upon consent

 401 Missing authorization



### **Files**

File operations

### Get all files

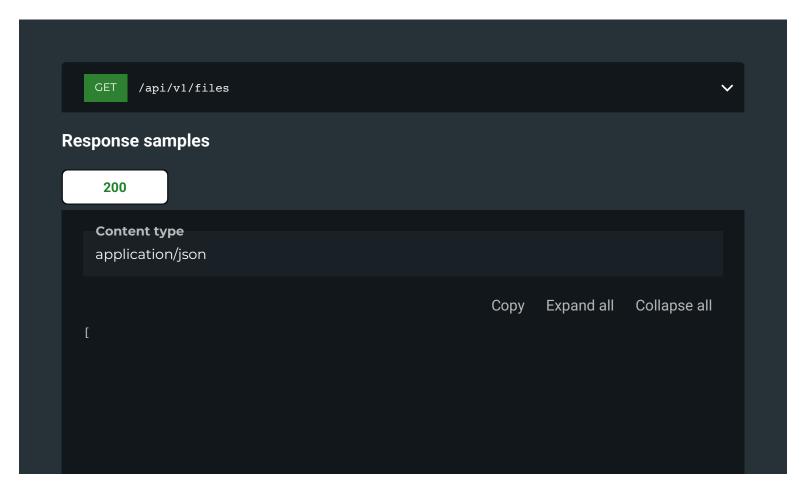
Load all files that exist.

AUTHORIZATIONS: API\_Key ( btcpay.server.canmodifyserversettings ) AND Basic

#### Responses

> 200 Files found

— 401 Missing authorization for loading the files



```
- {
    "id": "string",
    "userId": "string",
    "uri": "string",
    "originalName": "string",
    "storageName": "string",
    "created": 1592312018
  }
]
```

# Uploads a file

Uploads a file

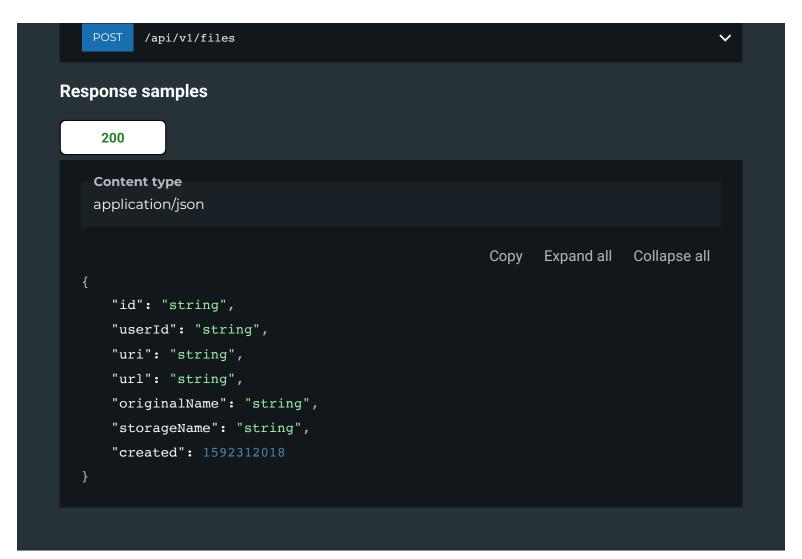
```
API_Key ( btcpay.server.canmodifyserversettings ) AND Basic

REQUEST BODY SCHEMA: multipart/form-data

If ile string <binary>
The profile picture
```

```
> 200 Uploads a file

— 415 The upload did not work
```



### Get file

View information about the specified file

```
API_Key ( btcpay.server.canmodifyserversettings ) AND Basic

PATH PARAMETERS

— fileId string The file information to fetch
```

```
> 200 File found

— 401 Missing authorization for loading the file
```

```
/api/v1/files/{fileId}
Response samples
     200
    Content type
    application/json
                                                                          Collapse all
                                                               Expand all
                                                        Copy
      "id": "string",
      "userId": "string",
      "uri": "string",
      "url": "string",
      "originalName": "string",
       "storageName": "string",
       "created": 1592312018
```

# Delete file

Deletes the file

AUTHORIZATIONS: API\_Key ( btcpay.server.canmodifyserversettings ) AND Basic

PATH PARAMETERS

→ fileId string required The fi

The file to delete

#### Responses

- 200 File deleted successfully

- 404 The file could not be found

DELETE /api/v1/files/{fileId}

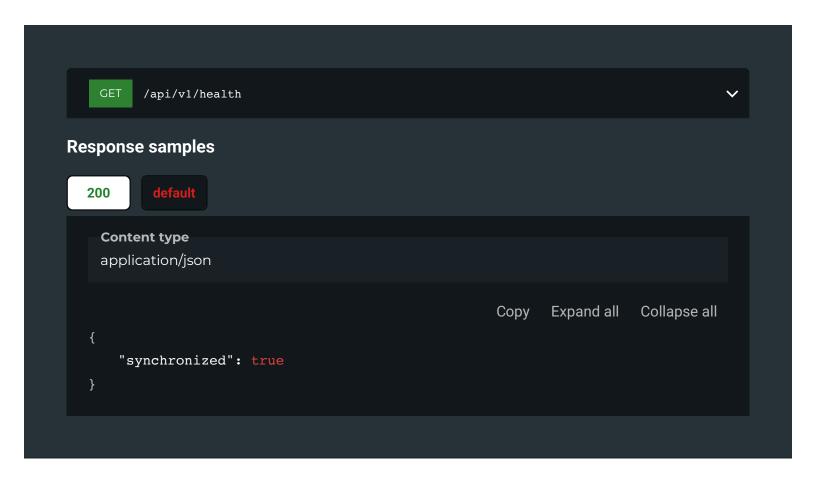
# Health

Health operations

### Get health status

Check the instance health status

```
    200 Instance is up
    403 If you are authenticated but forbidden to get the data
    default Unexpected error
```



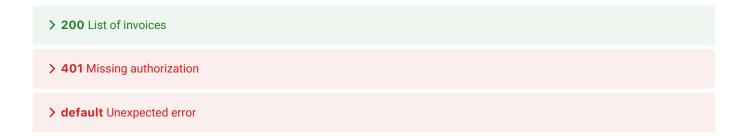
# Invoices

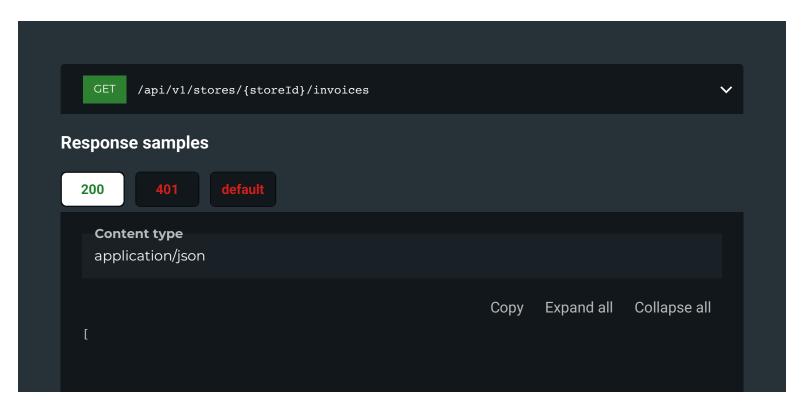
Invoice operations

### Get invoices

View information about the existing invoices

AUTHORIZATIONS:	API_Key( btcpay.store.canviewinvoices ) AND Basic
PATH PARAMETERS	
→ storeId required	string (StoreId) Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID
QUERY PARAMETERS	
⊣ orderId	Array of strings  Example: orderId=1000&orderId=1001&orderId=1002  Array of OrderIds to fetch the invoices for
⊣ textSearch	string A term that can help locating specific invoices.
⊣ status	string (InvoiceStatus)  Enum: "Expired" "Invalid" "New" "Processing" "Settled"  Array of statuses of invoices to be fetched
⊣ endDate	number <int32> (UnixTimestamp)  Example: endDate=1592312018</int32>





```
+ "metadata": { ... },
+ "checkout": { ... },
+ "receipt": { ... },
  "id": "string",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "amount": "5.00",
  "currency": "USD",
  "type": "Standard",
  "checkoutLink": "string",
  "createdTime": 1592312018,
  "expirationTime": 1592312018,
  "monitoringExpiration": 1592312018,
  "status": "Expired",
  "additionalStatus": "Invalid",
+ "availableStatusesForManualMarking": [ ... ],
  "archived": true
```

### Create a new invoice

REQUEST BODY SCHEMA: application/json

Create a new invoice

⊣ metadata > Buyer informations (object) or Product information (object) or Order information (object) or Payment request information (object) or Point of Sale (Cart view) (object) or Receipt information (object) (InvoiceMetadata) Additional information around the invoice that can be supplied. The mentioned properties are all optional and you can introduce any json format you wish. See our documentation for more information. - checkout > object Nullable Additional settings to customize the checkout flow - receipt > object Nullable Additional settings to customize the public receipt ⊣ amount string <decimal> Nullable The amount of the invoice. If null or unspecified, the invoice will be a top-up invoice. (ie. The invoice will consider any payment as a full payment) → currency string Nullable The currency of the invoice (if null, empty or unspecified, the currency will be the store's settings default)' additionalSearchTerms Array of strings Nullable Additional search term to help you find this invoice via text search

#### Responses

> 200 Information about the new invoice

> 400 A list of errors that occurred when creating the invoice

403 If you are authenticated but forbidden to add new invoices

/api/vi/scores/{scorera}/invoices

#### **Request samples**

#### **Payload**

```
Content type
application/json
                                                  Copy
                                                        Expand all Collapse all
- "metadata": {
      "orderId": "pos-app_346KRC5BjXXXo8cRFKwTBmdR6ZJ4",
      "orderUrl": "https://localhost:14142/apps/346KRC5BjXXXo8cRFKwTBmdR6ZJ4
      "itemDesc": "Tea shop",
   + "posData": { ... },
   + "receiptData": { ... }
  },
- "checkout": {
      "speedPolicy": "HighSpeed",
   + "paymentMethods": [ ... ],
      "defaultPaymentMethod": "BTC-CHAIN",
      "lazyPaymentMethods": true,
      "expirationMinutes": 90,
      "monitoringMinutes": 90,
      "paymentTolerance": 0,
      "redirectURL": "string",
      "redirectAutomatically": true,
      "defaultLanguage": "string"
  },
- "receipt": {
      "enabled": true,
      "showQR": null,
      "showPayments": null
  },
  "amount": "5.00",
  "currency": "USD",
```

```
- "additionalSearchTerms": [
        "string"
        ]
}
```

#### **Response samples**

200

400

```
Content type
application/json
                                                         Expand all Collapse all
                                                  Copy
- "metadata": {
      "orderId": "pos-app 346KRC5BjXXXo8cRFKwTBmdR6ZJ4",
      "orderUrl": "https://localhost:14142/apps/346KRC5BjXXXo8cRFKwTBmdR6ZJ4
      "itemDesc": "Tea shop",
   + "posData": { ... },
   + "receiptData": { ... }
  },
- "checkout": {
      "speedPolicy": "HighSpeed",
   + "paymentMethods": [ ... ],
      "defaultPaymentMethod": "BTC-CHAIN",
      "lazyPaymentMethods": true,
      "expirationMinutes": 90,
      "monitoringMinutes": 90,
      "paymentTolerance": 0,
      "redirectURL": "string",
      "redirectAutomatically": true,
      "defaultLanguage": "string"
  },
```

```
- "receipt": {
      "enabled": true,
     "showQR": null,
      "showPayments": null
  },
  "id": "string",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "amount": "5.00",
  "currency": "USD",
  "type": "Standard",
  "checkoutLink": "string",
  "createdTime": 1592312018,
  "expirationTime": 1592312018,
  "monitoringExpiration": 1592312018,
  "status": "Expired",
  "additionalStatus": "Invalid",
- "availableStatusesForManualMarking": [
      "Expired"
  ],
  "archived": true
```

### Get invoice

View information about the specified invoice

```
API_Key ( btcpay.store.canviewinvoices ) AND Basic

PATH PARAMETERS

- invoiceId string required
```

The invoice ID

storeId string (StoreId)

required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

```
    200 specified invoice
    403 If you are authenticated but forbidden to view the specified invoice
    404 The key is not found for this invoice
```

```
GET
          /api/v1/stores/{storeId}/invoices/{invoiceId}
Response samples
     200
    Content type
    application/json
                                                                           Collapse all
                                                               Expand all
                                                        Copy
      "metadata": {
           "orderId": "pos-app_346KRC5BjXXXo8cRFKwTBmdR6ZJ4",
          "orderUrl": "https://localhost:14142/apps/346KRC5BjXXXo8cRFKwTBmdR6ZJ4
          "itemDesc": "Tea shop",
          "posData": { ... },
          "receiptData": { ... }
      },
```

```
- "checkout": {
      "speedPolicy": "HighSpeed",
   + "paymentMethods": [ ... ],
      "defaultPaymentMethod": "BTC-CHAIN",
      "lazyPaymentMethods": true,
      "expirationMinutes": 90,
      "monitoringMinutes": 90,
      "paymentTolerance": 0,
      "redirectURL": "string",
      "redirectAutomatically": true,
      "defaultLanguage": "string"
  },
- "receipt": {
      "enabled": true,
      "showQR": null,
      "showPayments": null
  },
  "id": "string",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "amount": "5.00",
  "currency": "USD",
  "type": "Standard",
  "checkoutLink": "string",
  "createdTime": 1592312018,
  "expirationTime": 1592312018,
  "monitoringExpiration": 1592312018,
  "status": "Expired",
  "additionalStatus": "Invalid",
- "availableStatusesForManualMarking": [
      "Expired"
  ],
  "archived": true
```

#### Archive invoice

Archives the specified invoice.

### Responses

200 The invoice has been archived
400 A list of errors that occurred when archiving the invoice
403 If you are authenticated but forbidden to archive the specified invoice
404 The key is not found for this invoice



```
Copy Expand all Collapse all

[
- {
    "path": "string",
    "message": "string"
    }
]
```

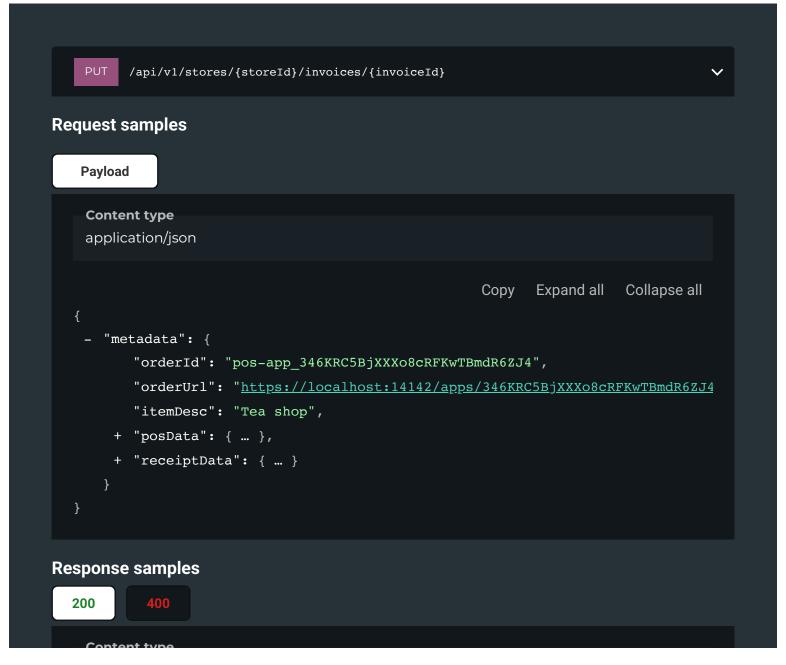
## Update invoice

Updates the specified invoice.

```
AUTHORIZATIONS:
                         API_Key (| btcpay.store.canmodifyinvoices |) AND Basic
PATH PARAMETERS
   invoiceId
                         string
   required
                         The invoice ID
  storeId
                         string (StoreId)
   required
                         Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                         The store ID
REQUEST BODY SCHEMA:
                          application/json

→ metadata >
                         Buyer informations (object) or Product information (object) or Order information
                         (object) or Payment request information (object) or Point of Sale (Cart view)
                         (object) or Receipt information (object) (InvoiceMetadata)
                         Additional information around the invoice that can be supplied. The
                         mentioned properties are all optional and you can introduce any json format
                         you wish. See our documentation for more information.
```

```
    200 The invoice that has been updated
    400 A list of errors that occurred when updating the invoice
    403 If you are authenticated but forbidden to update the specified invoice
    404 The key is not found for this invoice
```



application/json Expand all Collapse all Copy - "metadata": { "orderId": "pos-app\_346KRC5BjXXXo8cRFKwTBmdR6ZJ4", "orderUrl": "https://localhost:14142/apps/346KRC5BjXXXo8cRFKwTBmdR6ZJ4 "itemDesc": "Tea shop", + "posData": { ... }, + "receiptData": { ... } }, - "checkout": { "speedPolicy": "HighSpeed", + "paymentMethods": [ ... ], "defaultPaymentMethod": "BTC-CHAIN", "lazyPaymentMethods": true, "expirationMinutes": 90, "monitoringMinutes": 90, "paymentTolerance": 0, "redirectURL": "string", "redirectAutomatically": true, "defaultLanguage": "string" }, - "receipt": { "enabled": true, "showQR": null, "showPayments": null }, "id": "string", "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776", "amount": "5.00", "currency": "USD", "type": "Standard", "checkoutLink": "string", "createdTime": 1592312018, "expirationTime": 1592312018,

## Get invoice payment methods

View information about the specified invoice's payment methods

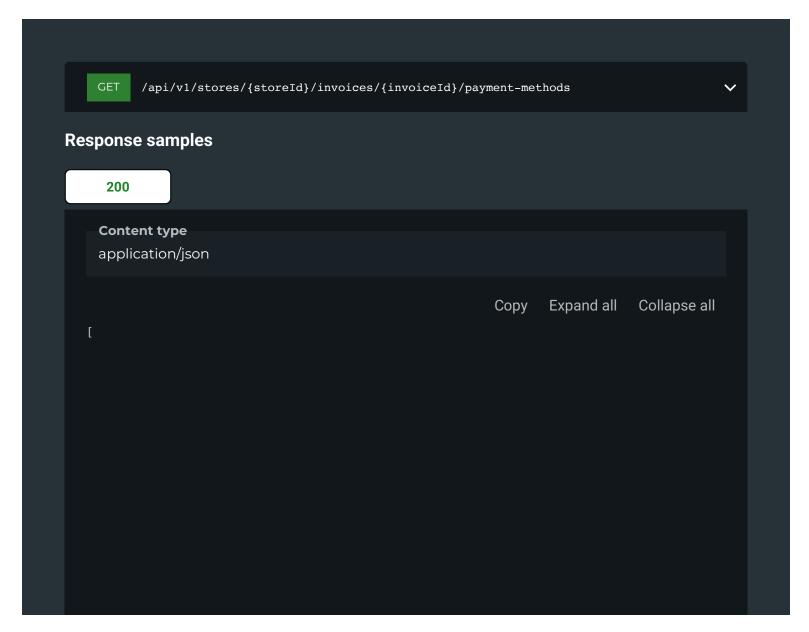
```
API_Key ( btcpay.store.canviewinvoices ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
⊣ invoiceId
                       string
  required
                       The invoice ID

→ storeId

                       string (StoreId)
  required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
OUERY PARAMETERS
→ includeSensitive
                            boolean
                            Default: false
                            If true, additionalData might include sensitive data (such as
                            xpub). Requires the permission
                            btcpay.store.canmodifystoresettings.
→ onlyAccountedPayments boolean
                            Default: true
                            If default or true, only returns payments which are accounted (in
```

Bitcoin, this mean not returning RBF'd or double spent payments)

- **> 200** specified invoice payment methods data
- 403 If you are authenticated but forbidden to view the specified invoice
- 404 The key is not found for this invoice



```
- {
    "paymentMethodId": "BTC-CHAIN",
    "currency": "BTC",
    "destination": "string",
    "paymentLink": "string",
    "rate": "64392.23",
    "paymentMethodPaid": "string",
    "totalPaid": "string",
    "due": "string",
    "amount": "string",
    "paymentMethodFee": "string",
    "paymentMethodFee": "string",
    "paymentMethodFee": "string",
    + "payments": [ ... ],
    "activated": true,
    + "additionalData": { ... }
}
```

### Mark invoice status

Mark an invoice as invalid or settled.

```
API_Key ( btcpay.store.canmodifyinvoices ) AND Basic

PATH PARAMETERS

H invoiceId string The invoice ID

H storeId string (StoreId) string (StoreId) Example: 9CinzkoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID

REQUEST BODY SCHEMA: application/json
```

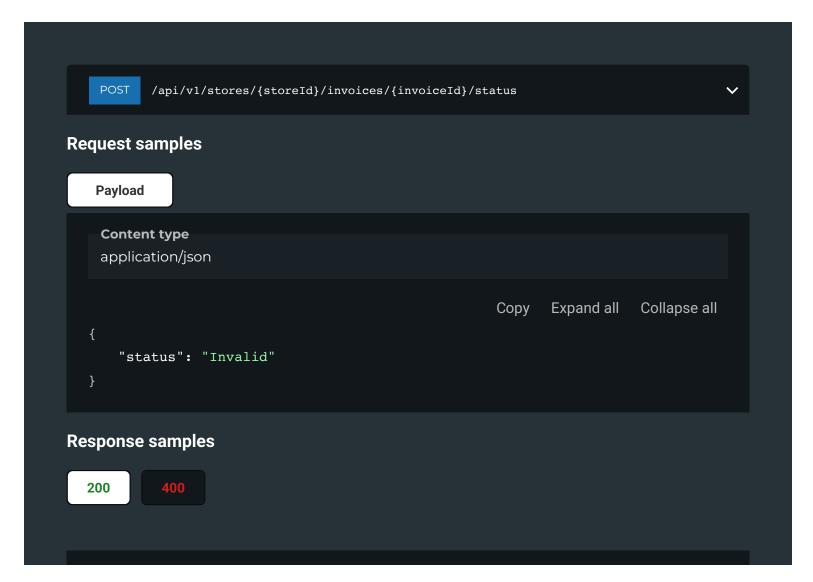
— status string

Enum: "Invalid" "Settled"

Mark an invoice as completed or invalid.

#### Responses

200 The updated invoice
 400 A list of errors that occurred when updating the invoice
 403 If you are authenticated but forbidden to update the invoice



# **Content type** application/json Copy Expand all Collapse all - "metadata": { "orderId": "pos-app 346KRC5BjXXXo8cRFKwTBmdR6ZJ4", "orderUrl": "https://localhost:14142/apps/346KRC5BjXXXo8cRFKwTBmdR6ZJ4 "itemDesc": "Tea shop", + "posData": { ... }, + "receiptData": { ... } }, - "checkout": { "speedPolicy": "HighSpeed", + "paymentMethods": [ ... ], "defaultPaymentMethod": "BTC-CHAIN", "lazyPaymentMethods": true, "expirationMinutes": 90, "monitoringMinutes": 90, "paymentTolerance": 0, "redirectURL": "string", "redirectAutomatically": true, "defaultLanguage": "string" }, - "receipt": { "enabled": true, "showQR": null, "showPayments": null }, "id": "string", "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776", "amount": "5.00", "currency": "USD", "type": "Standard", "checkoutLink": "string", "createdTime": 1592312018, "expirationTime": 1592312018,

#### Unarchive invoice

Unarchive an invoice

AUTHORIZATIONS: API\_Key ( btcpay.store.canmodifyinvoices ) AND Basic

#### PATH PARAMETERS

```
- invoiceId string
required The invoice ID

- storeId string (StoreId)
required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID
```

- > 200 The unarchived invoice
- > 400 A list of errors that occurred when updating the invoice
- 403 If you are authenticated but forbidden to update the invoice

POST /api/v1/stores/{storeId}/invoices/{invoiceId}/unarchive **Response samples** 200 **Content type** application/json Copy Expand all Collapse all - "metadata": { "orderId": "pos-app\_346KRC5BjXXXo8cRFKwTBmdR6ZJ4", "orderUrl": "https://localhost:14142/apps/346KRC5BjXXXo8cRFKwTBmdR6ZJ4 "itemDesc": "Tea shop", + "posData": { ... }, + "receiptData": { ... } }, - "checkout": { "speedPolicy": "HighSpeed", + "paymentMethods": [ ... ], "defaultPaymentMethod": "BTC-CHAIN", "lazyPaymentMethods": true, "expirationMinutes": 90, "monitoringMinutes": 90, "paymentTolerance": 0, "redirectURL": "string", "redirectAutomatically": true, "defaultLanguage": "string" },

```
receipt": {
      "enabled": true,
     "showQR": null,
      "showPayments": null
  },
  "id": "string",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "amount": "5.00",
  "currency": "USD",
  "type": "Standard",
  "checkoutLink": "string",
  "createdTime": 1592312018,
  "expirationTime": 1592312018,
  "monitoringExpiration": 1592312018,
  "status": "Expired",
  "additionalStatus": "Invalid",
- "availableStatusesForManualMarking": [
      "Expired"
  ],
  "archived": true
```

## Activate Payment Method

Activate an invoice payment method (if lazy payments mode is enabled)

```
required

Example: BTC-CHAIN

The payment method id of the payment method to update

storeId

required

string (StoreId)

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID
```

```
    200
    400 A list of errors that occurred when updating the invoice
    403 If you are authenticated but forbidden to activate the invoice payment method
```

## Refund invoice

#### Refund invoice

AUTHORIZATIONS:	API_Key( btcpay.store.cancreatepullpayments ) AND Basic
PATH PARAMETERS	
→ invoiceId required	string The invoice ID
⊣ storeId required	string (StoreId) Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID
REQUEST BODY SCHEMA:	application/json
— name	string Nullable  Name of the pull payment (Default: 'Refund' followed by the invoice id)
→ description	string Description of the pull payment
⊣ payoutMethodId	string (PayoutMethodId) Payout method IDs. Available payment method IDs for Bitcoin are:  • "BTC-CHAIN": Onchain • "BTC-LN": Lightning
⊣ refundVariant	<pre>string Enum: "CurrentRate" "Custom" "Fiat" "OverpaidAmount"  "RateThen"  • RateThen: Refund the crypto currency price, at the rate the invoice got paid.  • CurrentRate: Refund the crypto currency price, at the current rate.</pre>

Fiat: Refund the invoice currency, at the rate when the refund will be sent.

 OverpaidAmount: Refund the crypto currency amount that was overpaid.

 Custom: Specify the amount, currency, and rate of the refund. (see customAmount and customCurrency)

subtractPercentage string <decimal>
Optional percentage by which to reduce the refund, e.g. as processing charge or to compensate for the mining fee.

customAmount string <decimal>
The amount to refund if the refundVariant is Custom.

customCurrency string
The currency to refund if the refundVariant is Custom

#### Responses

200 Pull payment for refunding the invoice
 400 A list of errors that occurred when refunding the invoice
 403 If you are authenticated but forbidden to refund the invoice



```
Content type
application/json

Copy Expand all Collapse all

{
    "name": "string",
    "description": "string",
    "payoutMethodId": "BTC-LN",
    "refundVariant": "CurrentRate",
    "subtractPercentage": "2.1",
    "customAmount": "5.00",
    "customCurrency": "USD"
}
```

#### **Response samples**

```
Content type
application/json

Copy Expand all Collapse all

"id": "string",
   "name": "string",
   "description": "string",
   "currency": "BTC",
   "amount": "1.12000000",
   "BOLT11Expiration": 30,
   "autoApproveClaims": false,
   "archived": true,
   "viewLink": "string"

}
```

# Lightning (Internal Node)

Lightning (Internal Node) operations

## Get node information

View information about the lightning node

## Responses

200 Lightning node information such as reachable nodeinfos
 404 The lightning node configuration was not found
 503 Unable to access the lightning node



# **Response samples** 200 **Content type** application/json Expand all Collapse all Copy - "nodeURIs": [ "string" ], "blockHeight": 0, "alias": "string", "color": "string", "version": "string", "peersCount": 0, "activeChannelsCount": 0, "inactiveChannelsCount": 0, "pendingChannelsCount": 0

## Get node balance

View balance of the lightning node

```
AVITHORIZATIONS:

PATH PARAMETERS

API_Key ( btcpay.server.canuseinternallightningnode ) AND Basic

required string

Example: BTC
```

#### The cryptoCode of the lightning-node to query

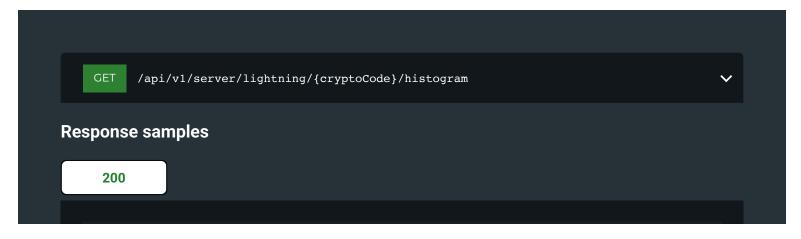
- 200 Lightning node balance for on-chain and off-chain funds
   404 The lightning node configuration was not found
   503 Unable to access the lightning node
- /api/v1/server/lightning/{cryptoCode}/balance Response samples 200 **Content type** application/json Expand all Collapse all Copy - "onchain": { "confirmed": "string", "unconfirmed": "string", "reserved": "string" }, - "offchain": { "opening": "string", "local": "string", "remote": "string", "closing": "string"

## Get node balance histogram

View balance histogram of the lightning node

## Responses

200 Lightning node balance histogram for off-chain funds
 404 The lightning node configuration was not found
 503 Unable to access the lightning node



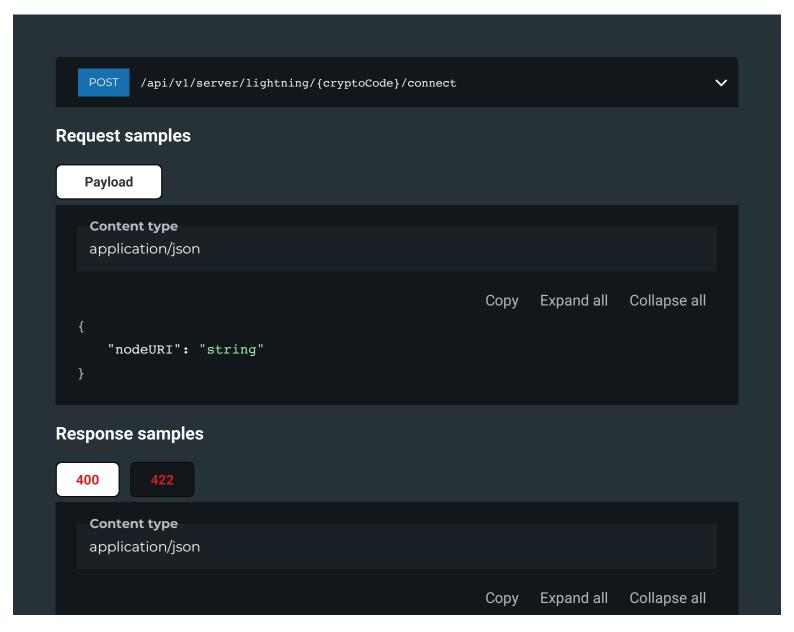
```
Copy Expand all Collapse all

{
    "type": "Week",
    "balance": "string",
    - "series": [
        "string"
    ],
    - "labels": [
        0
    ]
}
```

## Connect to lightning node

Connect to another lightning node.

```
    200 Successfully connected
    400 Wellknown error codes are: could-not-connect
    404 The lightning node configuration was not found
    422 Unable to validate the request
    503 Unable to access the lightning node
```



```
{
    "code": "string",
    "message": "string"
}
```

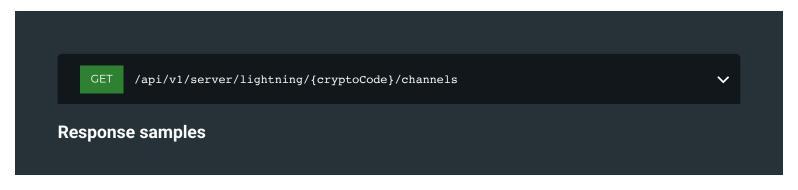
## Get channels

View information about the current channels of the lightning node

## Responses

> 200 list of channels

— 404 The lightning node configuration was not found



```
Content type
application/json

Copy Expand all Collapse all

[
- {
    "remoteNode": "string",
    "isPublic": true,
    "isActive": true,
    "capacity": "string",
    "localBalance": "string",
    "channelPoint": "string"
}

]
```

## Open channel

Open a channel with another lightning node. You should connect to that node first.

```
The amount to fund (in satoshi)

I feeRate number

The amount to fund (in satoshi per byte)
```

```
- 200 Successfully opened

> 400
Wellknown error codes are: channel-already-exists , cannot-afford-funding , need-more-confirmations , peer-not-connected

- 404 The lightning node configuration was not found
> 422 Unable to validate the request
```

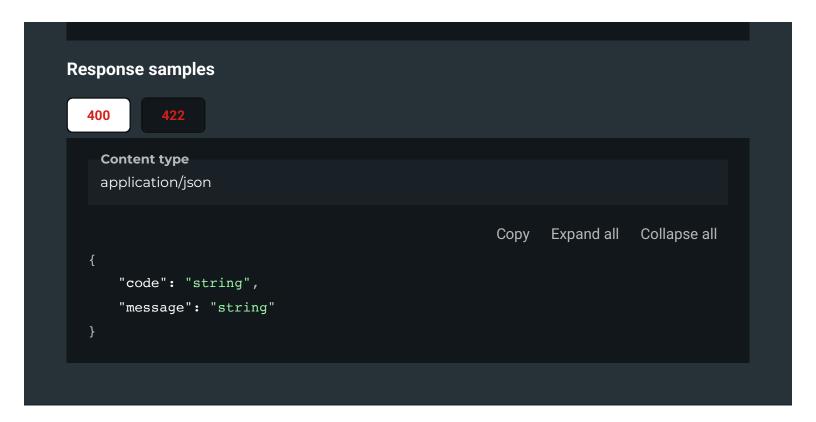
```
POST /api/v1/server/lightning/{cryptoCode}/channels

Request samples

Payload

Content type application/json

Copy Expand all Collapse all {
    "nodeURI": "string",
    "channelAmount": "string",
    "feeRate": 0
}
```



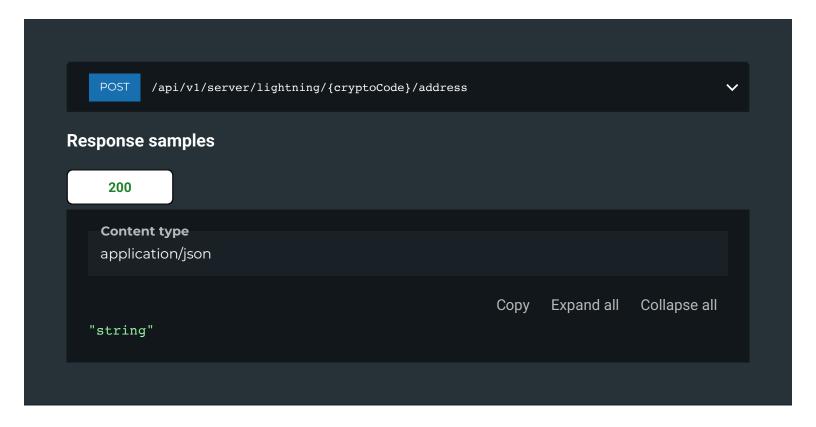
## Get deposit address

Get an on-chain deposit address for the lightning node

```
> 200 deposit address

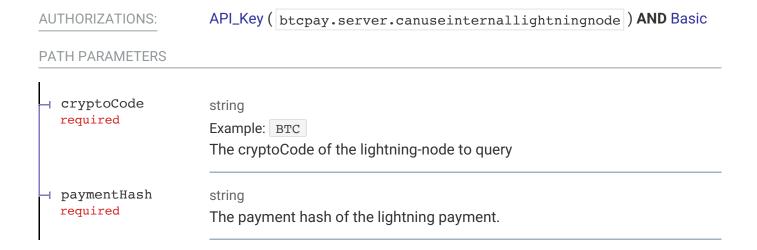
— 404 The lightning node configuration was not found
```

- 503 Unable to access the lightning node



## Get payment

View information about the requested lightning payment



```
    200 Lightning payment data
    404 The lightning node configuration or the specified invoice was not found
    503 Unable to access the lightning node
```

```
/api/v1/server/lightning/{cryptoCode}/payments/{paymentHash}
Response samples
     200
    Content type
    application/json
                                                               Expand all
                                                                           Collapse all
                                                        Copy
      "id": "string",
      "status": "Complete",
      "BOLT11": "string",
      "paymentHash": "string",
       "preimage": "string",
       "createdAt": 1592312018,
       "totalAmount": "string",
      "feeAmount": "string"
```

#### UCL IIIVUICC

View information about the requested lightning invoice

AUTHORIZATIONS: API\_Key(|btcpay.server.canviewlightninginvoiceinternalnode|)

**AND** Basic

#### PATH PARAMETERS

required string

Example: BTC

The cryptoCode of the lightning-node to query

id string

required The id of the lightning invoice.

## Responses

200 Lightning invoice data
 404 The lightning node configuration or the specified invoice was not found
 503 Unable to access the lightning node



```
Copy Expand all Collapse all

{
    "id": "string",
    "status": "Expired",
    "BOLT11": "string",
    "paidAt": 1592312018,
    "expiresAt": 1592312018,
    "amount": "string",
    "amountReceived": "string",
    "paymentHash": "string",
    "preimage": "string",
    "customRecords": { }
}
```

## Pay Lightning Invoice

Pay a lightning invoice. In case the payment response times out, the status will be reported as pending and the final status can be resolved using the Get payment endpoint. The default wait time for payment responses is 30 seconds — it might take longer if multiple routes are tried or a hold invoice is getting paid.

→ amount string Nullable
Optional explicit payment amount in millisatoshi (if specified, it overrides the BOLT11 amount)

→ maxFeePercent string <float> Nullable
The fee limit expressed as a percentage of the payment amount

→ maxFeeFlat string Nullable
The fee limit expressed as a fixed amount in satoshi

→ sendTimeout number <seconds> Nullable
Default: 30
The number of seconds after which the payment times out





# Content type application/json Copy Expand all Collapse all "BOLT11": "string", "amount": "string", "maxFeePercent": "6.15", "maxFeeFlat": "21", "sendTimeout": 30 }

## **Response samples**

```
Content type
application/json

Copy Expand all Collapse all

{
    "id": "string",
    "status": "Complete",
    "BOLT11": "string",
    "paymentHash": "string",
    "preimage": "string",
    "createdAt": 1592312018,
    "totalAmount": "string",
    "feeAmount": "string",
}
```

- . .

#### Get invoices

View information about the lightning invoices

AUTHORIZATIONS: API\_Key ( btcpay.server.canviewlightninginvoiceinternalnode )

**AND** Basic

PATH PARAMETERS

→ cryptoCode string required Systems

Example: BTC

The cryptoCode of the lightning-node to query

**QUERY PARAMETERS** 

⊣ pendingOnly boolean Nullable

Default: false

Limit to pending invoices only

→ offsetIndex number Nullable

Default: 0

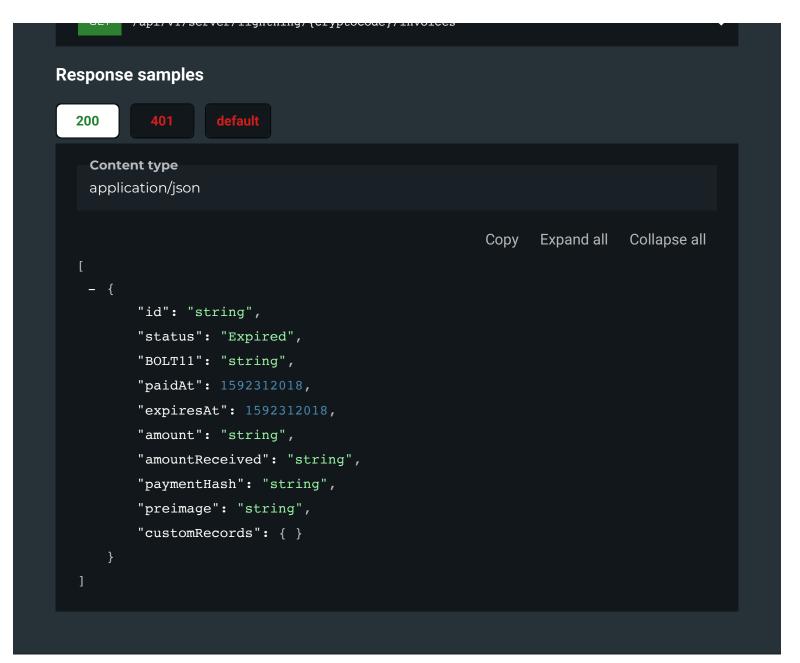
The index of an invoice that will be used as the start of the list

## Responses

- > 200 Lightning invoice data
- > 401 Missing authorization
- 503 Unable to access the lightning node
- > default Unexpected error

GET

/ani/v1/server/lightning//grvntoCodel/invoice



## Create lightning invoice

Create a lightning invoice.

AUTHORIZATIONS: API\_Key( btcpay.server.cancreatelightninginvoiceinternalnode )

**AND** Basic

PATH PARAMETERS

→ cryptoCode string

required Example: BTC

The cryptoCode of the lightning-node to query

REQUEST BODY SCHEMA: application/json

amount string

Amount wrapped in a string, represented in a millistatoshi string. (1000

millisatoshi = 1 satoshi)

→ description string Nullable

Description of the invoice in the BOLT11

→ descriptionHashOnly boolean Nullable

Default: false

If descriptionHashOnly is true (default is false ), then the

BOLT11 returned contains a hash of the description, rather than the description, itself. This allows for much longer descriptions, but they

must be communicated via some other mechanism.

⊣ expiry number < seconds >

Expiration time in seconds

⊣ privateRouteHints boolean Nullable

Default: false

True if the invoice should include private route hints

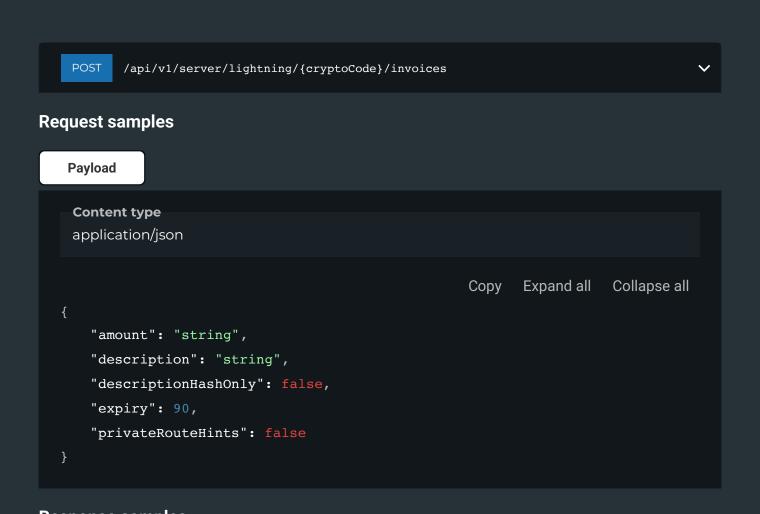
## Responses

> 200 Successfully created

> 400 Wellknown error codes are: invoice-error

404 The lightning node configuration was not found

503 Unable to access the lightning node



## **Response samples**

200

400

```
Content type
application/json

Copy Expand all Collapse all

"id": "string",

"status": "Expired",

"BOLT11": "string",

"paidAt": 1592312018,

"expiresAt": 1592312018,

"amount": "string",
```

```
"paymentHash": "string",
    "preimage": "string",
    "customRecords": { }
}
```

## Get payments

View information about the lightning payments

```
API_Key ( btcpay.server.canuseinternallightningnode ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
→ cryptoCode
                       string
   required
                       Example: BTC
                       The cryptoCode of the lightning-node to query
QUERY PARAMETERS

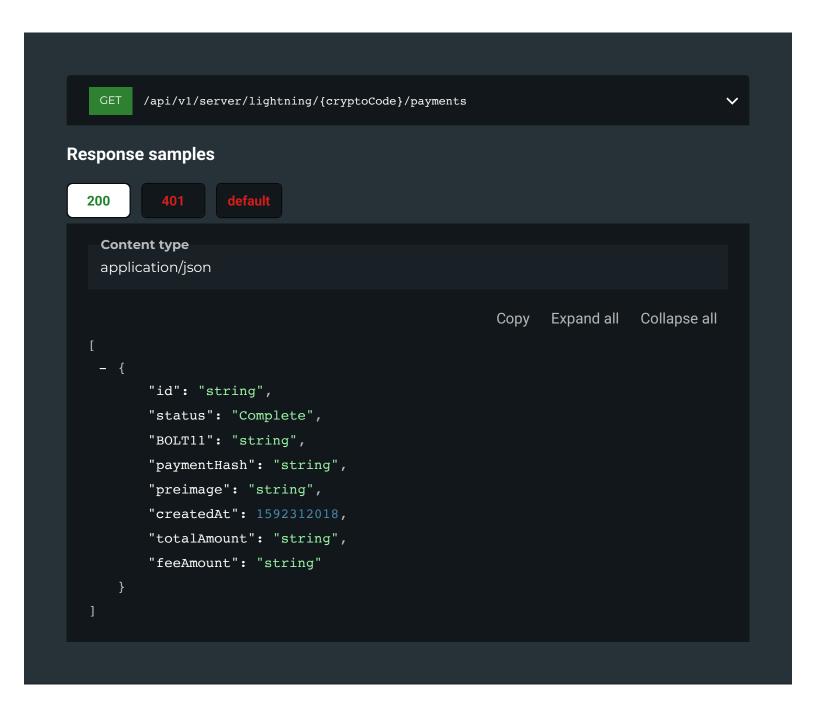
→ includePending

                       boolean Nullable
                       Default: false
                       Also include pending payments
→ offsetIndex
                       number Nullable
                       Default: 0
                       The index of a payment that will be used as the start of the list
```

## Responses

> 200 Lightning payment data

401 Missing authorization
 503 Unable to access the lightning node
 default Unexpected error



# Lightning (Store)

Lightning (Store) operations

## Get node information

View information about the lightning node

AUTHORIZATIONS: API\_Key (| btcpay.store.canuselightningnode |) AND Basic

**PATH PARAMETERS** 

→ cryptoCode string

required Example: BTC

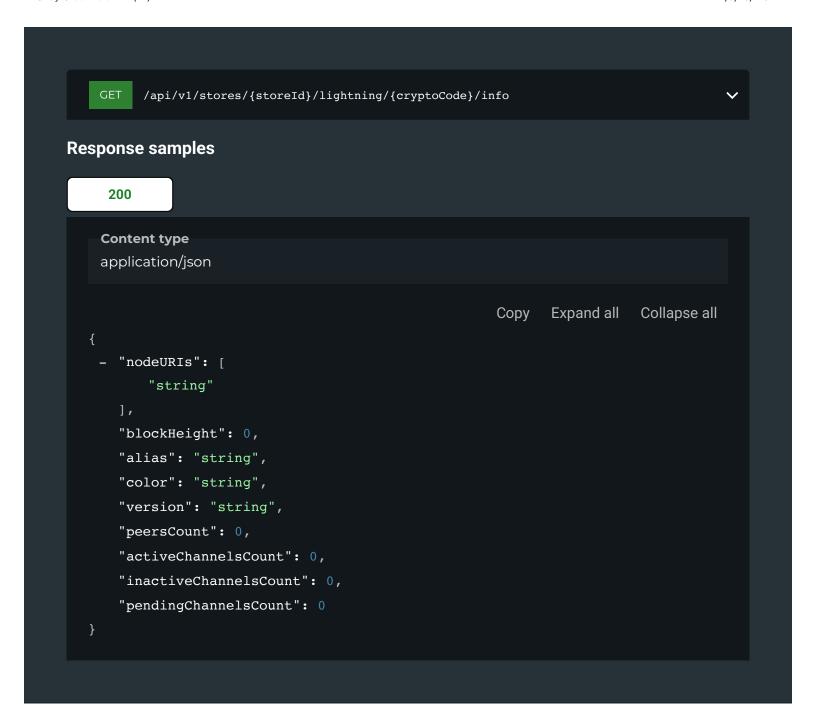
The cryptoCode of the lightning-node to query

storeId string (StoreId)

required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

- > 200 Lightning node information such as reachable nodeinfos
- 404 The lightning node configuration was not found
- 503 Unable to access the lightning node



## Get node balance

View balance of the lightning node

AUTHORIZATIONS: API\_Key ( btcpay.store.canuselightningnode ) AND Basic

PATH PARAMETERS

required string

Example: BTC

The cryptoCode of the lightning-node to query

storeId

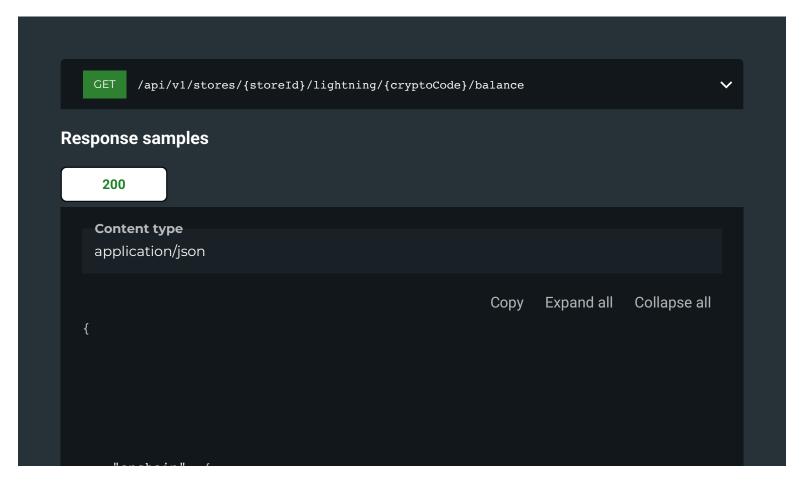
required string (StoreId)

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

## Responses

200 Lightning node balance for on-chain and off-chain funds
 404 The lightning node configuration was not found
 503 Unable to access the lightning node



```
- Onenain : {
    "confirmed": "string",
    "unconfirmed": "string",
    "reserved": "string"
},
- "offchain": {
    "opening": "string",
    "local": "string",
    "remote": "string",
    "closing": "string"
}
}
```

## Get node balance histogram

View balance histogram of the lightning node

```
API_Key ( btcpay.store.canuselightningnode ) AND Basic

PATH PARAMETERS

string
Example: BTC
The cryptoCode of the lightning-node to query

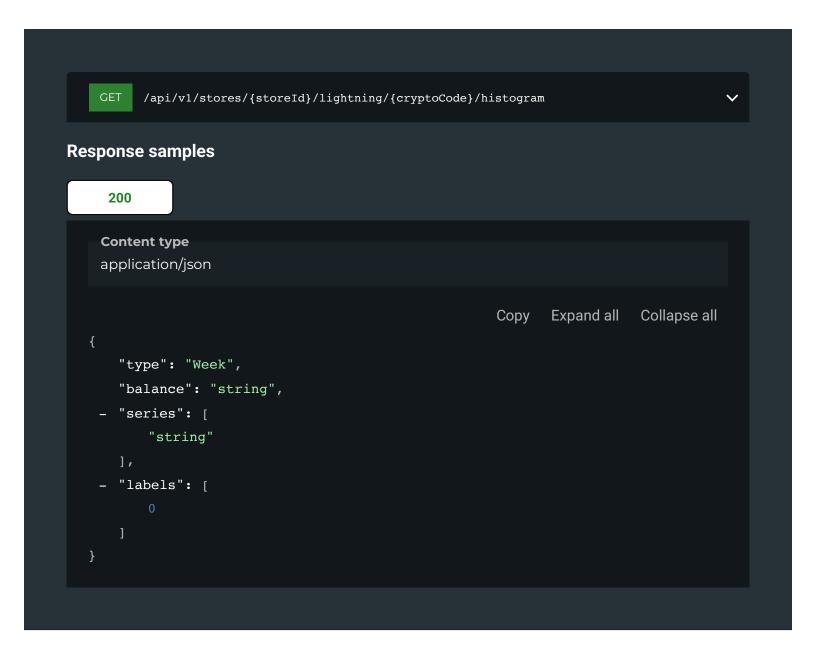
storeId
required

string (StoreId)
Example: 9CinzKoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID
```

## Responses

**> 200** Lightning node balance histogram for off-chain funds

404 The lightning node configuration was not found
503 Unable to access the lightning node



## Connect to lightning node

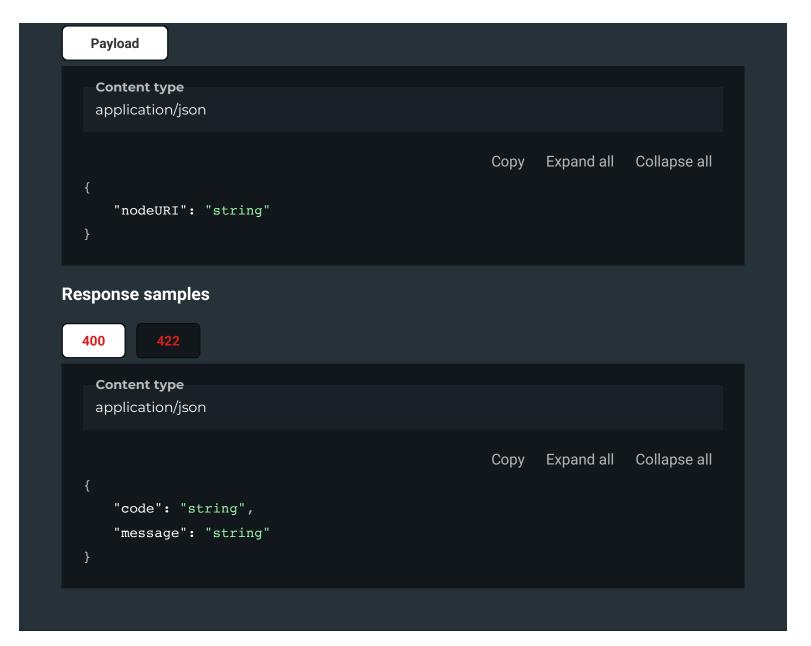
Connect to another lightning node.

API\_Key (|btcpay.store.cancreatelightninginvoice|) AND Basic **AUTHORIZATIONS:** PATH PARAMETERS cryptoCode string required Example: BTC The cryptoCode of the lightning-node to query ⊣ storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json → nodeURI string Nullable Node URI in the form pubkey@endpoint[:port]

## Responses

200 Successfully connected
 400 Wellknown error codes are: could-not-connect
 404 The lightning node configuration was not found
 422 Unable to validate the request
 503 Unable to access the lightning node





## Get channels

View information about the current channels of the lightning node

```
AVITHORIZATIONS:

PATH PARAMETERS

API_Key ( btcpay.store.cancreatelightninginvoice ) AND Basic

required string

Example: BTC
```

```
The cryptoCode of the lightning-node to query

storeId string (StoreId)

required Example: 9CinzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID
```

```
> 200 list of channels

— 404 The lightning node configuration was not found
```

```
/api/v1/stores/{storeId}/lightning/{cryptoCode}/channels
Response samples
     200
    Content type
    application/json
                                                                           Collapse all
                                                                Expand all
                                                        Copy
           "remoteNode": "string",
          "isPublic": true,
           "isActive": true,
           "capacity": "string",
          "localBalance": "string",
           "channelPoint": "string"
```

# Open channel

Open a channel with another lightning node. You should connect to that node first.

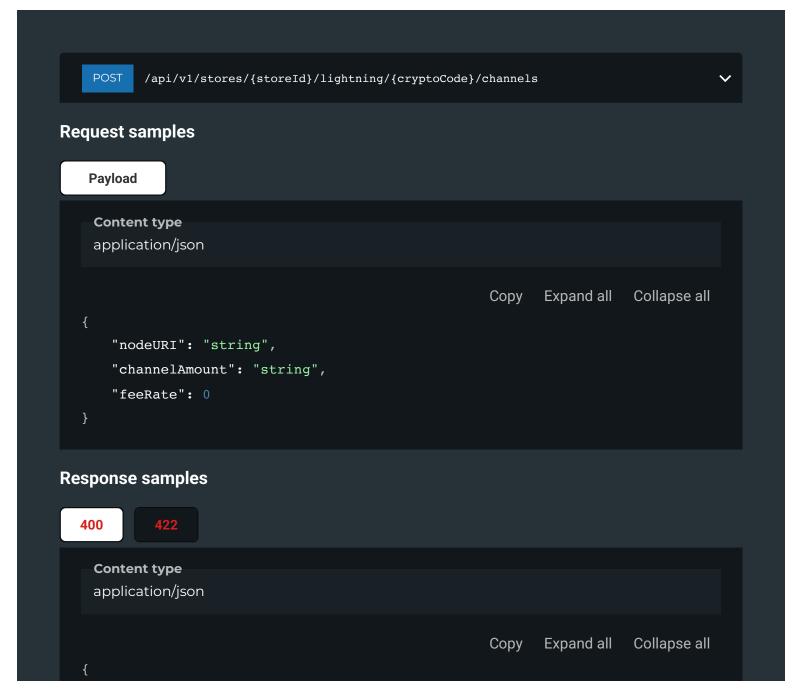
AUTHORIZATIONS:	API_Key( btcpay.store.cancreatelightninginvoice ) AND Basic
PATH PARAMETERS	
⊣ cryptoCode required	string Example: BTC The cryptoCode of the lightning-node to query
⊣ storeId required	string (StoreId)  Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776  The store ID
REQUEST BODY SCHEMA:	application/json
-ı nodeURI	string Node URI in the form pubkey@endpoint[:port]
⊣ channelAmount	string The amount to fund (in satoshi)
—∣ feeRate	number The amount to fund (in satoshi per byte)

## Responses

— 200 Successfully opened

```
> 400
Wellknown error codes are: channel-already-exists , cannot-afford-funding , need-more-
confirmations , peer-not-connected

- 404 The lightning node configuration was not found
> 422 Unable to validate the request
- 503 Unable to access the lightning node
```



```
"code": "string",
    "message": "string"
}
```

# Get deposit address

Get an on-chain deposit address for the lightning node

```
API_Key ( btcpay.store.cancreatelightninginvoice ) AND Basic

PATH PARAMETERS

String (StoreId)
Example: 9CinzKoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID

cryptoCode
required
Example: BTC
The cryptoCode of the lightning-node to query
```

```
    200 deposit address
    404 The lightning node configuration was not found
    503 Unable to access the lightning node
```



## Get payment

View information about the requested lightning payment



```
    200 Lightning payment data
    404 The lightning node configuration or the specified invoice was not found
    503 Unable to access the lightning node
```

```
/api/v1/stores/{storeId}/lightning/{cryptoCode}/payments/{paymentHash}
Response samples
     200
    Content type
    application/json
                                                               Expand all
                                                                           Collapse all
                                                        Copy
      "id": "string",
      "status": "Complete",
      "BOLT11": "string",
      "paymentHash": "string",
       "preimage": "string",
       "createdAt": 1592312018,
       "totalAmount": "string",
      "feeAmount": "string"
```

#### UCL IIIVUICC

View information about the requested lightning invoice

API\_Key ( btcpay.store.canviewlightninginvoice ) AND Basic

PATH PARAMETERS

| cryptoCode required | string | Example: BTC | The cryptoCode of the lightning-node to query

| storeId required | string (StoreId) | Example: 9CinzkoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 | The store ID

The id of the lightning invoice.

### Responses

required

200 Lightning invoice data
 404 The lightning node configuration or the specified invoice was not found
 503 Unable to access the lightning node



```
Copy Expand all Collapse all

{
    "id": "string",
    "status": "Expired",
    "BOLT11": "string",
    "paidAt": 1592312018,
    "expiresAt": 1592312018,
    "amount": "string",
    "amountReceived": "string",
    "paymentHash": "string",
    "preimage": "string",
    "customRecords": { }
}
```

## Pay Lightning Invoice

Pay a lightning invoice. In case the payment response times out, the status will be reported as pending and the final status can be resolved using the Get payment endpoint. The default wait time for payment responses is 30 seconds — it might take longer if multiple routes are tried or a hold invoice is getting paid.

```
API_Key ( btcpay.store.cancreatelightninginvoice ) AND Basic

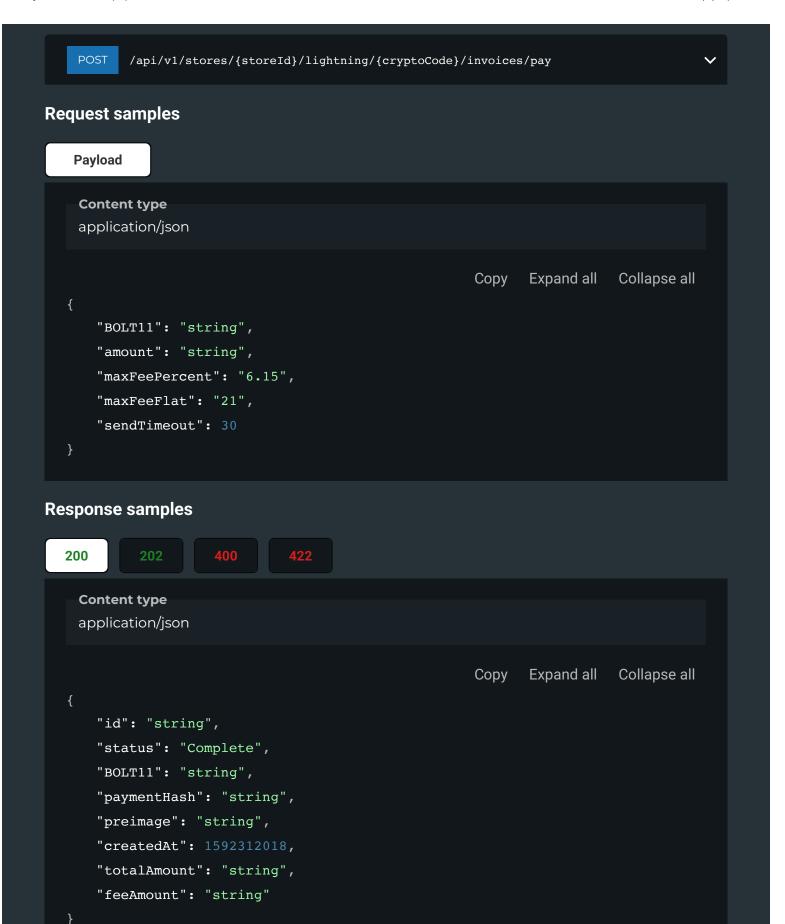
PATH PARAMETERS

| cryptoCode required | Example: BTC | The cryptoCode of the lightning-node to query

| storeId string (StoreId) | Example: 9CinzkoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
```

	The store ID application/json	
REQUEST BODY SCHEMA:		
⊢ BOLT11	string	
	The BOLT11 of the invoice to pay	
⊣ amount	string Nullable	
	Optional explicit payment amount in millisatoshi (if specified, it overrides the	
	BOLT11 amount)	
→ maxFeePercent	string <float> Nullable</float>	
	The fee limit expressed as a percentage of the payment amount	
─ maxFeeFlat	string Nullable	
	The fee limit expressed as a fixed amount in satoshi	
⊣ sendTimeout	number <seconds> Nullable</seconds>	
	Default: 30	
	The number of seconds after which the payment times out	

> 200 Successfully paid		
> 202 Payment initiated		
> 400 Wellknown error codes are: could-not-find-route, generic-error		
<ul> <li>404 The lightning node configuration was not found</li> </ul>		
> 422 Unable to validate the request		
— 503 Unable to access the lightning node		



### Get invoices

View information about the lightning invoices

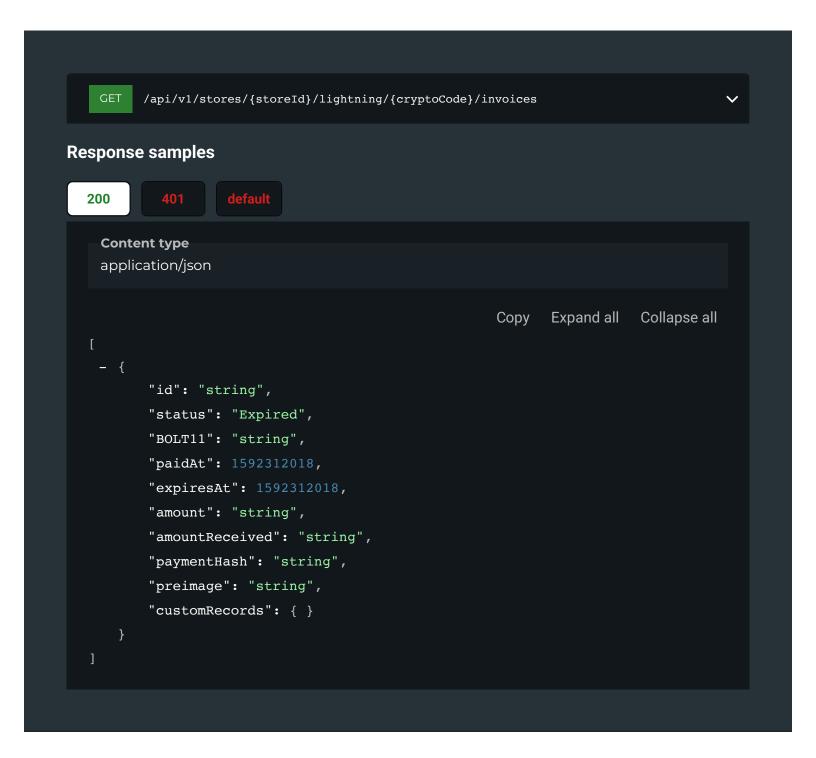
API\_Key (|btcpay.store.canviewlightninginvoice|) AND Basic **AUTHORIZATIONS:** PATH PARAMETERS → cryptoCode string required Example: BTC The cryptoCode of the lightning-node to query ⊣ storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID **QUERY PARAMETERS** → pendingOnly boolean Nullable Default: false Limit to pending invoices only → offsetIndex number Nullable Default: 0 The index of an invoice that will be used as the start of the list

### Responses

> 200 Lightning invoice data

> 401 Missing authorization

503 Unable to access the lightning node
 default Unexpected error



### Create lightning invoice

0,000 119,1011119 11,1000

Create a lightning invoice.

AUTHORIZATIONS: API\_Key(|btcpay.server.cancreatelightninginvoiceinternalnode|)

**AND** Basic

#### PATH PARAMETERS

⊣ cryptoCode string

required Example: BTC

The cryptoCode of the lightning-node to query

- storeId string (StoreId)

required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

### REQUEST BODY SCHEMA: application/json

⊣ amount string

Amount wrapped in a string, represented in a millistatoshi string. (1000

millisatoshi = 1 satoshi)

→ description string Nullable

Description of the invoice in the BOLT11

→ descriptionHashOnly boolean Nullable

Default: false

If descriptionHashOnly is true (default is false ), then the

BOLT11 returned contains a hash of the description, rather than the description, itself. This allows for much longer descriptions, but they

must be communicated via some other mechanism.

→ expiry number < seconds >

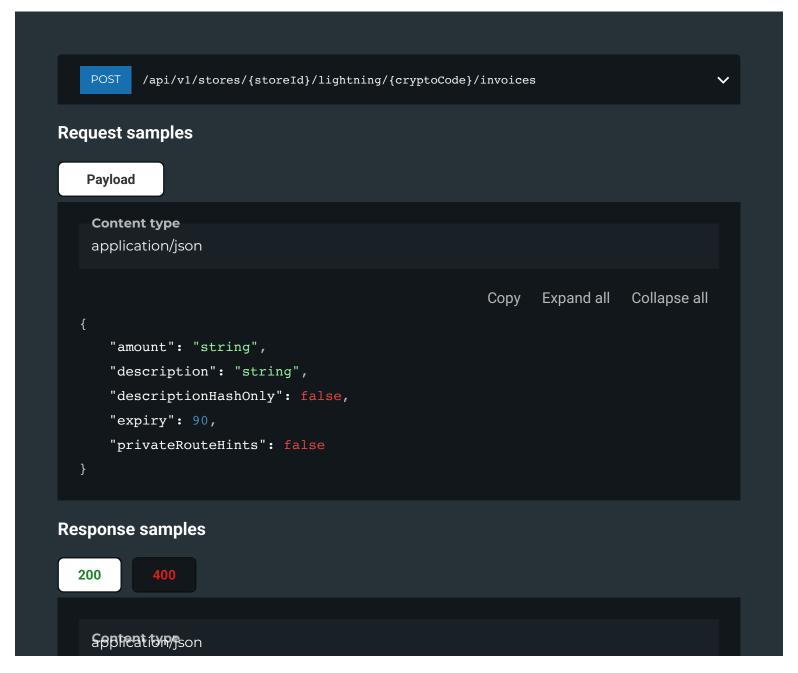
Expiration time in seconds

→ privateRouteHints boolean Nullable

Default: false

True if the invoice should include private route hints

```
    > 200 Successfully created
    > 400 Wellknown error codes are: invoice-error
    - 404 The lightning node configuration was not found
    - 503 Unable to access the lightning node
```



```
Copy Expand all Collapse all

{
    "id": "string",
    "status": "Expired",
    "BOLT11": "string",
    "paidAt": 1592312018,
    "expiresAt": 1592312018,
    "amount": "string",
    "amountReceived": "string",
    "paymentHash": "string",
    "preimage": "string",
    "customRecords": { }
}
```

## Get payments

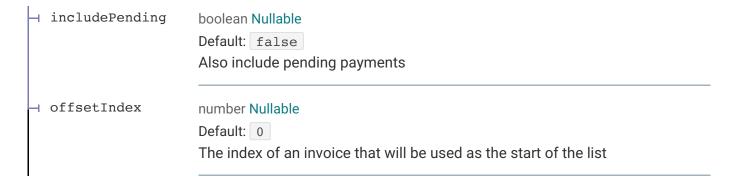
View information about the lightning payments

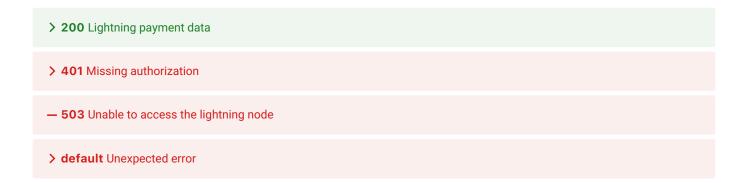
```
API_Key ( btcpay.store.cancreatelightninginvoice ) AND Basic

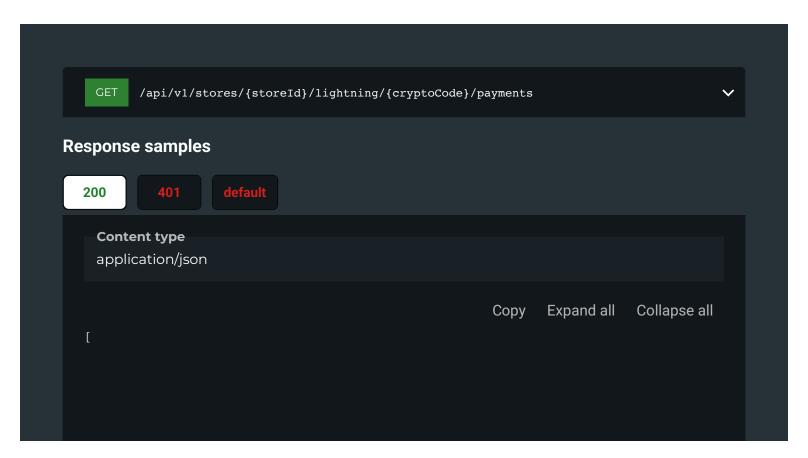
PATH PARAMETERS

| cryptoCode required | Example: BTC | The cryptoCode of the lightning-node to query

| storeId required | string (StoreId) | Example: 9CinzkoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 | The store ID
```







```
- {
    "id": "string",
    "status": "Complete",
    "BOLT11": "string",
    "paymentHash": "string",
    "preimage": "string",
    "createdAt": 1592312018,
    "totalAmount": "string",
    "feeAmount": "string"
}
```

# Miscelleneous

Miscelleneous operations

## Get available rate sources

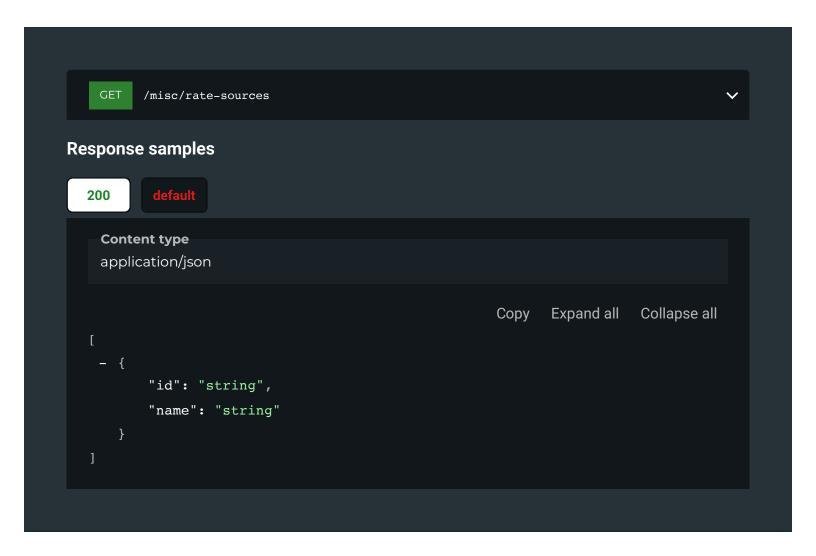
View available rate providers that you can use in stores

### Responses

```
> 200 rate providers array
```

- 403 If you are authenticated but forbidden to get the data

> default Unexpected error



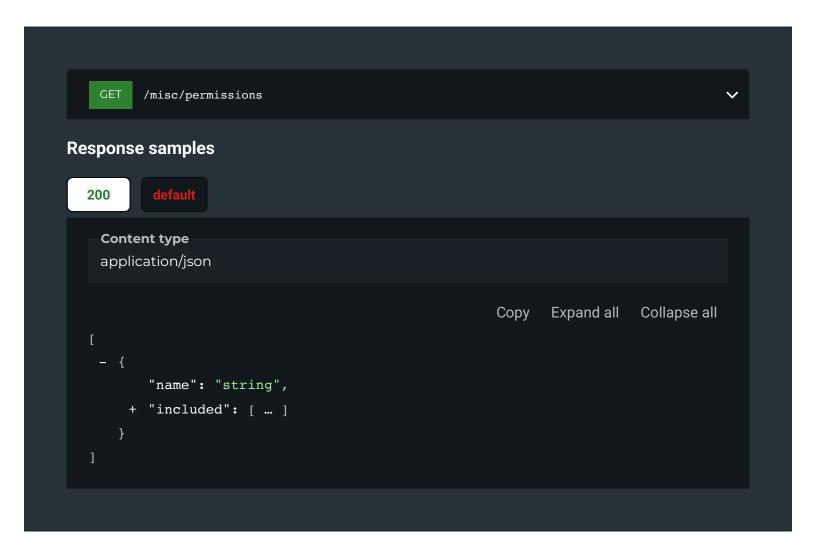
## Permissions metadata

The metadata of available permissions

```
> 200 The metadata of available permissions

— 403 If you are authenticated but forbidden to get the data
```

> default Unexpected error



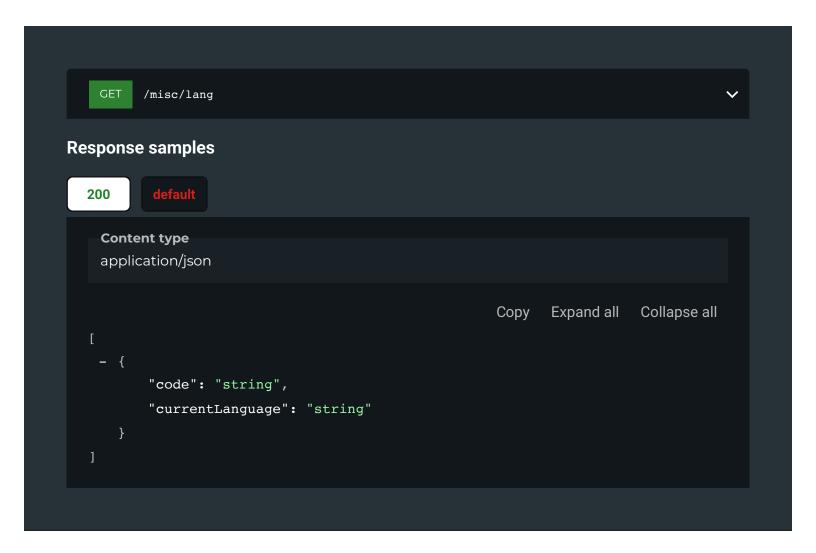
# Language codes

The supported language codes

```
> 200 The supported language codes

— 403 If you are authenticated but forbidden to get the data
```

> default Unexpected error



## Invoice checkout

View the checkout page of an invoice

### PATH PARAMETERS

invoiceId
required

string

The invoice ID

### **QUERY PARAMETERS**

→ lang

string

The preferred language of the checkout page. You can use "auto" to use the language of the customer's browser or see the list of language codes with this operation.

### Responses

**— 200** The checkout page

- 404 The invoice could not be found



# Notifications (Current User)

Notifications operations

## Get notifications

View current user's notifications

AUTHORIZATIONS: API\_Key ( btcpay.user.canmanagenotificationsforuser,

btcpay.user.canviewnotificationsforuser ) AND Basic

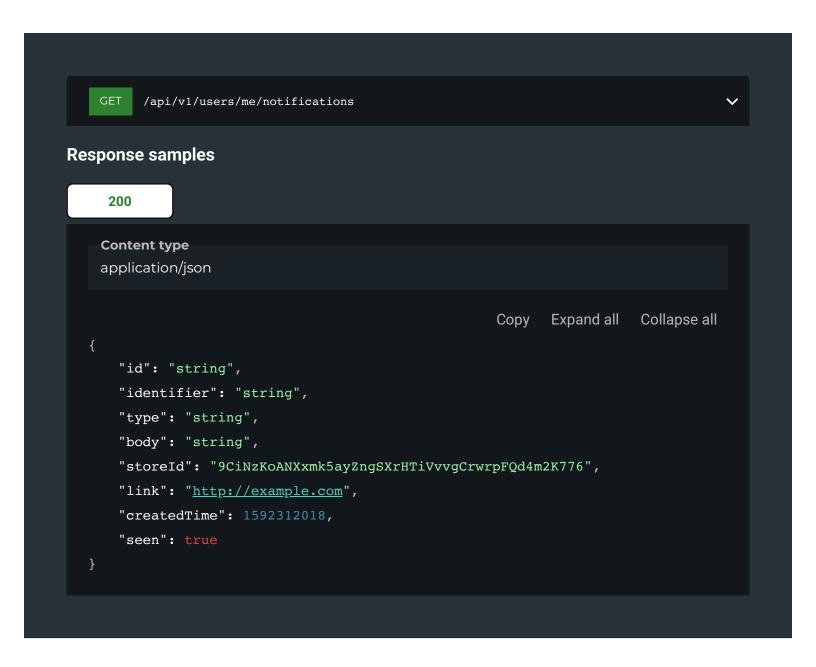
### **QUERY PARAMETERS**

⊣ storeId	Array of strings
	Example: storeId=&storeId=ABCDE&storeId=FGHIJ
	Array of store ids to fetch the notifications for
⊣ take	number Nullable
	Number of records returned in response
⊣ skip	number Nullable
	Number of records to skip
⊣ seen	string Nullable
	filter by seen notifications

## Responses

> 200 list of notifications

- 403 If you are authenticated but forbidden to get the data



## Get notification

View information about the specified notification

API\_Key ( btcpay.user.canmanagenotificationsforuser, btcpay.user.canviewnotificationsforuser ) AND Basic

PATH PARAMETERS

- id string required The notification to fetch

```
    200 specified notification
    403 If you are authenticated but forbidden to view the specified notification
    404 The key is not found for this notification
```

```
Response samples

Content type
application/json

Copy Expand all Collapse all

{
    "id": "string",
    "identifier": "string",
    "type": "string",
    "body": "string",
    "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "link": "http://example.com",
```

```
"createdTime": 1592312018,

"seen": true
}
```

# Update notification

Updates the notification

AUTHORIZATIONS: API\_Key ( btcpay.user.canmanagenotificationsforuser ) AND Basic

**PATH PARAMETERS** 

→ id string

required The notification to update

REQUEST BODY SCHEMA: application/json

→ seen boolean Nullable

Sets the notification as seen/unseen. If left null, sets it to the opposite value

- > 200 updated notification
- 403 If you are authenticated but forbidden to update the specified notification
- 404 The key is not found for this notification

```
Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

{
    "seen": true
}
```

### **Response samples**

200

```
Content type
application/json

Copy Expand all Collapse all

{
    "id": "string",
    "identifier": "string",
    "type": "string",
    "body": "string",
    "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "link": "http://example.com",
    "createdTime": 1592312018,
    "seen": true
}
```

# Remove Notification

Removes the specified notification.

### Responses

200 The notification has been deleted
403 If you are authenticated but forbidden to remove the specified notification
404 The key is not found for this notification



# Get notification settings

View information about your notification settings

**AUTHORIZATIONS:** 

API\_Key ( btcpay.user.canmanagenotificationsforuser ) AND Basic

### Responses

> 200 The current user's notification settings

- 403 If you are authenticated but forbidden to view the notification settings

```
/api/v1/users/me/notification-settings
Response samples
     200
    Content type
    application/json
                                                               Expand all
                                                                           Collapse all
                                                        Copy
          "identifier": "newversion",
           "name": "New version",
           "enabled": false
      },
          "identifier": "pluginupdate",
           "name": "Plugin update",
           "enabled": false
      },
```

```
"identifier": "external-payout-transaction",
    "name": "External payout approval",
   "enabled": true
},
   "identifier": "inviteaccepted",
   "name": "User accepted invitation",
   "enabled": true
},
   "identifier": "invoicestate",
   "name": "All invoice updates",
   "enabled": true
},
   "identifier": "invoicestate_invoice_confirmed",
   "name": "Invoice is settled",
   "enabled": true
},
   "identifier": "invoicestate invoice expiredPaidPartial",
   "name": "Invoice expired with partial payments",
   "enabled": true
},
   "identifier": "invoicestate_invoice_failedToConfirm",
   "name": "Invoice has payments that failed to confirm on time",
   "enabled": true
},
   "identifier": "invoicestate invoice paidAfterExpiration",
    "name": "Invoice was paid after expiration",
    "enabled": true
},
```

```
- {
    "identifier": "newuserrequiresapproval",
    "name": "New user requires approval",
    "enabled": true
},
- {
    "identifier": "payout",
    "name": "Payouts",
    "enabled": true
}
```

## Update notification settings

Updates the current user's notification settings

API\_Key ( btcpay.user.canmanagenotificationsforuser ) AND Basic

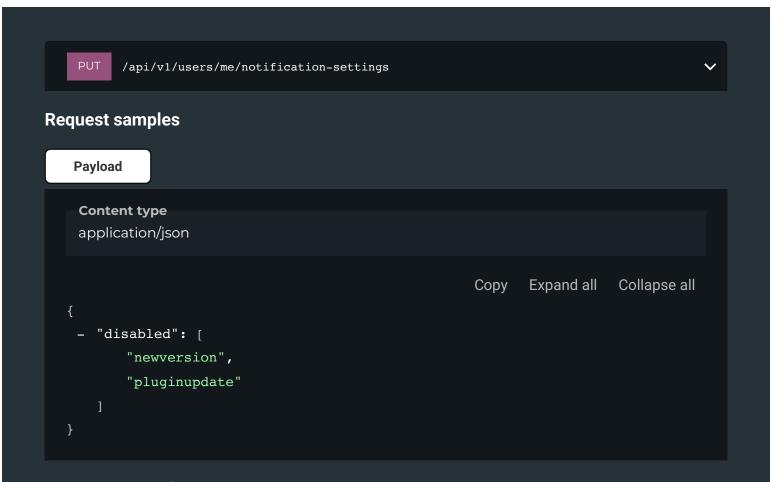
REQUEST BODY SCHEMA: application/json

—I disabled Array of strings
List of the notification type identifiers, which should be disabled. Can also be a single item 'all'.

### Responses

> 200 The current user's notification settings

- 403 If you are authenticated but forbidden to update the notification settings



### **Response samples**

200

```
Content type
application/json

Copy Expand all Collapse all

[
- {
    "identifier": "newversion",
    "name": "New version",
    "enabled": false
},
```

```
"identifier": "pluginupdate",
   "name": "Plugin update",
   "enabled": false
},
   "identifier": "external-payout-transaction",
   "name": "External payout approval",
   "enabled": true
},
   "identifier": "inviteaccepted",
   "name": "User accepted invitation",
   "enabled": true
},
   "identifier": "invoicestate",
   "name": "All invoice updates",
   "enabled": true
},
   "identifier": "invoicestate invoice confirmed",
   "name": "Invoice is settled",
   "enabled": true
},
   "identifier": "invoicestate_invoice_expiredPaidPartial",
   "name": "Invoice expired with partial payments",
   "enabled": true
},
   "identifier": "invoicestate invoice failedToConfirm",
    "name": "Invoice has payments that failed to confirm on time",
    "enabled": true
},
```

```
- {
    "identifier": "invoicestate_invoice_paidAfterExpiration",
    "name": "Invoice was paid after expiration",
    "enabled": true
},
- {
    "identifier": "newuserrequiresapproval",
    "name": "New user requires approval",
    "enabled": true
},
- {
    "identifier": "payout",
    "name": "Payouts",
    "enabled": true
}
```

# Payment Requests

Payment Requests operations

## Get payment requests

View information about the existing payment requests

AUTHORIZATIONS: API\_Key ( btcpay.store.canviewpaymentrequests ) AND Basic PATH PARAMETERS

→ storeId required

string (StoreId)

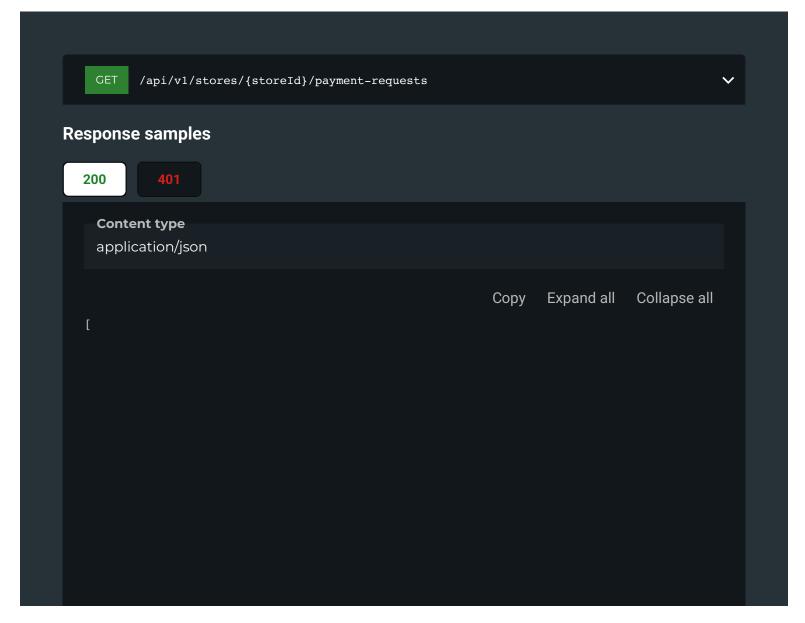
Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

### Responses

> 200 List of payment requests

> 401 Missing authorization



```
- {
    "amount": "string",
    "title": "string",
    "currency": "string",
    "email": "user@example.com",
    "description": "string",
    "expiryDate": 1592312018,
    "allowCustomPaymentAmounts": true,
    "formId": "string",
    "formResponse": { },
    "id": "string",
    "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "status": "Pending",
    "createdTime": 1592312018
  }
}
```

## Create a new payment request

Create a new payment request

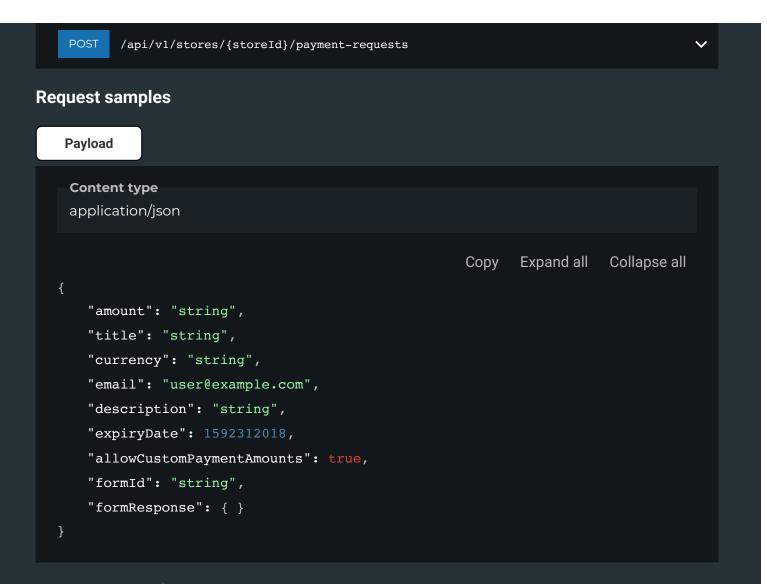
H	title	string The title of the payment request
H	currency	string <iso 4217="" code(btc,="" currency="" etc)="" eur,="" usd,=""> Nullable The currency of the payment request. If empty, the store's default currency code will be used.</iso>
H	email	string <email> Nullable The email used in invoices generated by the payment request</email>
H	description	string

## Responses

> 200 Information about the new payment request

> 400 A list of errors that occurred when creating the payment request

— 403 If you are authenticated but forbidden to add new payment requests



### **Response samples**

```
Content type
application/json

Copy Expand all Collapse all

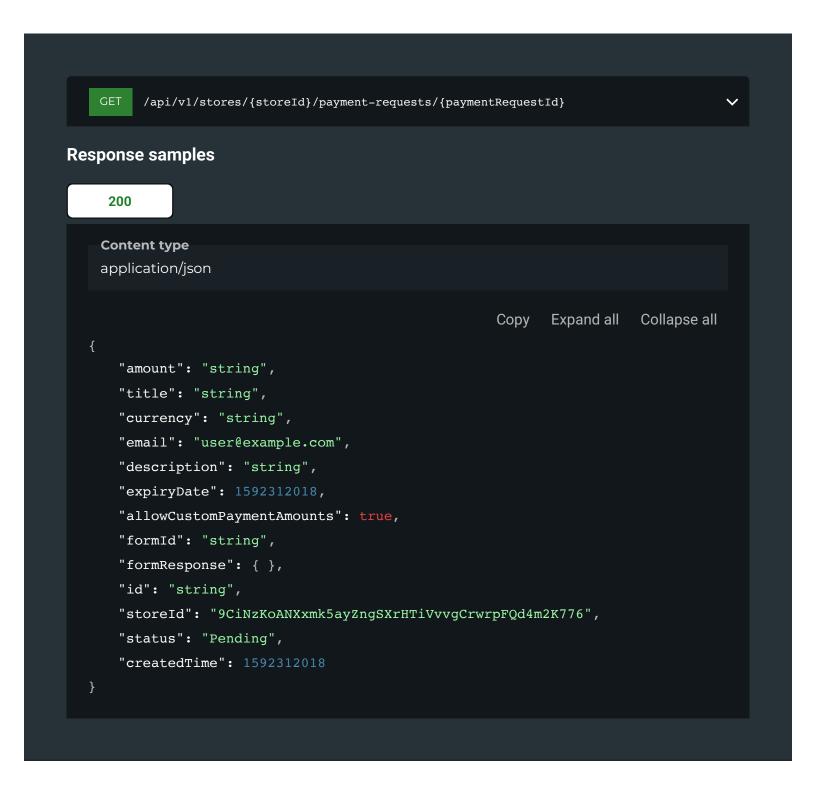
"amount": "string",
"title": "string",
"currency": "string",
"email": "user@example.com",
"description": "string",
"expiryDate": 1592312018
```

```
"allowCustomPaymentAmounts": true,
    "formId": "string",
    "formResponse": { },
    "id": "string",
    "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "status": "Pending",
    "createdTime": 1592312018
}
```

## Get payment request

View information about the specified payment request

```
    200 specified payment request
    403 If you are authenticated but forbidden to view the specified payment request
    404 The key is not found for this payment request
```

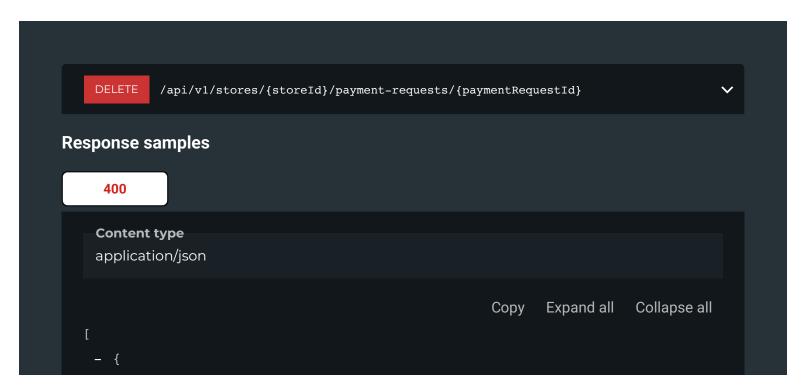


## Archive payment request

Archives the specified payment request.

### 

```
    200 The payment request has been archived
    400 A list of errors that occurred when archiving the payment request
    403 If you are authenticated but forbidden to archive the specified payment request
    404 The key is not found for this payment request
```



## Update payment request

Update a payment request

```
API_Key ( btcpay.store.canmodifypaymentrequests ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
 ⊣ storeId
                        string (StoreId)
   required
                        Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                        The store ID
→ paymentRequestId
                        string
   required
                        The payment request to update
REQUEST BODY SCHEMA:
                        application/json
                                  string <decimal> > 0
  amount
                                  The amount of the payment request

→ title

                                  The title of the payment request
                                  string <ISO 4217 Currency code(BTC, EUR, USD, etc)> Nullable
currency
                                  The currency of the payment request. If empty, the store's default
                                  currency code will be used.
                                  string <email> Nullable
 ⊣ email
                                  The email used in invoices generated by the payment request
```

→ description string <a href="https://www.numbersolution.com">string <a href="https://www.numbersolution.com">https://www.numbersolution.com</a> The description of the payment request

→ expiryDate number <a href="https://www.numbersolution.com">number <a href="https:/

### Responses

200 The updated payment request
 400 A list of errors that occurred when updating the payment request
 403 If you are authenticated but forbidden to update the payment request



```
Copy Expand all Collapse all

"amount": "string",
    "title": "string",
    "email": "user@example.com",
    "description": "string",
    "expiryDate": 1592312018,
    "allowCustomPaymentAmounts": true,
    "formId": "string",
    "formResponse": { }
}
```

#### **Response samples**

200

400

```
Content type
application/json
                                                                    Collapse all
                                                  Copy
                                                        Expand all
  "amount": "string",
  "title": "string",
  "currency": "string",
  "email": "user@example.com",
  "description": "string",
  "expiryDate": 1592312018,
  "allowCustomPaymentAmounts": true,
  "formId": "string",
  "formResponse": { },
  "id": "string",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "status": "Pending",
  "createdTime": 1592312018
```

## Create a new invoice for the payment request

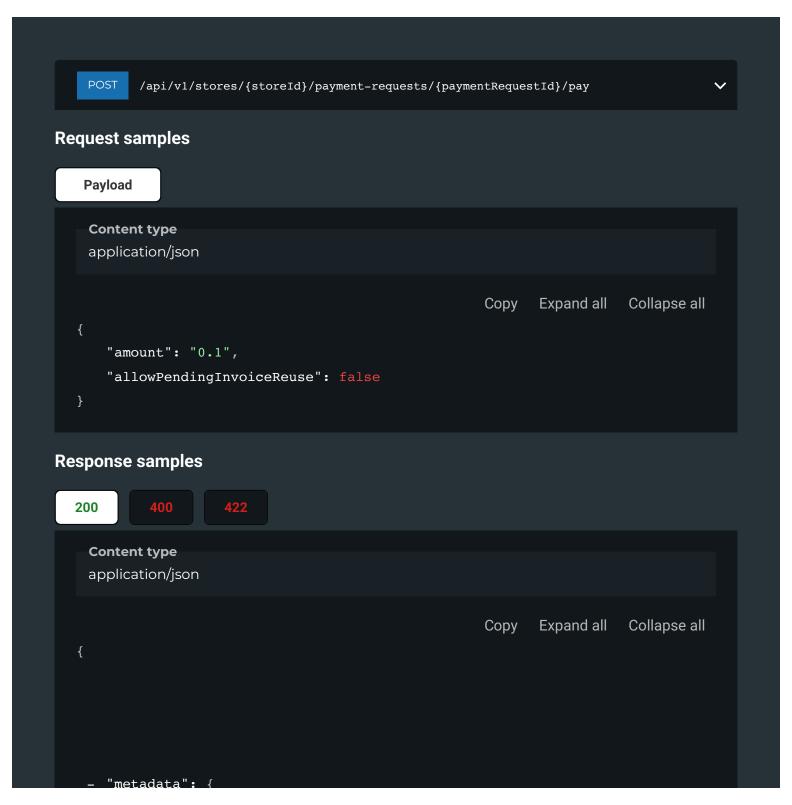
Create a new invoice for the payment request, or reuse an existing one

```
AUTHORIZATIONS:
                       API_Key ( btcpay.store.canviewpaymentrequests ) AND Basic
PATH PARAMETERS
 storeId
                       string (StoreId)
  required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
→ paymentRequestId
                       string
   required
                       The payment request to create
REQUEST BODY SCHEMA:
                        application/json
Invoice creation request
                                string <decimal> > 0 Nullable
→ amount
                                The amount of the invoice. If null or unspecified, it will be
                                set to the payment request's due amount. Note that the payment's
                                request allowCustomPaymentAmounts must be true, or a 422
                                error will be sent back.'
 - allowPendingInvoiceReuse boolean Nullable
                                Default: false
                                If true, this endpoint will not necessarily create a new invoice,
                                and instead attempt to give back a pending one for this payment
                                request.
```

## Responses

> 200 A new invoice

```
> 400 Wellknown error codes are: archived, already-paid, expired
> 422 Unable to validate the request
```



```
"orderId": "pos-app 346KRC5BjXXX08cRFKwTBmdR6ZJ4",
      "orderUrl": "https://localhost:14142/apps/346KRC5BjXXXo8cRFKwTBmdR6ZJ4
      "itemDesc": "Tea shop",
   + "posData": { ... },
   + "receiptData": { ... }
  },
- "checkout": {
      "speedPolicy": "HighSpeed",
   + "paymentMethods": [ ... ],
      "defaultPaymentMethod": "BTC-CHAIN",
      "lazyPaymentMethods": true,
      "expirationMinutes": 90,
      "monitoringMinutes": 90,
      "paymentTolerance": 0,
      "redirectURL": "string",
      "redirectAutomatically": true,
      "defaultLanguage": "string"
  },
- "receipt": {
      "enabled": true,
      "showQR": null,
      "showPayments": null
  },
  "id": "string",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "amount": "5.00",
  "currency": "USD",
  "type": "Standard",
  "checkoutLink": "string",
  "createdTime": 1592312018,
  "expirationTime": 1592312018,
  "monitoringExpiration": 1592312018,
  "status": "Expired",
  "additionalStatus": "Invalid",
  "availableStatusesForManualMarking": [
```

```
"Expired"
],
"archived": true
}
```

# **Payout Processors**

**Payout Processors operations** 

# Get payout processors

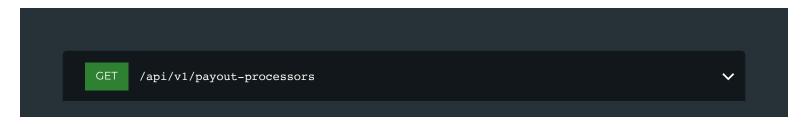
Get payout processors available in this instance

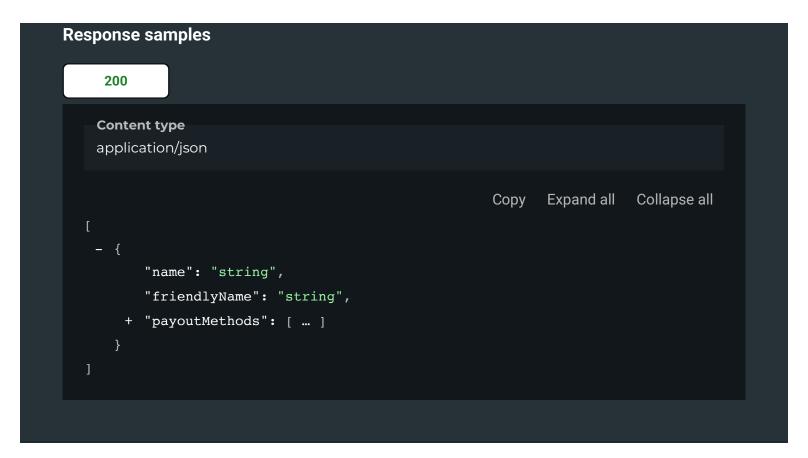
AUTHORIZATIONS: API\_Key AND Basic

## Responses

> 200 available payout processors

— 403 If you are authenticated but forbidden to get the data





# Pull payments (Management)

Pull payments (Management) operations

## Get store's pull payments

Get the pull payments of a store

```
AUTHORIZATIONS: API_Key ( btcpay.store.canmanagepullpayments ) AND Basic

PATH PARAMETERS

— storeId string (StoreId)
```

required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

QUERY PARAMETERS

→ includeArchived

boolean

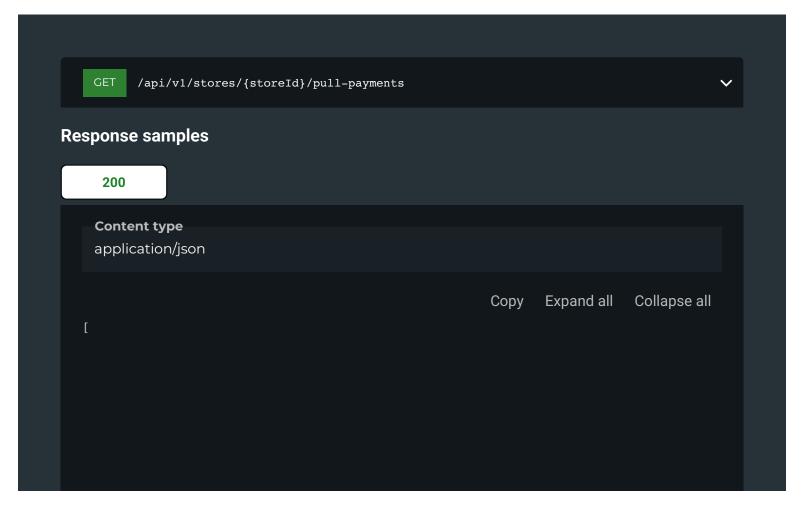
Default: false

Whether this should list archived pull payments

### Responses

> 200 List of pull payments

- 403 If you are authenticated but forbidden to get the data



```
- {
    "id": "string",
    "name": "string",
    "description": "string",
    "currency": "BTC",
    "amount": "1.12000000",
    "BOLT11Expiration": 30,
    "autoApproveClaims": false,
    "archived": true,
    "viewLink": "string"
  }
}
```

## Create a new pull payment

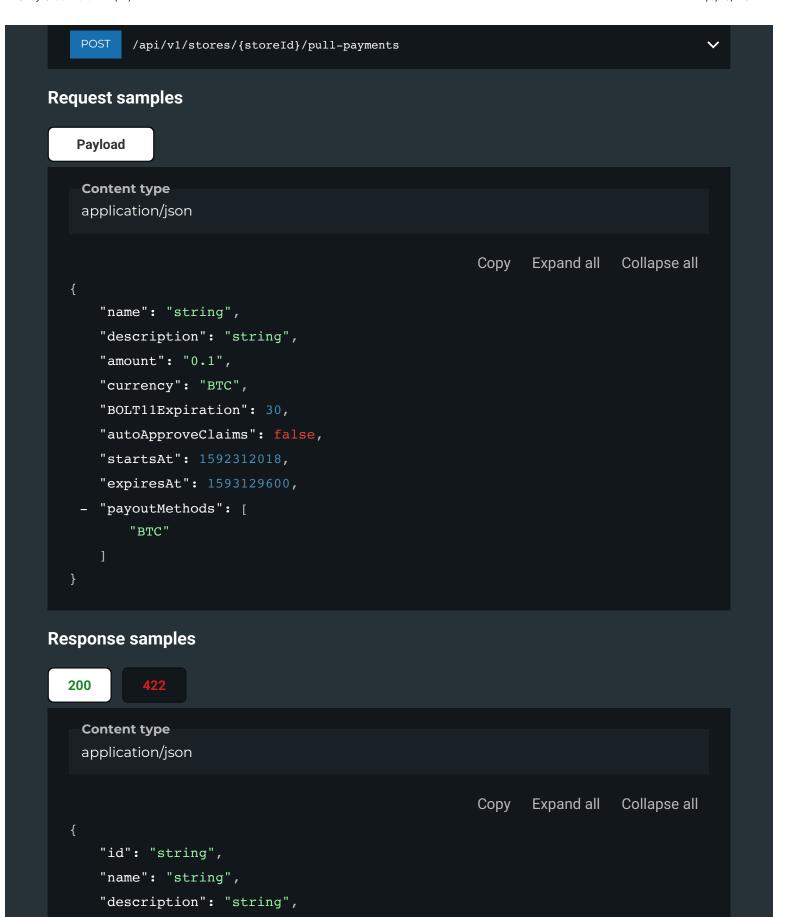
A pull payment allows its receiver to ask for payouts up to amount of currency every period. API\_Key (|btcpay.store.cancreatepullpayments, **AUTHORIZATIONS:** btcpay.store.cancreatenonapprovedpullpayments ) AND Basic PATH PARAMETERS → storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json ⊣ name string Nullable The name of the pull payment → description string Nullable The description of the pull payment

amount string <decimal> The amount in currency of this pull payment as a decimal string - currency string The currency of the amount. → BOLT11Expiration string Nullable Default: 30 If lightning is activated, do not accept BOLT11 invoices with expiration less than ... days → autoApproveClaims boolean Nullable Default: false Any payouts created for this pull payment will skip the approval phase upon creation ⊣ startsAt integer <unix timestamp in seconds> Nullable When this pull payment is effective. Already started if null or unspecified. → expiresAt integer <unix timestamp in seconds> Nullable When this pull payment expires. Never expires if null or unspecified. payoutMethods Array of strings Nullable The list of supported payout methods supported by this pull payment. Available options can be queried from the StorePaymentMethods GetStorePaymentMethods endpoint. If null, all available payout methods will be supported.

## Responses

> 200 The create pull payment

> 422 Unable to validate the request



```
"currency": "BTC",
    "amount": "1.12000000",
    "BOLT11Expiration": 30,
    "autoApproveClaims": false,
    "archived": true,
    "viewLink": "string"
}
```

# Archive a pull payment

Archive this pull payment (Will cancel all payouts awaiting for payment)

AUTHORIZATIONS:	API_Key( btcpay.store.canarchivepullpayments ) AND Basic
PATH PARAMETERS	
⊣ storeId required	string (StoreId)  Example: 9CinzKoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776  The store ID
→ pullPaymentId required	string The ID of the pull payment

- 200 The pull payment has been archived
- 404 The pull payment has not been found, or does not belong to this store



# Pull payments (Public)

Pull payments (Public) operations

PATH PARAMETERS

## Link a boltcard to a pull payment

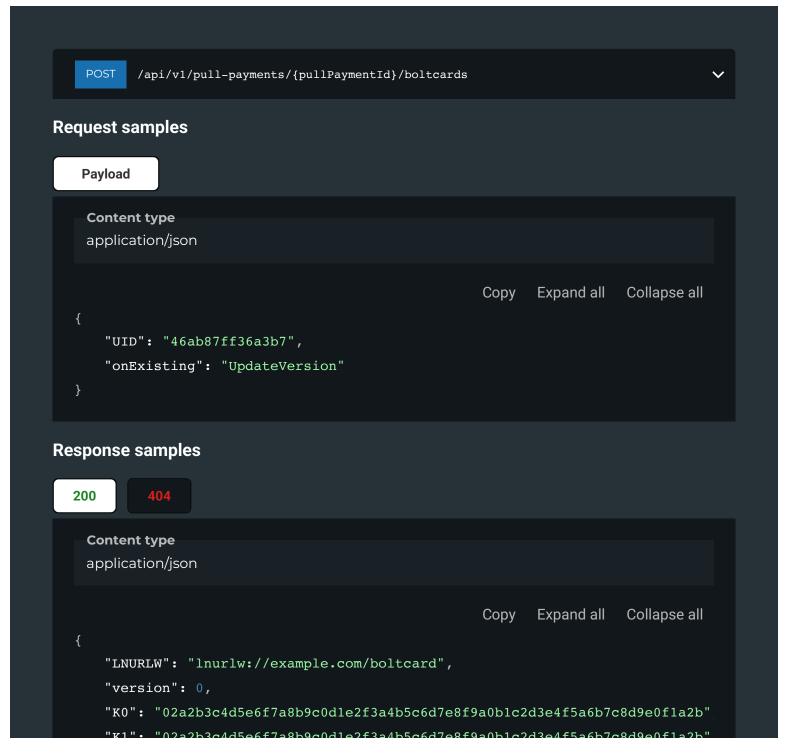
Linking a boltcard to a pull payment will allow you to pay via NFC with it, the money will be sent from the pull payment. The boltcard keys are generated using Deterministic Boltcard Key Generation.

## → pullPaymentId string required The ID of the pull payment REQUEST BODY SCHEMA: application/json UID string required The **UID** of the NTag424 onExisting string Default: "UpdateVersion" Enum: "KeepVersion" "UpdateVersion" What to do if the boltcard is already linked. **KeepVersion** will return the keys (K0-K4) that are already registered. UpdateVersion will increment the version of the key, and thus return different keys (K0-K4). (See Deterministic Boltcard Key Generation)

#### Responses

> 200 The boltcard has been linked to the pull payment.

> 404 The pull payment has not been found. Wellknown error code is: pullpayment-not-found



## Get Pull Payment

Get a pull payment

#### PATH PARAMETERS

→ pullPaymentId required strina

The ID of the pull payment

- > 200 Information about the pull payment
- 404 Pull payment not found



```
Content type application/json

Copy Expand all Collapse all 

{
    "id": "string",
    "name": "string",
    "description": "string",
    "currency": "BTC",
    "amount": "1.12000000",
    "BOLT11Expiration": 30,
    "autoApproveClaims": false,
    "archived": true,
    "viewLink": "string"
}
```

# **Get Payouts**

Get payouts

#### **PATH PARAMETERS**

- pullPaymentId
 required

string

The ID of the pull payment

#### **QUERY PARAMETERS**

→ includeCancelled boolean

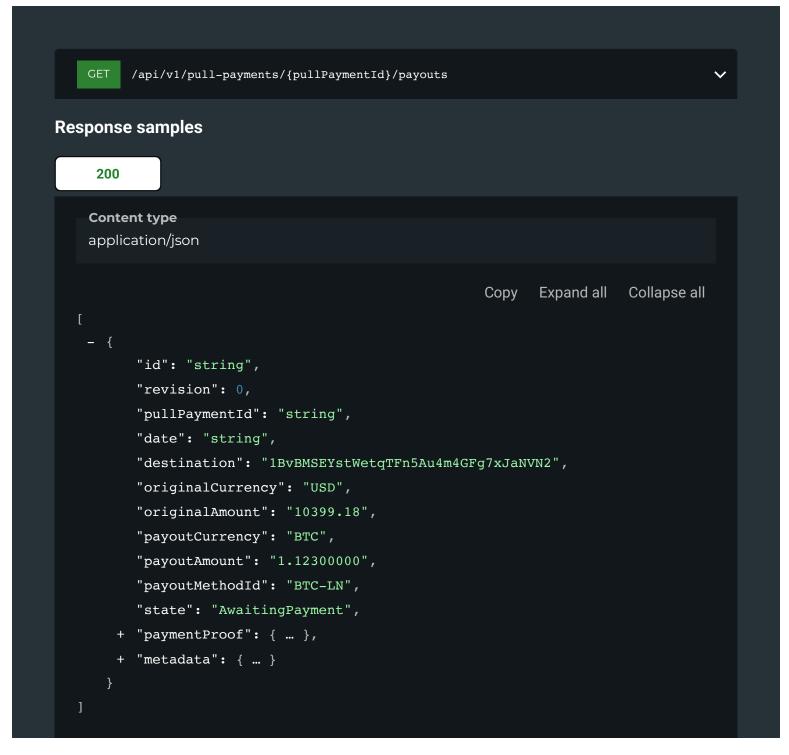
Default: false

Whether this should list cancelled payouts

#### Responses

> 200 The payouts of the pull payment

- 404 Pull payment not found



## Create Payout

Create a new payout

#### PATH PARAMETERS

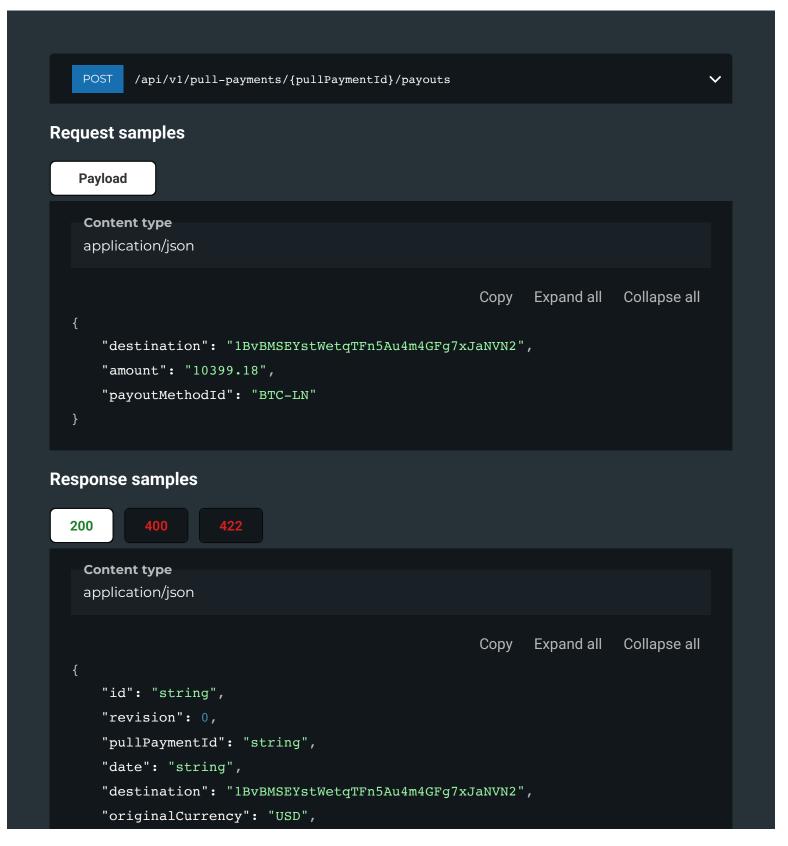
→ pullPaymentId required	string The ID of the pull payment
REQUEST BODY SCHEMA:	application/json
⊣ destination	string
	The destination of the payout (can be an address or a BIP21 url)
⊣ amount	string <decimal> The amount of the payout in the currency of the pull payment (eg. USD).</decimal>
⊣ payoutMethodId	string (PayoutMethodId) Payout method IDs. Available payment method IDs for Bitcoin are:
	<ul><li>"BTC-CHAIN": Onchain</li><li>"BTC-LN": Lightning</li></ul>

```
> 200 A new payout has been created

> 400
Wellknown error codes are: duplicate-destination , expired , not-started , archived ,
overdraft , amount-too-low , payment-method-not-supported

- 404 Pull payment not found
```

> 422 Unable to validate the request



```
"originalAmount": "10399.18",
    "payoutCurrency": "BTC",
    "payoutAmount": "1.12300000",
    "payoutMethodId": "BTC-LN",
    "state": "AwaitingPayment",
    - "paymentProof": {
        "id": "string"
    },
    - "metadata": {
        "source": "Payout created through the API"
    }
}
```

## Get Payout

#### Get payout

#### PATH PARAMETERS

```
required
The ID of the pull payment

payoutId string
required
The ID of the pull payment payout

The ID of the pull payment payout
```

### Responses

```
> 200 A specific payout of a pull payment
```

- 404 Pull payment payout not found

GET /api/v1/pull-payments/{pullPaymentId}/payouts/{payoutId}

#### Response samples

200

```
Content type
application/json
                                                        Expand all Collapse all
                                                 Copy
  "id": "string",
  "revision": 0,
  "pullPaymentId": "string",
  "date": "string",
  "destination": "1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2",
  "originalCurrency": "USD",
  "originalAmount": "10399.18",
  "payoutCurrency": "BTC",
  "payoutAmount": "1.12300000",
  "payoutMethodId": "BTC-LN",
  "state": "AwaitingPayment",
- "paymentProof": {
      "id": "string"
  },
- "metadata": {
      "source": "Payout created through the API"
```

## Get Pull Payment LNURL details

Get Pull Payment LNURL details

#### **PATH PARAMETERS**

- pullPaymentId
 required

string

The ID of the pull payment

```
> 200 Pull payment LNURL details

— 400 Pull payment found but does not support LNURL

— 404 Pull payment not found
```

```
Response samples

200

Content type
application/json

Copy Expand all Collapse all

{
  "lnurlBech32": "lightning:lnurlldp68gup69uhnzv3h9cczuvpwxyarzdp3xsez7sj5g
  "lnurlUri": "lnurlw://example.com/BTC/UILNURL/withdraw/pp/42kktmpL5d6qVc8!
}
```

# Pull payments payout (Public)

Pull payments payout (Public) operations

## Get Payout

Get payout

#### **PATH PARAMETERS**

pullPaymentId string
required The ID of the pull payment

→ payoutId required

string

The ID of the pull payment payout

## Responses

> 200 A specific payout of a pull payment

- 404 Pull payment payout not found



/api/v1/pull-payments/{pullPaymentId}/payouts/{payoutId}



#### **Response samples**

200

```
Content type
application/json
                                                        Expand all Collapse all
                                                 Copy
  "id": "string",
  "revision": 0,
  "pullPaymentId": "string",
  "date": "string",
  "destination": "1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2",
  "originalCurrency": "USD",
  "originalAmount": "10399.18",
  "payoutCurrency": "BTC",
  "payoutAmount": "1.12300000",
  "payoutMethodId": "BTC-LN",
  "state": "AwaitingPayment",
- "paymentProof": {
      "id": "string"
  },
- "metadata": {
      "source": "Payout created through the API"
```

## ServerInfo

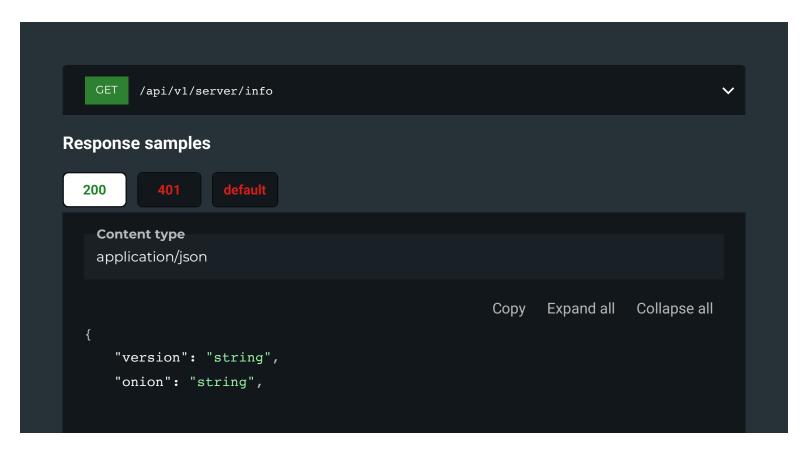
Server Info operations

### Get server info

Information about the server, chains and sync states

AUTHORIZATIONS: API\_Key AND Basic

```
    200 Server information
    401 Missing authorization
    default Unexpected error
```

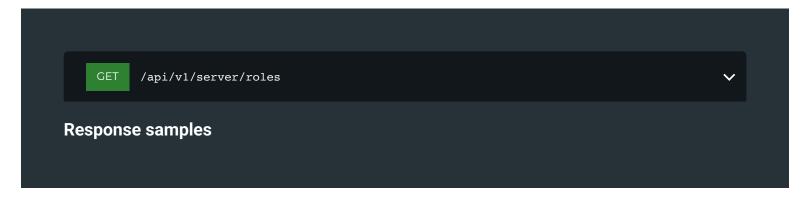


### Get store's roles

View information about the store's roles at the server's scope

```
AUTHORIZATIONS: API_Key ( btcpay.server.canmodifyserversettings ) AND Basic
```

```
    200 The user roles available at the server's scope
    403 If you are authenticated but forbidden to get the store's roles
    404 Store not found
```



```
Content type
application/json

Copy Expand all Collapse all

{
    "id": "Owner",
    "role": "Owner",
    - "permissions": [
        "btcpay.store.canmodifystoresettings"
    ],
    "isServerRole": true
}
```

# Stores (Email)

Store Emails operations

## Get store email settings

View email settings of the specified store

The store ID

### Responses

200 specified store email settings
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store

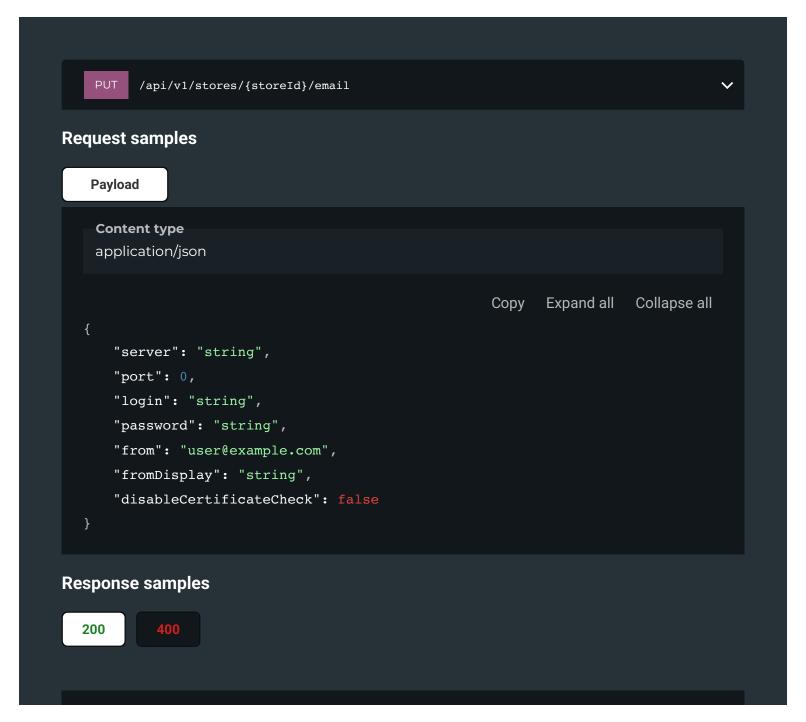
```
/api/v1/stores/{storeId}/email
Response samples
     200
    Content type
    application/json
                                                               Expand all
                                                                          Collapse all
                                                        Copy
      "server": "string",
      "port": 0,
      "login": "string",
      "password": "string",
      "from": "user@example.com",
       "fromDisplay": "string",
       "disableCertificateCheck": false
```

# Update store email settings

Update a store's email settings

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS	
→ storeId     required	string (StoreId)  Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776  The store ID
REQUEST BODY SCHEMA:	application/json
⊣ server	string Smtp server host
⊣ port	number Smtp server port
⊣ login	string Smtp server username
⊢ password	string Smtp server password
⊣ from	string <email> Email to send from</email>
⊢ fromDisplay	string The name of the sender
⊣ disableCertificat	eCheck boolean  Default: false  Disable TLS certificate security checks

```
    200 The settings were updated
    400 A list of errors that occurred when updating the settings
    403 If you are authenticated but forbidden to modify the store
```



```
Content type
application/json

Copy Expand all Collapse all

{
    "server": "string",
    "port": 0,
    "login": "string",
    "password": "string",
    "from": "user@example.com",
    "fromDisplay": "string",
    "disableCertificateCheck": false
}
```

## Send an email for a store

Send an email using the store's SMTP server

```
AUTHORIZATIONS:
                       API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS
→ storeId
                       string (StoreId)
   required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
REQUEST BODY SCHEMA:
                       application/json
 ⊣ email
                       string
                       Email of the recipient
→ subject
                       string
                       Subject of the email
```

⊣ body

string

Body of the email to send as plain text.

```
200 The email was sent (scheduled) successfully
400 The store's SMTP is not configured
403 If you are authenticated but forbidden to add new stores
404 The store was not found
```

```
Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

{
    "email": "string",
    "subject": "string",
    "body": "string"
}
```

# Store (Payment Methods)

Store Payment Methods operations

## Get store payment methods

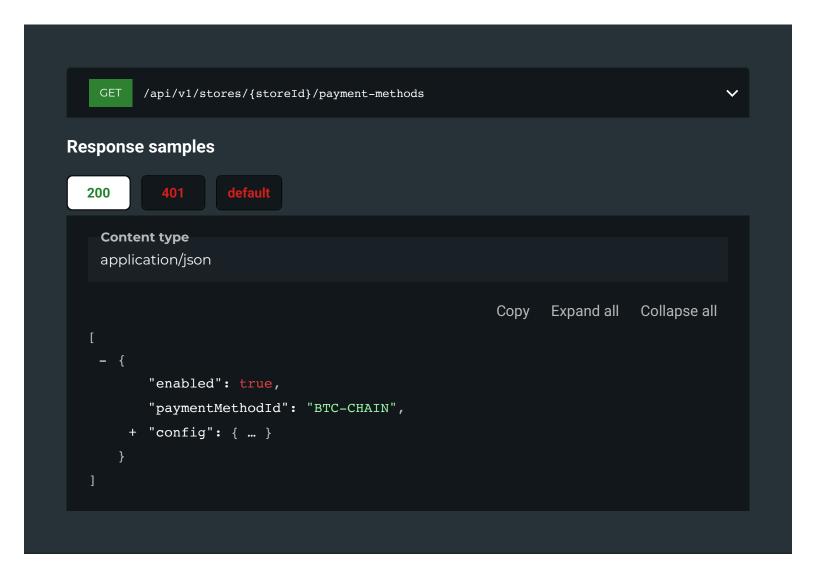
View information about the stores' configured payment methods

AUTHORIZATIONS:	API_Key( btcpay.store.canviewstoresettings) AND Basic
PATH PARAMETERS	
→ storeId required	string (StoreId) Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID
QUERY PARAMETERS	
→ onlyEnabled	boolean Fetch payment methods that are enabled/disabled only
⊣ includeConfig	boolean  Fetch the config of the payment methods, if true, the permission  btcpay.store.canmodifystoresettings is required.

## Responses

> 200 List of payment methods

> 401 Missing authorization
> default Unexpected error

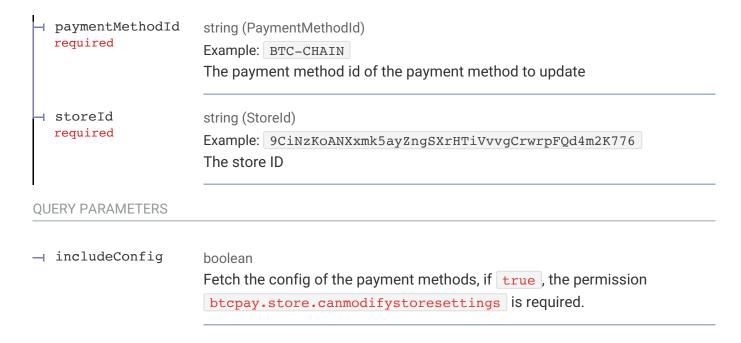


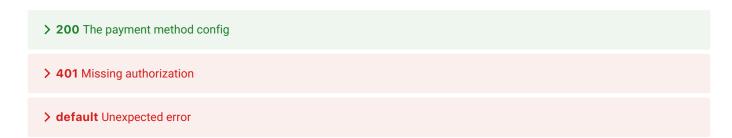
## Get store payment method

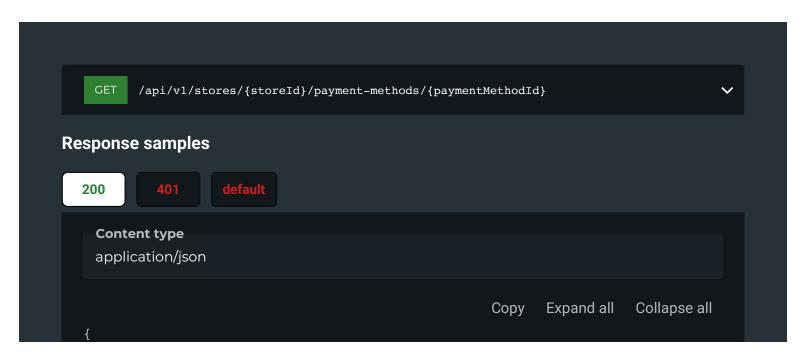
View information about the stores' configured payment method

```
AVI_Key ( btcpay.store.canviewstoresettings ) AND Basic

PATH PARAMETERS
```







```
"enabled": true,
    "paymentMethodId": "BTC-CHAIN",
- "config": {
        "useBech32Scheme": true,
        "lud12Enabled": true
}
```

## Update store's payment method

Update information about the stores' configured payment method

```
AUTHORIZATIONS:
                        API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS
 → paymentMethodId
                        string (PaymentMethodId)
   required
                        Example: BTC-CHAIN
                        The payment method id of the payment method to update
 ⊣ storeId
                        string (StoreId)
   required
                        Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                        The store ID
REQUEST BODY SCHEMA:
                        application/json
 - enabled
                        boolean Nullable
                        Default: null
                        Whether the payment method is enabled, leave null or unspecified to not
                        change current setting
- config >
                        LNURLPayPaymentMethodBaseData (object) or
                        LightningNetworkPaymentMethodBaseData (object) or
                        OnChainPaymentMethodBaseData (object) or object or Derivation scheme (object)
```

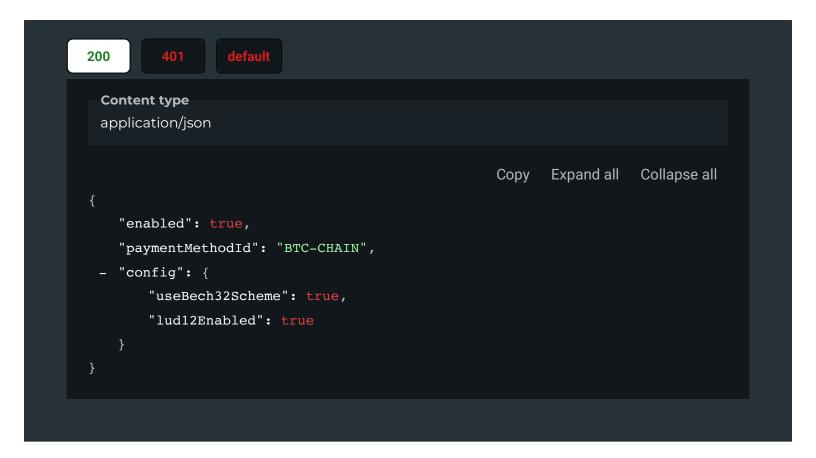
or Lightning connection string (object) or Output descriptor (object) Nullable

Default: null

The new payment method config, leave null or unspecified to not change current setting

```
    200 The payment method config
    401 Missing authorization
    default Unexpected error
```

```
/api/v1/stores/{storeId}/payment-methods/{paymentMethodId}
Request samples
   Payload
    Content type
    application/json
                                                                            Collapse all
                                                                Expand all
                                                         Copy
       "enabled": true,
    - "config": {
           "useBech32Scheme": true,
          "lud12Enabled": true
Response samples
```

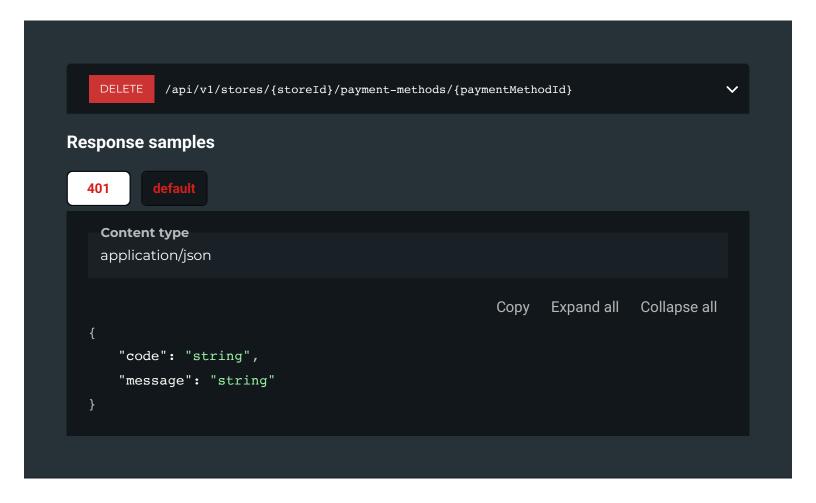


## Delete store's payment method

Delete information about the stores' configured payment method

#### Responses

```
200 The configuration got deleted
401 Missing authorization
default Unexpected error
```



# Stores (Rates Config)

Store Rates Config operations

## Get store rate settings

View rate settings of the specified store

#### Responses

200 specified store rate settings
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store



```
{
    "spread": "string",
    "preferredSource": "string",
    "isCustomScript": true,
    "effectiveScript": "string"
}
```

## Update store rate settings

Update a store's rate settings

```
AUTHORIZATIONS:
                         API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS

→ storeId

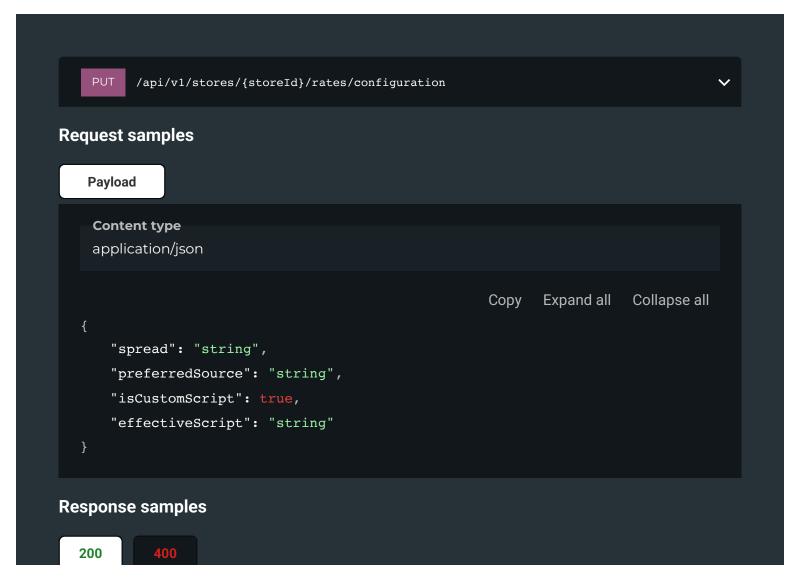
                         string (StoreId)
   required
                         Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                         The store ID
REQUEST BODY SCHEMA:
                          application/json
 - spread
                         string
                         A spread applies to the rate fetched in \( \). Must be \( >= 0 \) or \( <= 100 \)
→ preferredSource
                         string
                         If isCustomerScript is false affect use this source in the default's
                         effectiveScript , if isCustomerScript is false , this setting is set
                         to null. (See /misc/rate-sources for the available sources)

→ isCustomScript

                         boolean
                         Whether to use <a href="mailto:preferredSource">preferredSource</a> with default script or a custom
                          effectiveScript.
 → effectiveScript
                         string
```

When <u>isCustomScript</u> is <u>true</u>, this represent the custom script used to calculate a currency pair's exchange rate. Else, it represent the script generated by the default rules and <u>preferredSource</u>.

```
    200 The settings were updated
    400 A list of errors that occurred when updating the settings
    403 If you are authenticated but forbidden to modify the store
```



```
Content type
application/json

Copy Expand all Collapse all

{
    "spread": "string",
    "preferredSource": "string",
    "isCustomScript": true,
    "effectiveScript": "string"
}
```

## Preview rate configuration results

Preview rate configuration results before you set it on the store

```
API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS

→ storeId

                        string (StoreId)
   required
                        Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                        The store ID
QUERY PARAMETERS
→ currencyPair
                        Array of strings Nullable
                        The currency pairs to preview
REQUEST BODY SCHEMA:
                         application/json

→ spread

                        string
                        A spread applies to the rate fetched in \( \). Must be \( >= 0 \) or \( <= 100 \)
```

```
If isCustomerScript is false affect use this source in the default's

effectiveScript, if isCustomerScript is false, this setting is set

to null. (See /misc/rate-sources for the available sources)

boolean

Whether to use preferredSource with default script or a custom

effectiveScript.

effectiveScript

String

When isCustomScript is true, this represent the custom script used to

calculate a currency pair's exchange rate. Else, it represent the script

generated by the default rules and preferredSource.
```

#### Responses

200 The settings were executed and a preview was returned
 400 A list of errors that occurred when previewing the settings
 403 If you are authenticated but forbidden to modify the store



```
"spread": "string",
      "preferredSource": "string",
      "isCustomScript": true,
      "effectiveScript": "string"
Response samples
  200
    Content type
    application/json
                                                      Copy
                                                             Expand all
                                                                        Collapse all
          "currencyPair": "BTC_USD",
       + "errors": [ ... ],
          "rate": "64392.23"
```

## Stores (Rates)

Store Rates operations

### Get rates

#### Get rates on the store

API\_Key ( btcpay.store.canviewstoresettings ) AND Basic

PATH PARAMETERS

I storeId string (StoreId) Example: 9CinzkoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID

QUERY PARAMETERS

Array of strings Nullable Example: currencyPair=BTC\_USD&currencyPair=BTC\_EUR
The currency pairs to fetch rates for

#### Responses

200 The settings were executed and a preview was returned
 400 A list of errors that occurred when previewing the settings
 403 If you are authenticated but forbidden to modify the store



```
Copy Expand all Collapse all

[
- {
    "currencyPair": "BTC_USD",
    + "errors": [ ... ],
    "rate": "64392.23"
    }
]
```

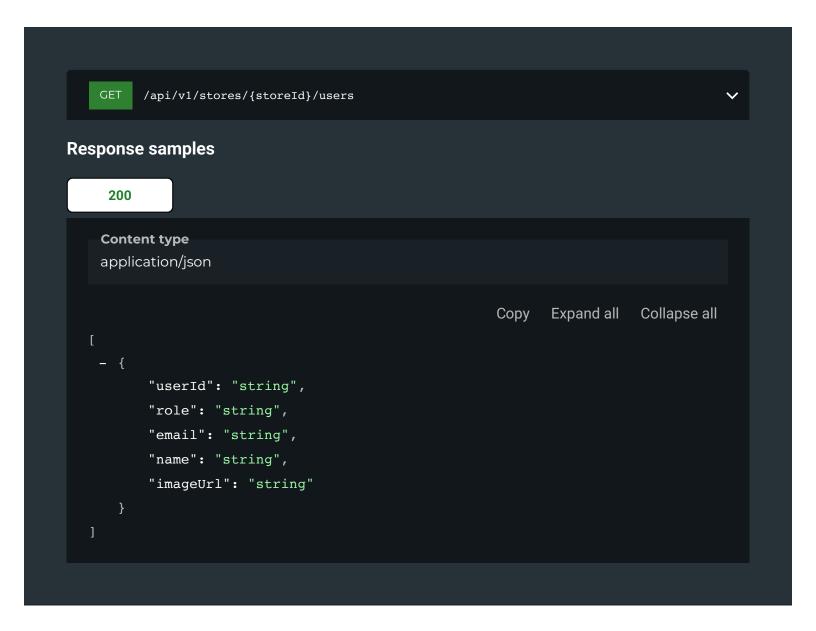
## Stores (Users)

Store Users operations

## Get store users

View users of the specified store

```
    200 specified store users
    403 If you are authenticated but forbidden to view the specified store's users
    404 The store could not be found
```



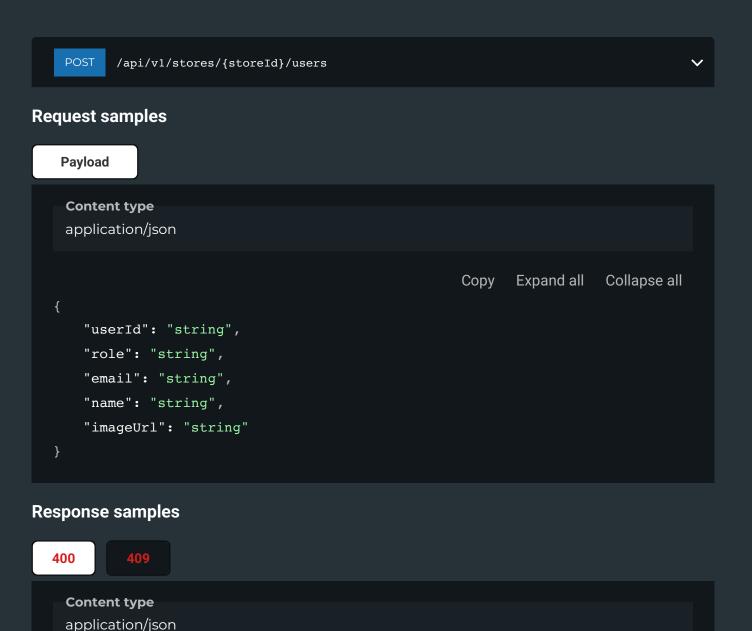
## Add a store user

Add a store user

API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic **AUTHORIZATIONS: PATH PARAMETERS** → storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json userId string The id of the user ⊣ role The role of the user. Default roles are Owner, Manager, Employee and Guest ⊣ email string Nullable The email of the user ⊣ name string Nullable The name of the user ⊣ imageUrl string Nullable The profile picture URL of the user

#### Responses

200 The user was added
 400 A list of errors that occurred when adding the store user
 403 If you are authenticated but forbidden to add new store users
 404 The store or user could not be found
 409 Error code: duplicate-store-user-role. Removing this user would result in the store having no owner.



```
Content type
application/json

Copy Expand all Collapse all

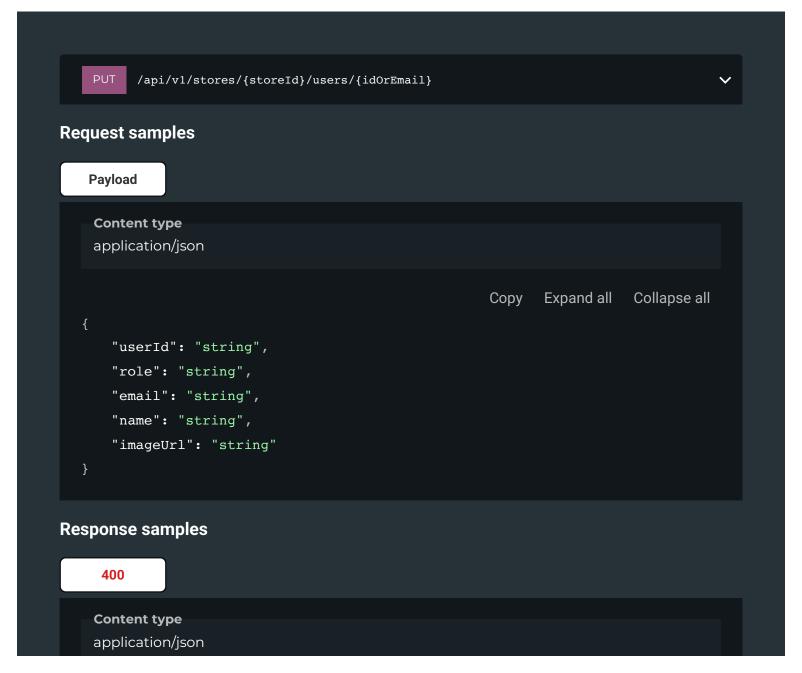
[
- {
    "path": "string",
    "message": "string"
}
]
```

## Updates a store user

Updates a store user

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS	
⊣ storeId required	string (StoreId)  Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776  The store ID
⊣ idOrEmail required	string The user's id or email
REQUEST BODY SCHEMA:	application/json
-ı userId	string The id of the user
⊣ role	string The role of the user. Default roles are Owner, Manager, Employee and Guest
—ı email	string Nullable The email of the user
—ı name	string Nullable The name of the user
⊣ imageUrl	string Nullable The profile picture URL of the user

```
200 The user was updated
400 A list of errors that occurred when updating the store user
403 If you are authenticated but forbidden to update store users
404 The store or user could not be found
```



```
Copy Expand all Collapse all

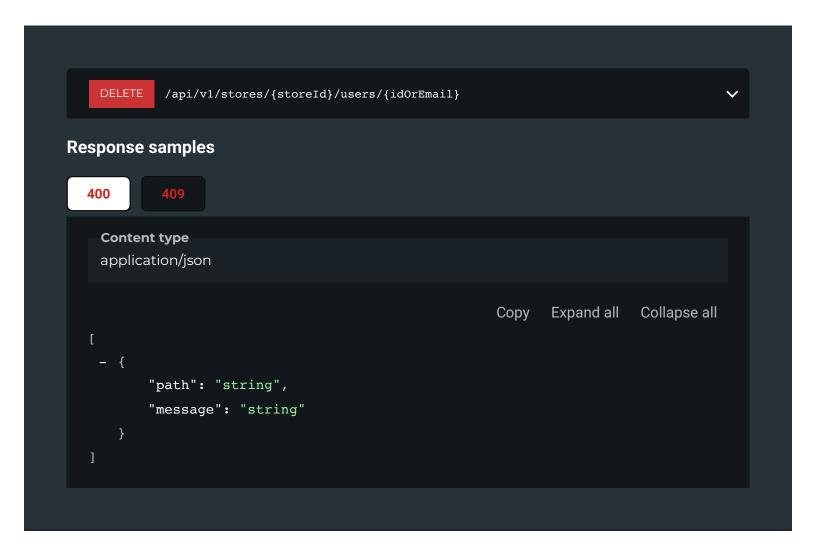
[
- {
    "path": "string",
    "message": "string"
    }
]
```

#### Remove Store User

Removes the specified store user. If there is no other owner, this endpoint will fail.

```
    200 The user has been removed
    400 A list of errors that occurred when removing the store user
    403 If you are authenticated but forbidden to remove the specified store user
    404 The store or user could not be found.
```

> 409 Error code: store-user-role-orphaned. Removing this user would result in the store having no owner.



# Store Wallet (On Chain)

Store Wallet (On Chain) operations

## Get store on-chain wallet overview

#### View information about the specified wallet

API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic

PATH PARAMETERS

H paymentMethodId required string (PaymentMethodId)
Example: BTC-CHAIN
The payment method id of the payment method to update

H storeId required string (StoreId)
Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID

#### Responses

200 specified wallet
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store/wallet



```
Copy Expand all Collapse all

{
    "balance": "string",
    "unconfirmedBalance": "string",
    "confirmedBalance": "string"
}
```

## Get store on-chain wallet balance histogram

View the balance histogram of the specified wallet

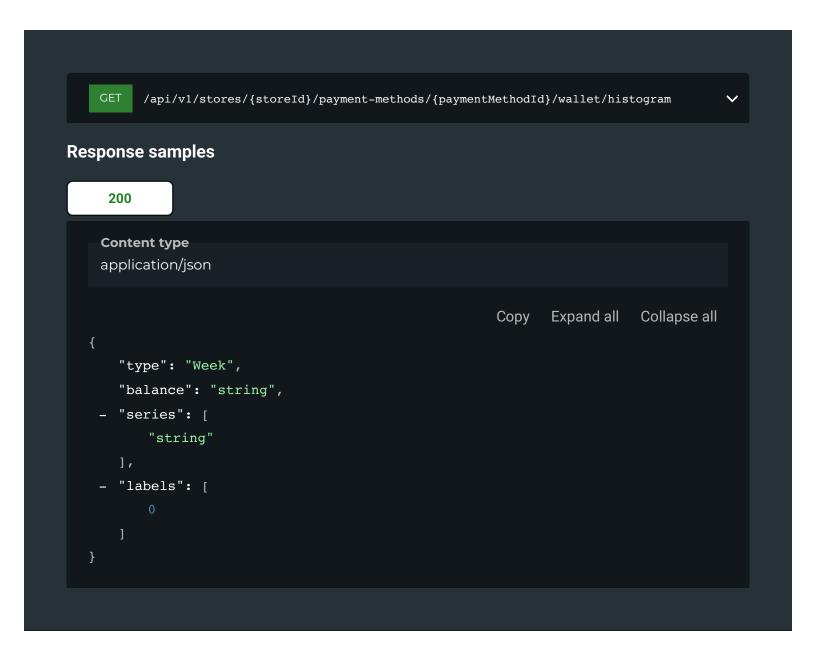
```
API_Key ( btcpay.store.canmodifystoresettings ) AND Basic

PATH PARAMETERS

I paymentMethodId required string (PaymentMethodId)
Example: BTC-CHAIN
The payment method id of the payment method to update

I storeId string (StoreId)
Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID
```

```
    200 specified wallet
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store/wallet
```



## Get store on-chain wallet fee rate

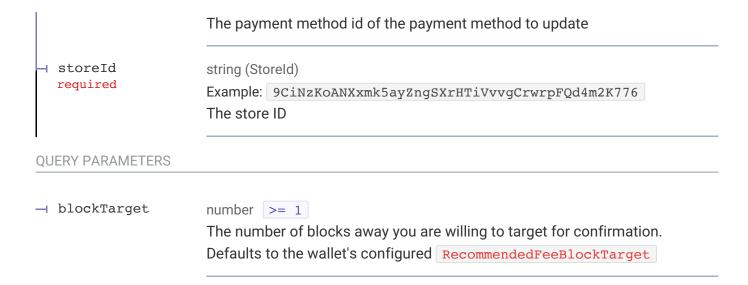
Get wallet onchain fee rate

```
AVITHORIZATIONS:

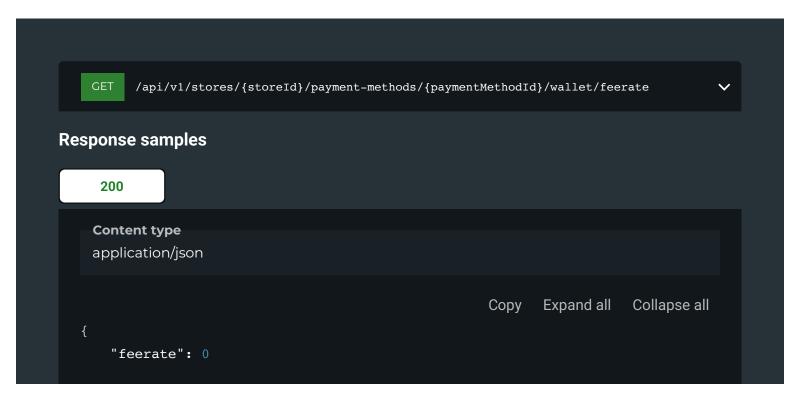
PATH PARAMETERS

- paymentMethodId required string (PaymentMethodId)

Example: BTC-CHAIN
```



```
    200 fee rate
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store/wallet
```



.

#### Get store on-chain wallet address

was not used

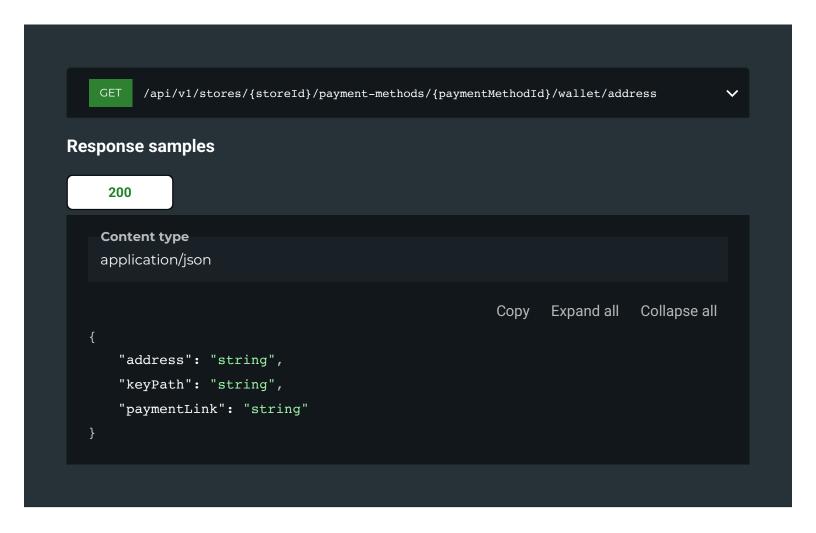
Get or generate address for wallet

**AUTHORIZATIONS:** API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic PATH PARAMETERS → paymentMethodId string (PaymentMethodId) required Example: BTC-CHAIN The payment method id of the payment method to update storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID **QUERY PARAMETERS** → forceGenerate boolean Default: false

Whether to generate a new address for this request even if the previous one

## Responses

200 reserved address
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store/wallet



### UnReserve last store on-chain wallet address

#### UnReserve address

## Responses

200 address unreserved
403 If you are authenticated but forbidden to view the specified store
404 The key is not found for this store/wallet or there was no address reserved



#### Get store on-chain wallet transactions

Get store on-chain wallet transactions

**AUTHORIZATIONS:** API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic PATH PARAMETERS → paymentMethodId string (PaymentMethodId) required Example: BTC-CHAIN The payment method id of the payment method to update → storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID **QUERY PARAMETERS** ⊣ labelFilter string Example: labelFilter=invoice Transaction label to filter by → limit integer Maximum number of transactions to return ⊣ skip integer Number of transactions to skip from the start ⊣ statusFilter Array of strings (TransactionStatus) Items Enum: "Confirmed" "Unconfirmed" Statuses to filter the transactions with

### Responses

> 200 transactions list

— 403 If you are authenticated but forbidden to view the specified store

- 404 The key is not found for this store/wallet
  503 You need to allow non-admins to use hotwallets for their stores (in /server/policies)
- **Response samples** 200 **Content type** application/json Collapse all Expand all Copy "transactionHash": "string", "comment": "string", "amount": "string", "blockHash": "string", "blockHeight": "string", "confirmations": "string", "timestamp": 1592312018, "status": "Confirmed", + "labels": [ ... ]

## Create store on-chain wallet transaction

#### Create store on-chain wallet transaction

**AUTHORIZATIONS:** API\_Key (| btcpay.store.canmodifystoresettings | ) AND Basic PATH PARAMETERS paymentMethodId string (PaymentMethodId) required Example: BTC-CHAIN The payment method id of the payment method to update ⊣ storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json destinations > Array of objects (CreateOnChainTransactionRequestDestination) What and where to send money ⊣ feerate number <decimal or long (sats/byte)> Transaction fee. → proceedWithPayjoin boolean Nullable Default: true Whether to attempt to do a BIP78 payjoin if one of the destinations is a BIP21 with payjoin enabled proceedWithBroadcast boolean Nullable Default: true Whether to broadcast the transaction after creating it or to simply return the transaction in hex format. noChange boolean Nullable Default: false Whether to send all the spent coins to the destinations (THIS CAN COST YOU SIGNIFICANT AMOUNTS OF MONEY, LEAVE FALSE UNLESS YOU KNOW WHAT YOU ARE DOING). rbf

boolean Nullable
Whether to enable RBF for the transaction. Leave blank to have it random (beneficial to privacy)

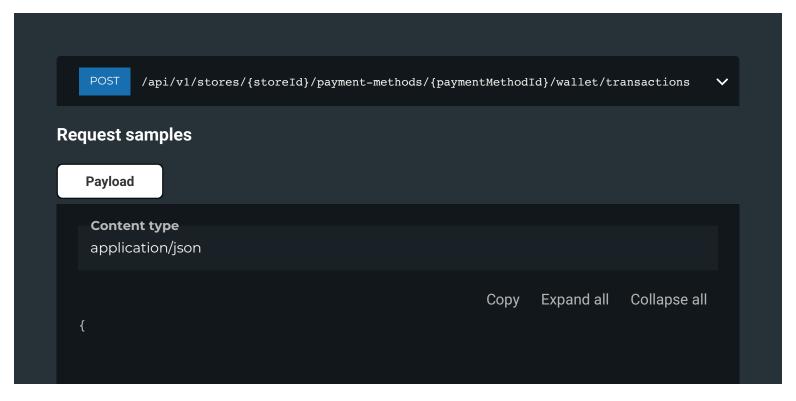
boolean Nullable
Default: false
Whether to exclude unconfirmed UTXOs from the transaction.

reselectedInputs

Array of strings Nullable
Restrict the creation of the transactions from the outpoints provided ONLY (coin selection)

#### Responses

200 the tx
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store/wallet



#### **Response samples**

200

## Get store on-chain wallet transaction

Get store on-chain wallet transaction

AUTHORIZATIONS: API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic

**PATH PARAMETERS** 

The payment method id of the payment method to update

→ storeId string (StoreId)

required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

transactionId string

required

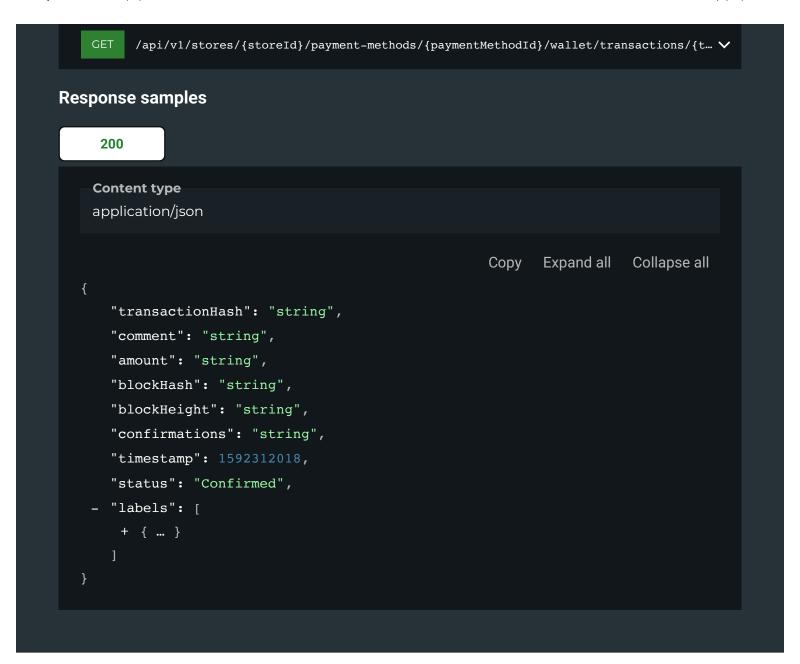
The transaction id to fetch

### Responses

> 200 transaction

- 403 If you are authenticated but forbidden to view the specified store

— 404 The key is not found for this store/wallet



## Patch store on-chain wallet transaction info

Patch store on-chain wallet transaction info

required Example: BTC-CHAIN The payment method id of the payment method to update → storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID → transactionId string required The transaction id to fetch **QUERY PARAMETERS** - force string Whether to update the label/comments even if the transaction does not yet exist REQUEST BODY SCHEMA: application/json

- comment string Nullable

Transaction comment

→ <del>labels</del> Array of strings Nullable

Deprecated

Transaction labels

## Responses

> 200 transaction

— 403 If you are authenticated but forbidden to view the specified store

— 404 The key is not found for this store/wallet

/api/vi/stores/{storeid}/payment-methods/{paymentMethodid}/wallet/transactions... 🗸

## **Request samples**

#### **Payload**

```
Content type
application/json

Copy Expand all Collapse all

{
    "comment": "string",
    - "labels": [
        "string"
    ]
}
```

#### **Response samples**

200

## Get store on-chain wallet UTXOS

Get store on-chain wallet utxos

## Responses

200 utxo list
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store/wallet



## Generate store on-chain wallet

Generate a wallet and update the specified store's payment method to it

```
API_Key (| btcpay.store.canmodifystoresettings ) AND Basic

PATH PARAMETERS

| paymentMethodId required | string (PaymentMethodId) |
| Example: BTC-CHAIN |
| The payment method id of the payment method to update |
| storeId | string (StoreId) |
| required | string (StoreId) |
```

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json ⊣ label string A label that will be shown in the UI → existingMnemonic string (Mnemonic) A BIP39 mnemonic → passphrase string A passphrase for the BIP39 mnemonic seed accountNumber number Default: 0 The account to derive from the BIP39 mnemonic seed → savePrivateKeys boolean Default: false Whether to store the seed inside BTCPay Server to enable some additional services. IF false AND existingMnemonic IS NOT SPECIFIED, BE SURE TO SECURELY STORE THE SEED IN THE RESPONSE! → importKeysToRPC boolean Default: false Whether to import all addresses generated via BTCPay Server into the underlying node wallet. (Private keys will also be imported if savePrivateKeys is set to true. ⊣ wordList string Default: "English" Enum: "ChineseSimplified" "ChineseTraditional" "Czech" "English" "French" "Japanese" "PortugueseBrazil" "Spanish" If existingMnemonic is not set, a mnemonic is generated using the specified wordList. → wordCount number Default: 12

```
Enum: 12 15 18 21 24

If existingMnemonic is not set, a mnemonic is generated using the specified wordCount.

I scriptPubKeyType string

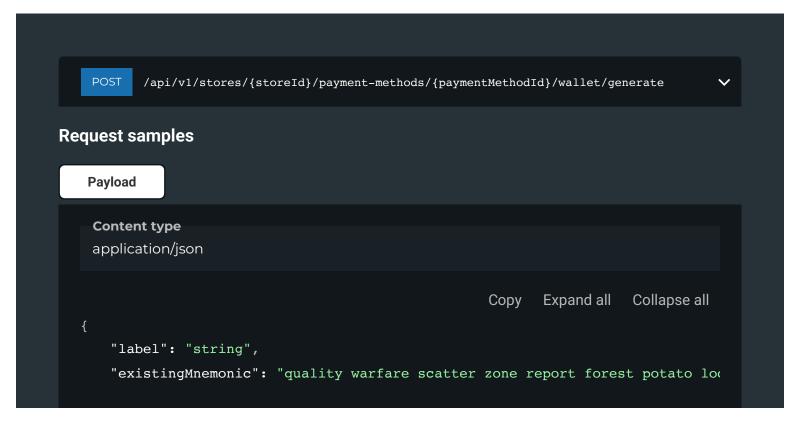
Default: "Segwit"

Enum: "Legacy" "Segwit" "SegwitP2SH" "TaprootBIP86"

the type of wallet to generate
```

## Responses

```
    200 updated specified payment method with the generated wallet
    400 A list of errors that occurred when updating the store payment method
    403 If you are authenticated but forbidden to update the specified store
    404 The key is not found for this store
```



```
'passphrase": "string",
      "accountNumber": 0,
      "savePrivateKeys": false,
      "importKeysToRPC": false,
      "wordList": "English",
      "wordCount": 12,
      "scriptPubKeyType": "Segwit"
Response samples
  200
    Content type
    application/json
                                                            Expand all
                                                                        Collapse all
                                                      Copy
      "enabled": true,
      "paymentMethodId": "BTC-CHAIN",
    - "config": {
          "useBech32Scheme": true,
          "lud12Enabled": true,
          "derivationScheme": "xpub...",
          "label": "string",
          "accountKeyPath": "abcd82a1/84'/0'/0'"
      },
      "mnemonic": "quality warfare scatter zone report forest potato local swing
```

# Preview store on-chain payment method addresses

View addresses of the current payment method of the store

API\_Key ( btcpay.store.canviewstoresettings ) AND Basic **AUTHORIZATIONS:** PATH PARAMETERS ⊣ storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID paymentMethodId string required Example: BTC-CHAIN The payment method id of the payment method to update **QUERY PARAMETERS** ⊣ offset number From which index to fetch the addresses - count number Number of addresses to preview

## Responses

- 200 specified payment method addresses
   403 If you are authenticated but forbidden to view the specified store
   404 The key is not found for this store/payment method
- GET /api/v1/stores/{storeId}/payment-methods/{paymentMethodId}/wallet/preview 
  Response samples

```
Content type
application/json

Copy Expand all Collapse all

{
    - "addresses": [
    + { ... }
    ]
}
```

# Preview proposed store on-chain payment method addresses

View addresses of a proposed payment method of the store

```
API_Key ( btcpay.store.canviewstoresettings ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
→ paymentMethodId
                      string (PaymentMethodId)
  required
                      Example: BTC-CHAIN
                      The payment method id of the payment method to update
⊣ storeId
                      string (StoreId)
  required
                      Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                      The store ID
OUERY PARAMETERS
⊣ offset
                      number
                      From which index to fetch the addresses
```

Number of addresses to preview

REQUEST BODY SCHEMA: application/json

I derivationScheme string
The derivation scheme

## Responses

200 specified payment method addresses
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store

```
POST /api/v1/stores/(storeId)/payment-methods/(paymentMethodId)/wallet/preview 

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

{
  "derivationScheme": "xpub..."
}

Response samples
```

```
Content type
application/json

Copy Expand all Collapse all

{
- "addresses": [
    + { ... }
    ]
}
```

# Get store on-chain wallet objects

View wallet objects

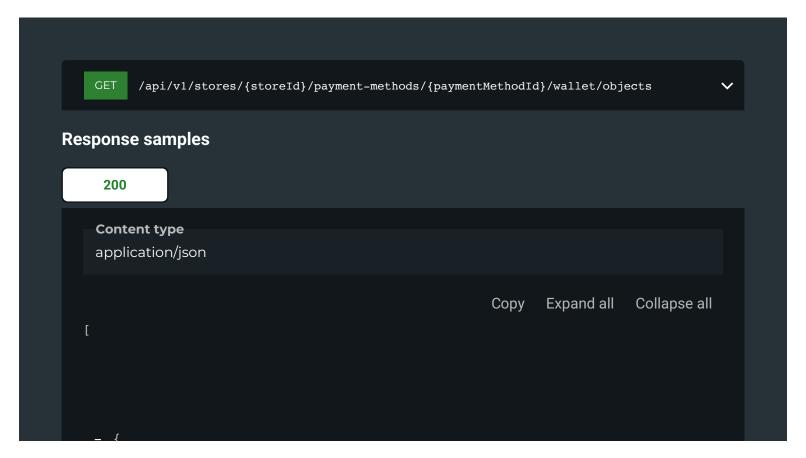
```
API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
 ⊣ paymentMethodId
                       string (PaymentMethodId)
   required
                       Example: BTC-CHAIN
                       The payment method id of the payment method to update
 ⊣ storeId
                       string (StoreId)
   required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
QUERY PARAMETERS
⊣ ids
                           Array of strings
                           Example: ids=03abcde...
                           The ids of objects to fetch, if used, type should be specified
```

The type of object to fetch

I includeNeighbourData boolean
Default: true
Whether or not you should include neighbour's node data in the result (ie, links.objectData)

#### Responses

200 Selected objects and their links
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store/wallet



```
"data": { },

+ "links": [ ... ],

"type": "string",

"id": "string"

}
```

# Add/Update store on-chain wallet objects

Add/Update wallet objects

```
API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
→ paymentMethodId
                       string (PaymentMethodId)
  required
                       Example: BTC-CHAIN
                       The payment method id of the payment method to update
                       string (StoreId)

→ storeId

  required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
REQUEST BODY SCHEMA:
                        application/json
⊣ data>
                       object
⊣ links >
                       Array of objects (OnChainWalletObjectLink) Nullable
                       Links of this object
⊣ type
                       string
                       The type of wallet object
  id
                       string
```

The identifier of the wallet object (unique per type, per wallet)

## Responses

- > 200 Wallet object's data and its links
- 403 If you are authenticated but forbidden to view the specified store
- 404 The key is not found for this store/wallet

```
POST
           /api/v1/stores/{storeId}/payment-methods/{paymentMethodId}/wallet/objects
Request samples
    Payload
    Content type
    application/json
                                                                 Expand all
                                                                             Collapse all
                                                          Copy
       "data": { },
     - "links": [
       ],
       "type": "string",
       "id": "string"
Response samples
```

# Get store on-chain wallet object

View wallet object

```
AUTHORIZATIONS:
                       API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS
→ paymentMethodId
                       string (PaymentMethodId)
  required
                       Example: BTC-CHAIN
                       The payment method id of the payment method to update
 ⊣ storeId
                       string (StoreId)
   required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
→ objectId
                       string
   required
                       Example: abc392...
                       The object id to fetch
```

d objectType
 required

string

Example: tx

The object type to fetch

**QUERY PARAMETERS** 

→ includeNeighbourData boolean

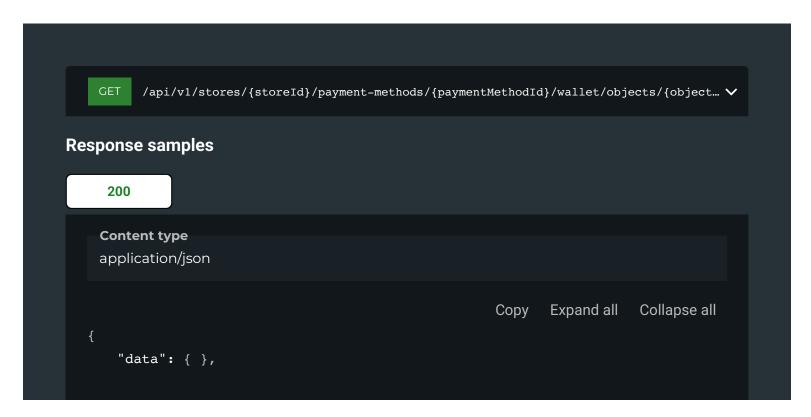
Default: true

Whether or not you should include neighbour's node data in the result (ie,

links.objectData)

## Responses

- > 200 Wallet object's data and its links
- 403 If you are authenticated but forbidden to view the specified store
- 404 The key is not found for this store/wallet



# Remove store on-chain wallet objects

Remove wallet object

```
API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
 → paymentMethodId
                       string (PaymentMethodId)
  required
                       Example: BTC-CHAIN
                       The payment method id of the payment method to update
⊣ storeId
                       string (StoreId)
  required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
→ objectId
                       string
  required
                       Example: abc392...
                       The object id to fetch
→ objectType
                       string
  required
                       Example: tx
                       The object type to fetch
```

## Responses

200 successful removal of filtered object
403 If you are authenticated but forbidden to view the specified store
404 The key is not found for this store/wallet

DELETE /api/v1/stores/{storeId}/payment-methods/{paymentMethodId}/wallet/objects/{obj... >

# Add/Update store on-chain wallet object link

Add/Update wallet object link

**AUTHORIZATIONS:** API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic PATH PARAMETERS paymentMethodId string (PaymentMethodId) required Example: BTC-CHAIN The payment method id of the payment method to update ⊣ storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID → objectId string required Example: abc392... The object id to fetch → objectType string required Example: tx The object type to fetch

REQUEST BODY SCHEMA: application/json

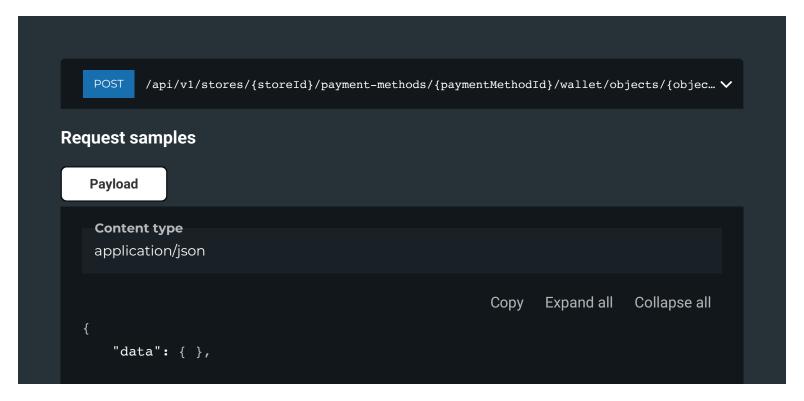
object
The data of the link

type string
The type of wallet object

id string
The identifier of the wallet object (unique per type, per wallet)

## Responses

200 action completed
403 If you are authenticated but forbidden to view the specified store
404 The key is not found for this store/wallet



```
"type": "string",

"id": "string"
}
```

# Remove store on-chain wallet object links

Remove wallet object link

```
AUTHORIZATIONS:
                       API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS
→ paymentMethodId
                       string (PaymentMethodId)
  required
                       Example: BTC-CHAIN
                       The payment method id of the payment method to update
                       string (StoreId)
⊣ storeId
   required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
⊣ linkId
                       string
   required
                       Example: abc392...
                       The object id of the linked neighbour
→ objectId
                       string
   required
                       Example: abc392...
                       The object id to fetch
→ linkType
                       string
   required
                       Example: tx
                       The object type of the linked neighbour
→ objectType
                       string
  required
                       Example: tx
                       The object type to fetch
```

Responses

- 200 successful removal of filtered object link
- 403 If you are authenticated but forbidden to view the specified store
- 404 The key is not found for this store/wallet

DELETE

/api/v1/stores/{storeId}/payment-methods/{paymentMethodId}/wallet/objects/{obj... 🗸

# Stores

Store operations

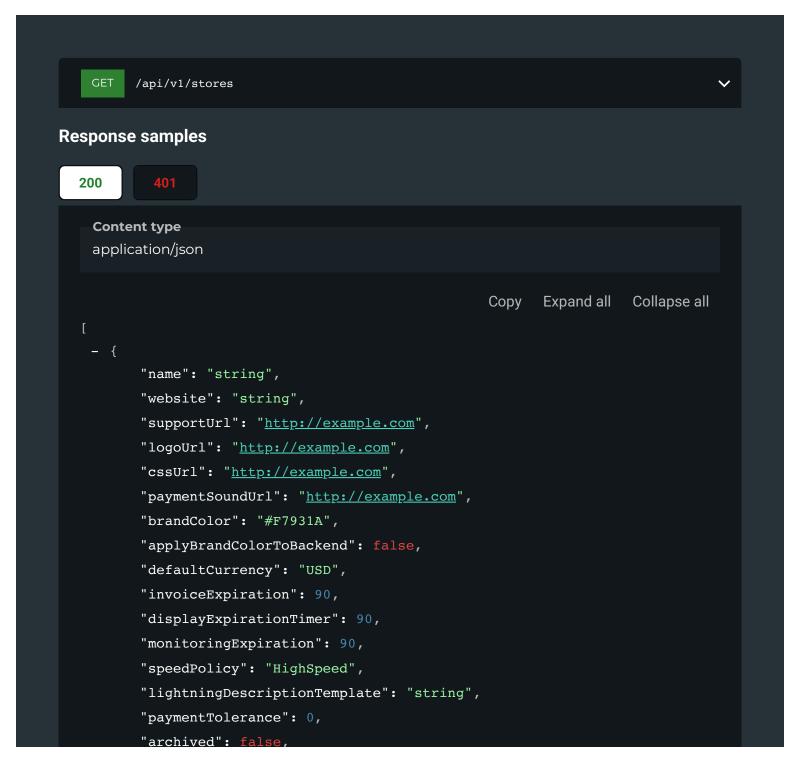
## Get stores

View information about the available stores

AUTHORIZATIONS: API\_Key ( btcpay.store.canviewstoresettings ) AND Basic

#### Responses

```
> 200 list of stores
> 401 Missing authorization
```



```
"anyoneCanCreateInvoice": false,
+ "receipt": { ... },
  "lightningAmountInSatoshi": false,
  "lightningPrivateRouteHints": false,
  "onChainWithLnInvoiceFallback": false,
  "redirectAutomatically": false,
  "showRecommendedFee": true,
  "recommendedFeeBlockTarget": 1,
  "defaultLang": "en",
  "htmlTitle": "string",
  "networkFeeMode": "Always",
  "payJoinEnabled": false,
  "autoDetectLanguage": false,
  "showPayInWalletButton": true,
  "showStoreHeader": true,
  "celebratePayment": true,
  "playSoundOnPayment": false,
  "lazyPaymentMethods": false,
  "defaultPaymentMethod": "BTC-CHAIN",
  "paymentMethodCriteria": { },
  "id": "string"
```

## Create a new store

Create a new store

AUTHORIZATIONS: API\_Key (| btcpay.store.canmodifystoresettings ) AND Basic

REQUEST BODY SCHEMA: application/json

→ name string

	The name of the store
→ website	string <url> Nullable</url>
	The absolute url of the store
⊣ supportUrl	string <uri> Nullable</uri>
	The support URI of the store, can contain the placeholders
	{OrderId} and {InvoiceId}. Can be any valid URI, such as a website, email, and nostr.
logottyl	
⊢ logoUrl	string <uri>Nullable Absolute URL to a logo file or a reference to an uploaded file id</uri>
	with fileid: ID
- cssUrl	string <uri> Nullable</uri>
	Absolute URL to CSS file to customize the public/customer-
	facing pages of the store. (Invoice, Payment Request, Pull
	Payment, etc.) or a reference to an uploaded file id with
	fileid:ID
→ paymentSoundUrl	string <uri> Nullable</uri>
	Absolute URL to a sound file or a reference to an uploaded file
	id with fileid:ID
⊣ brandColor	string Nullable
	The brand color of the store in HEX format
→ applyBrandColorToBackend	boolean
	Default: false
	Apply the brand color to the store's backend as well
→ defaultCurrency	string
	Default: "USD"
	The default currency of the store
→ invoiceExpiration	number < seconds [ 60 2073600 ]
	Default: 900
	The time after which an invoice is considered expired if not
	paid. The value will be rounded down to a minute.

→ displayExpirationTimer	number <seconds> [ 60 2073600 ]  Default: 300  The time left that will trigger the countdown timer on the checkout page to be shown. The value will be rounded down to a minute.</seconds>
→ monitoringExpiration	number <seconds> [ 600 2073600 ]  Default: 3600  The time after which an invoice which has been paid but not confirmed will be considered invalid. The value will be rounded down to a minute.</seconds>
⊣ speedPolicy	string (SpeedPolicy) Enum: "HighSpeed" "LowMediumSpeed" "LowSpeed"  "MediumSpeed"  This is a risk mitigation parameter for the merchant to configure how they want to fulfill orders depending on the number of block confirmations for the transaction made by the consumer on the selected cryptocurrency. "HighSpeed":  0 confirmations (1 confirmation if RBF enabled in transaction)  "MediumSpeed": 1 confirmation  "LowMediumSpeed": 2 confirmations  "LowSpeed": 6 confirmations
→ lightningDescriptionTemplate	string Nullable The BOLT11 description of the lightning invoice in the checkout. You can use placeholders '{StoreName}', '{ItemDescription}' and '{OrderId}'.
→ paymentTolerance	number <double> [ 0 100 ]  Default: 0  Consider an invoice fully paid, even if the payment is missing 'x' % of the full amount.</double>
⊣ archived	boolean  Default: false  If true, the store does not appear in the stores list by default.
⊣ anyoneCanCreateInvoice	boolean Default: false

	If true, then no authentication is needed to create invoices on this store.
⊣ receipt >	object Nullable Additional settings to customize the public receipt
⊣ lightningAmountInSatoshi	boolean  Default: false  If true, lightning payment methods show amount in satoshi in the checkout page.
⊣ lightningPrivateRouteHints	boolean  Default: false  Should private route hints be included in the lightning payment of the checkout page.
→ onChainWithLnInvoiceFallback	boolean  Default: false  Unify on-chain and lightning payment URL.
⊣ redirectAutomatically	boolean  Default: false  After successfull payment, should the checkout page redirect the user automatically to the redirect URL of the invoice?
→ showRecommendedFee	boolean Default: true
→ recommendedFeeBlockTarget	integer <int32> Default: 1 The fee rate recommendation in the checkout page for the onchain payment to be confirmed after 'x' blocks.</int32>
⊣ defaultLang	string  Default: "en"  The default language to use in the checkout page. (The different translations available are listed here
⊣ htmlTitle	string Nullable The HTML title of the checkout page (when you over the tab in

	your browser)
⊣ networkFeeMode	string (NetworkFeeMode)
	Enum: "Always" "MultiplePaymentsOnly" "Never"
	Check whether network fee should be added to the invoice if
	on-chain payment is used. (More information)
→ payJoinEnabled	boolean
	Default: false
	If true, payjoin will be proposed in the checkout page if
	possible. (More information)
→ autoDetectLanguage	boolean
	Default: false
	If true, the language on the checkout page will adapt to the
	language defined by the user's browser settings
	——————————————————————————————————————
→ showPayInWalletButton	boolean
	Default: true
	If true, the "Pay in wallet" button will be shown on the checkout
	page (Checkout V2)
⊣ showStoreHeader	boolean
	Default: true
	If true, the store header will be shown on the checkout page
	(Checkout V2)
→ celebratePayment	boolean
	Default: true
	If true, payments on the checkout page will be celebrated with
	confetti (Checkout V2)
→ playSoundOnPayment	boolean
	Default: false
	If true, sounds on the checkout page will be enabled (Checkout
	V2)
→ lazyPaymentMethods	boolean
	Default: false
	If true, payment methods are enabled individually upon user
	apon door

interaction in the invoice

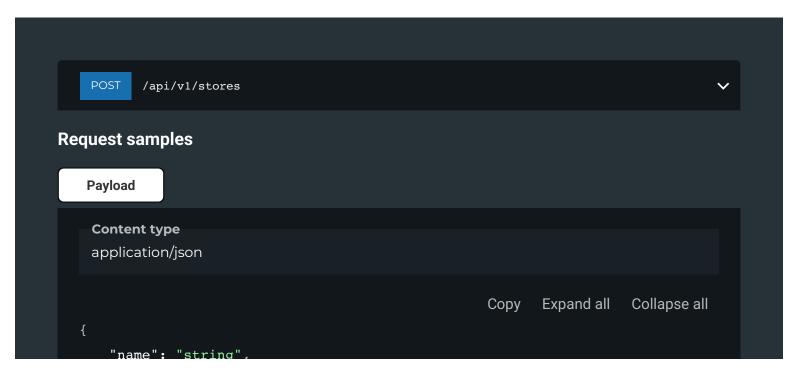
string (PaymentMethodId)
Payment method IDs. Available payment method IDs for Bitcoin are:

• "BTC-CHAIN": Onchain
• "BTC-LN": Lightning
• "BTC-LNURL": LNURL

paymentMethodCriteria object Nullable
The criteria required to activate specific payment methods.

## Responses

200 Information about the new store
 400 A list of errors that occurred when creating the store
 403 If you are authenticated but forbidden to add new stores



"website": "string", "supportUrl": "<a href="http://example.com"">http://example.com</a>", "logoUrl": "<a href="http://example.com"">http://example.com</a>", "cssUrl": "http://example.com", "paymentSoundUrl": "<a href="http://example.com"">http://example.com</a>", "brandColor": "#F7931A", "applyBrandColorToBackend": false, "defaultCurrency": "USD", "invoiceExpiration": 90, "displayExpirationTimer": 90, "monitoringExpiration": 90, "speedPolicy": "HighSpeed", "lightningDescriptionTemplate": "string", "paymentTolerance": 0, "archived": false, "anyoneCanCreateInvoice": false, - "receipt": { "enabled": true, "showQR": null, "showPayments": null }, "lightningAmountInSatoshi": false, "lightningPrivateRouteHints": false, "onChainWithLnInvoiceFallback": false, "redirectAutomatically": false, "showRecommendedFee": true, "recommendedFeeBlockTarget": 1, "defaultLang": "en", "htmlTitle": "string", "networkFeeMode": "Always", "payJoinEnabled": false, "autoDetectLanguage": false, "showPayInWalletButton": true, "showStoreHeader": true, "celebratePayment": true, "playSoundOnPayment": false, "lazvPavmentMethods": false

```
"defaultPaymentMethod": "BTC-CHAIN",
"paymentMethodCriteria": { }
```

#### Response samples

200

```
Content type
application/json
                                                         Copy
                                                                Expand all
                                                                             Collapse all
  "name": "string",
   "website": "string",
   "supportUrl": "<a href="http://example.com"">http://example.com</a>",
   "logoUrl": "http://example.com",
   "cssUrl": "<a href="http://example.com"">http://example.com</a>",
   "paymentSoundUrl": "<a href="http://example.com"">http://example.com</a>",
   "brandColor": "#F7931A",
   "applyBrandColorToBackend": false,
   "defaultCurrency": "USD",
   "invoiceExpiration": 90,
   "displayExpirationTimer": 90,
   "monitoringExpiration": 90,
   "speedPolicy": "HighSpeed",
   "lightningDescriptionTemplate": "string",
   "paymentTolerance": 0,
   "archived": false,
   "anyoneCanCreateInvoice": false,
- "receipt": {
       "enabled": true,
       "showQR": null,
       "showPayments": null
  },
   "lightningAmountInSatoshi": false,
  "lightningPrivateRouteHints": fal
```

```
"onChainWithLnInvoiceFallback": false,
"redirectAutomatically": false,
"showRecommendedFee": true,
"recommendedFeeBlockTarget": 1,
"defaultLang": "en",
"htmlTitle": "string",
"networkFeeMode": "Always",
"payJoinEnabled": false,
"autoDetectLanguage": false,
"showPayInWalletButton": true,
"showStoreHeader": true,
"celebratePayment": true,
"playSoundOnPayment": false,
"lazyPaymentMethods": false,
"defaultPaymentMethod": "BTC-CHAIN",
"paymentMethodCriteria": { },
"id": "string"
```

## Get store

View information about the specified store

AVITHORIZATIONS:

PATH PARAMETERS

API\_Key ( btcpay.store.canviewstoresettings ) AND Basic

PATH PARAMETERS

string (StoreId)

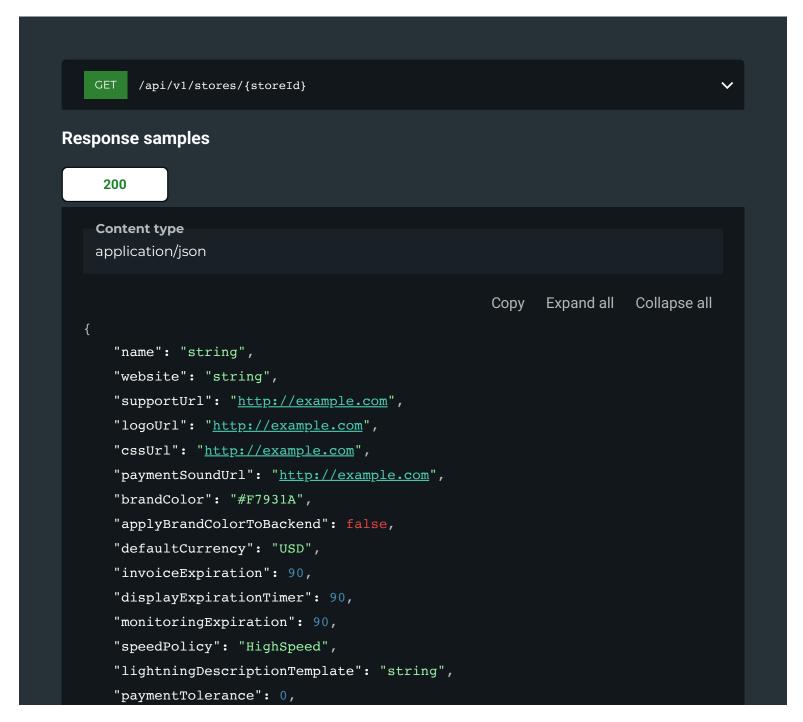
required

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

#### Responses

```
    200 specified store
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store
```



```
"archived": false,
  "anyoneCanCreateInvoice": false,
- "receipt": {
     "enabled": true,
     "showQR": null,
     "showPayments": null
 },
  "lightningAmountInSatoshi": false,
  "lightningPrivateRouteHints": false,
  "onChainWithLnInvoiceFallback": false,
  "redirectAutomatically": false,
  "showRecommendedFee": true,
  "recommendedFeeBlockTarget": 1,
  "defaultLang": "en",
  "htmlTitle": "string",
  "networkFeeMode": "Always",
  "payJoinEnabled": false,
  "autoDetectLanguage": false,
  "showPayInWalletButton": true,
  "showStoreHeader": true,
  "celebratePayment": true,
  "playSoundOnPayment": false,
  "lazyPaymentMethods": false,
  "defaultPaymentMethod": "BTC-CHAIN",
  "paymentMethodCriteria": { },
  "id": "string"
```

## Update store

Update the specified store

## API\_Key ( btcpay.store.canmodifystoresettings ) AND Basic **AUTHORIZATIONS:** PATH PARAMETERS → storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID REQUEST BODY SCHEMA: application/json - name strina The name of the store string <url> Nullable → website The absolute url of the store → supportUrl string <uri> Nullable The support URI of the store, can contain the placeholders {OrderId} and {InvoiceId}. Can be any valid URI, such as a website, email, and nostr. ⊣ logoUrl string <uri>> Nullable Absolute URL to a logo file or a reference to an uploaded file id with fileid:ID ⊣ cssUrl string <uri>> Nullable Absolute URL to CSS file to customize the public/customerfacing pages of the store. (Invoice, Payment Request, Pull Payment, etc.) or a reference to an uploaded file id with fileid:ID → paymentSoundUrl string <uri> Nullable Absolute URL to a sound file or a reference to an uploaded file id with fileid:ID → brandColor string Nullable The brand color of the store in HEX format

boolean

Default: false

applyBrandColorToBackend

Apply the brand color to the store's backend as well → defaultCurrency string Default: "USD" The default currency of the store → invoiceExpiration number < seconds > [ 60 .. 2073600 ] Default: 900 The time after which an invoice is considered expired if not paid. The value will be rounded down to a minute. displayExpirationTimer number < seconds > [ 60 .. 2073600 ] Default: 300 The time left that will trigger the countdown timer on the checkout page to be shown. The value will be rounded down to a minute. monitoringExpiration number < seconds > [ 600 .. 2073600 ] Default: 3600 The time after which an invoice which has been paid but not confirmed will be considered invalid. The value will be rounded down to a minute. → speedPolicy string (SpeedPolicy) Enum: "HighSpeed" "LowMediumSpeed" "LowSpeed" "MediumSpeed" This is a risk mitigation parameter for the merchant to configure how they want to fulfill orders depending on the number of block confirmations for the transaction made by the consumer on the selected cryptocurrency. "HighSpeed": 0 confirmations (1 confirmation if RBF enabled in transaction) "MediumSpeed": 1 confirmation "LowMediumSpeed": 2 confirmations "LowSpeed": 6 confirmations → lightningDescriptionTemplate string Nullable The BOLT11 description of the lightning invoice in the checkout. You can use placeholders '{StoreName}', '{ItemDescription}' and '{OrderId}'.

H	paymentTolerance	number <double> [ 0 100 ] Default: 0</double>
		Consider an invoice fully paid, even if the payment is missing 'x' % of the full amount.
H	archived	boolean
		Default: false
		If true, the store does not appear in the stores list by default.
H	anyoneCanCreateInvoice	boolean
		Default: false
		If true, then no authentication is needed to create invoices on
		this store.
H	receipt >	object Nullable
		Additional settings to customize the public receipt
H	lightningAmountInSatoshi	boolean
		Default: false
		If true, lightning payment methods show amount in satoshi in
		the checkout page.
H	lightningPrivateRouteHints	boolean
		Default: false
		Should private route hints be included in the lightning payment
		of the checkout page.
H	$\verb"onChainWithLnInvoiceFallback"$	boolean
		Default: false
		Unify on-chain and lightning payment URL.
H	redirectAutomatically	boolean
		Default: false
	After successfull payment, should the checkout page redirect	
		the user automatically to the redirect URL of the invoice?
H	showRecommendedFee	boolean
		Default: true
H	recommendedFeeBlockTarget	integer <int32></int32>

Default: 1 The fee rate recommendation in the checkout page for the onchain payment to be confirmed after 'x' blocks. defaultLang string Default: "en" The default language to use in the checkout page. (The different translations available are listed here → htmlTitle string Nullable The HTML title of the checkout page (when you over the tab in your browser) networkFeeMode string (NetworkFeeMode) Enum: "Always" "MultiplePaymentsOnly" "Never" Check whether network fee should be added to the invoice if on-chain payment is used. (More information) → payJoinEnabled boolean Default: false If true, payjoin will be proposed in the checkout page if possible. (More information) → autoDetectLanguage boolean Default: false If true, the language on the checkout page will adapt to the language defined by the user's browser settings → showPayInWalletButton boolean Default: true If true, the "Pay in wallet" button will be shown on the checkout page (Checkout V2) → showStoreHeader boolean Default: true If true, the store header will be shown on the checkout page (Checkout V2) - celebratePayment boolean Default: true

If true, payments on the checkout page will be celebrated with confetti (Checkout V2) → playSoundOnPayment boolean Default: false If true, sounds on the checkout page will be enabled (Checkout V2) → lazyPaymentMethods boolean Default: false If true, payment methods are enabled individually upon user interaction in the invoice → defaultPaymentMethod string (PaymentMethodId) Payment method IDs. Available payment method IDs for Bitcoin are: "BTC-CHAIN": Onchain "BTC-LN": Lightning "BTC-LNURL": LNURL → paymentMethodCriteria object Nullable The criteria required to activate specific payment methods. ⊣ id string The id of the store

#### Responses

200 updated specified store
 400 A list of errors that occurred when updating the store
 403 If you are authenticated but forbidden to update the specified store
 404 The key is not found for this store

PUT /api/v1/stores/{storeId} Request samples **Payload Content type** application/json Expand all Collapse all Copy "name": "string", "website": "string", "supportUrl": "<a href="http://example.com"">http://example.com</a>", "logoUrl": "<a href="http://example.com"">http://example.com</a>", "cssUrl": "http://example.com", "paymentSoundUrl": "<a href="http://example.com"">http://example.com</a>", "brandColor": "#F7931A", "applyBrandColorToBackend": false, "defaultCurrency": "USD", "invoiceExpiration": 90, "displayExpirationTimer": 90, "monitoringExpiration": 90, "speedPolicy": "HighSpeed", "lightningDescriptionTemplate": "string", "paymentTolerance": 0, "archived": false, "anyoneCanCreateInvoice": false, - "receipt": { "enabled": true, "showQR": null,

**}**,

"showPayments": null

"lightningAmountInSatoshi": false,

```
"lightningPrivateRouteHints": false,
"onChainWithLnInvoiceFallback": false,
"redirectAutomatically": false,
"showRecommendedFee": true,
"recommendedFeeBlockTarget": 1,
"defaultLang": "en",
"htmlTitle": "string",
"networkFeeMode": "Always",
"payJoinEnabled": false,
"autoDetectLanguage": false,
"showPayInWalletButton": true,
"showStoreHeader": true,
"celebratePayment": true,
"playSoundOnPayment": false,
"lazyPaymentMethods": false,
"defaultPaymentMethod": "BTC-CHAIN",
"paymentMethodCriteria": { },
"id": "string"
```

#### **Response samples**

200

400

```
Content type
application/json

Copy Expand all Collapse all

{
    "name": "string",
    "website": "string",
    "supportUrl": "http://example.com",
    "logoUrl": "http://example.com",
    "cssUrl": "http://example.com",
    "paymentSoundUrl": "http://example.com",
    "brandColor": "#F7931A",
```

```
"applyBrandColorToBackend": false,
  "defaultCurrency": "USD",
  "invoiceExpiration": 90,
  "displayExpirationTimer": 90,
  "monitoringExpiration": 90,
  "speedPolicy": "HighSpeed",
  "lightningDescriptionTemplate": "string",
  "paymentTolerance": 0,
  "archived": false,
  "anyoneCanCreateInvoice": false,
- "receipt": {
      "enabled": true,
      "showQR": null,
      "showPayments": null
  "lightningAmountInSatoshi": false,
  "lightningPrivateRouteHints": false,
  "onChainWithLnInvoiceFallback": false,
  "redirectAutomatically": false,
  "showRecommendedFee": true,
  "recommendedFeeBlockTarget": 1,
  "defaultLang": "en",
  "htmlTitle": "string",
  "networkFeeMode": "Always",
  "payJoinEnabled": false,
  "autoDetectLanguage": false,
  "showPayInWalletButton": true,
  "showStoreHeader": true,
  "celebratePayment": true,
  "playSoundOnPayment": false,
  "lazyPaymentMethods": false,
  "defaultPaymentMethod": "BTC-CHAIN",
  "paymentMethodCriteria": { },
  "id": "string"
```

### Remove Store

Removes the specified store. If there is another user with access, only your access will be removed.

#### Responses

200 The store has been removed
400 A list of errors that occurred when removing the store
403 If you are authenticated but forbidden to remove the specified store
404 The key is not found for this store



```
Copy Expand all Collapse all

[
- {
    "path": "string",
    "message": "string"
    }
]
```

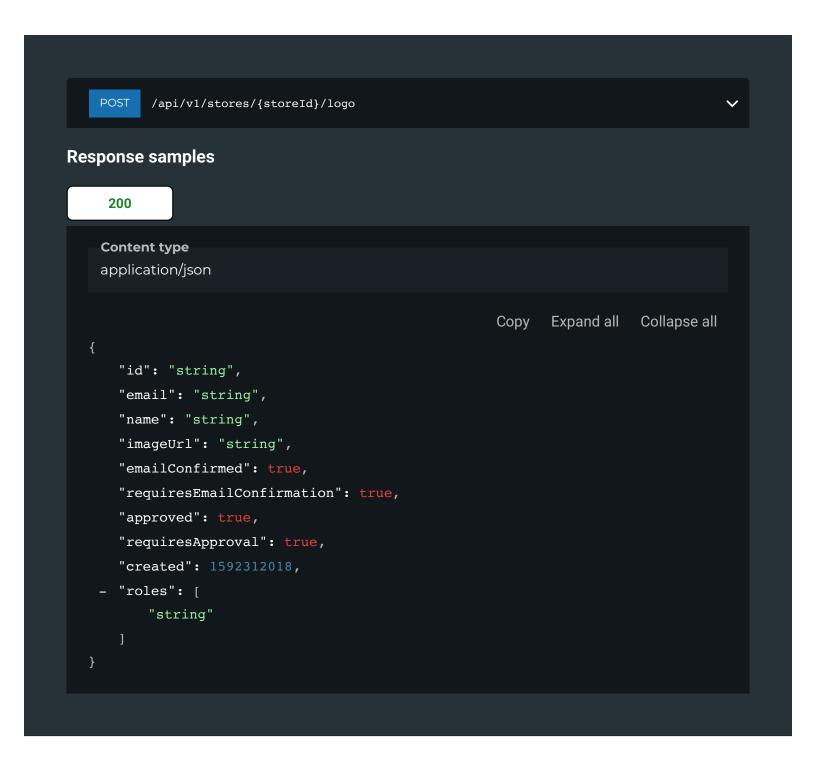
# Uploads a logo for the store

Uploads a logo for the store

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS	
→ storeId  required	string (StoreId)
	Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
	The store ID
REQUEST BODY SCHEMA:	multipart/form-data
⊣ file	string string >
	The logo

```
> 200 Uploads a logo for the store

— 404 The store could not be found
```



# Deletes the store logo

Delete the store's logo

#### Responses

- 200 Store logo deleted successfully

- 404 The store could not be found



## Get store's roles

View information about the specified store's roles

```
    200 The user roles available for this store
    403 If you are authenticated but forbidden to get the store's roles
    404 Store not found
```

```
Response samples

Content type
application/json

Copy Expand all Collapse all

{
    "id": "Owner",
    "role": "Owner",
    "permissions": [
        "btcpay.store.canmodifystoresettings"
    ],
    "isServerRole": true
```

]

Stores (Payout Processors)

Stores (Payout Processors) operations

# Get store configured payout processors

Get store configured payout processors

AUTHORIZATIONS: API\_Key ( btcpay.store.canviewstoresettings ) AND Basic

PATH PARAMETERS

→ storeId string (StoreId)

required

To the string (StoreId)

The

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

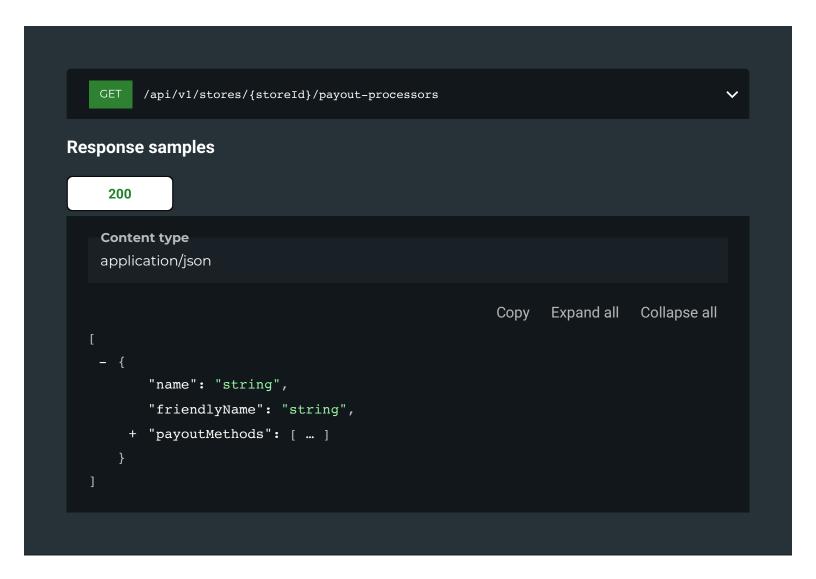
The store ID

## Responses

> 200 configured payout processors

- 403 If you are authenticated but forbidden to view the specified store

- 404 The key is not found for this store



# Remove store configured payout processor

Remove store configured payout processor



#### Responses

```
200 removed
403 If you are authenticated but forbidden to view the specified store
404 The key is not found for this store
```

```
DELETE /api/v1/stores/{storeId}/payout-processors/{processor}/{paymentMethodId}
```

# Get configured store onchain automated payout processors

Get configured store onchain automated payout processors

```
API_Key ( btcpay.store.canviewstoresettings ) AND Basic

PATH PARAMETERS

I paymentMethodId string (PaymentMethodId) Example: BTC-CHAIN
```

→ storeId

required

The payment method id of the payment method to update

string (StoreId)

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

```
    200 configured processors
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store
```

```
Response samples

Content type
application/json

Copy Expand all Collapse all

[

    "payoutMethodId": "BTC-LN",
    "feeTargetBlock": 0,
    "intervalSeconds": 90,
    "threshold": "0.1",
    "processNewPayoutsInstantly": false
}
```

J

# Update configured store onchain automated payout processors

Update configured store onchain automated payout processors

```
AUTHORIZATIONS:
                        API_Key (| btcpay.store.canviewstoresettings | ) AND Basic
PATH PARAMETERS
→ paymentMethodId
                        string (PaymentMethodId)
  required
                        Example: BTC-CHAIN
                        The payment method id of the payment method to update
⊣ storeId
                        string (StoreId)
   required
                        Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                        The store ID
REQUEST BODY SCHEMA:
                        application/json

→ feeTargetBlock

                                   number Nullable
                                   How many blocks should the fee rate calculation target to
                                   confirm in. Set to 1 if not provided

→ intervalSeconds

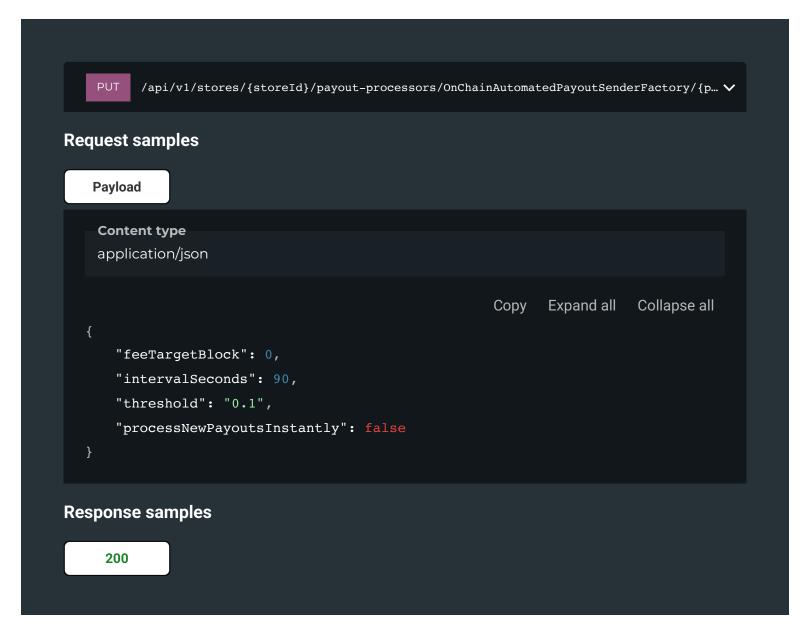
                                   number < seconds >
                                   How often should the processor run

→ threshold

                                   string <decimal> >= 0
                                   Only process payouts when this payout sum is reached.
→ processNewPayoutsInstantly boolean
                                   Default: false
                                   Skip the interval when ane eligible payout has been approved (or
```

#### created with pre-approval)

- > 200 configured processor
- 403 If you are authenticated but forbidden to view the specified store
- 404 The key is not found for this store



```
Content type
application/json

Copy Expand all Collapse all

{
    "payoutMethodId": "BTC-LN",
    "feeTargetBlock": 0,
    "intervalSeconds": 90,
    "threshold": "0.1",
    "processNewPayoutsInstantly": false
}
```

# Get configured store Lightning automated payout processors

Get configured store Lightning automated payout processors

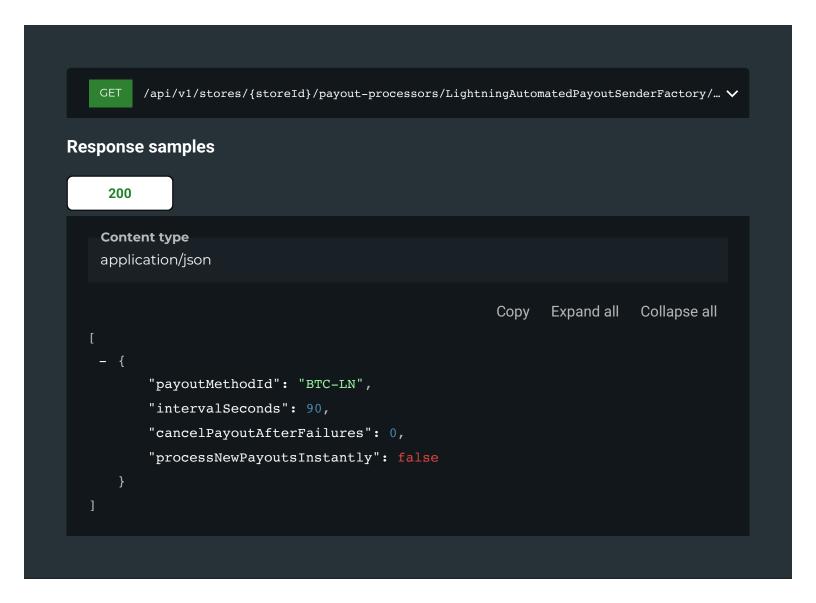
```
API_Key ( btcpay.store.canviewstoresettings ) AND Basic

PATH PARAMETERS

| payoutMethodId required | string (PayoutMethodId) | Example: BTC-CHAIN | The payout method id

| storeId required | string (StoreId) | Example: 9CinzKoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 | The store ID
```

```
    200 configured processors
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store
```



# Update configured store Lightning automated payout processors

Update configured store Lightning automated payout processors

AUTHORIZATIONS: API\_Key (|btcpay.store.canviewstoresettings|) AND Basic

#### PATH PARAMETERS

required string (PayoutMethodId)

Example: BTC-CHAIN

The payout method id

→ storeId string (StoreId)

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

#### REQUEST BODY SCHEMA: application/json

intervalSeconds number < seconds >

How often should the processor run

cancelPayoutAfterFailures number Nullable

How many failures should the processor tolerate before

cancelling the payout

→ processNewPayoutsInstantly boolean

Default: false

Skip the interval when ane eligible payout has been approved (or

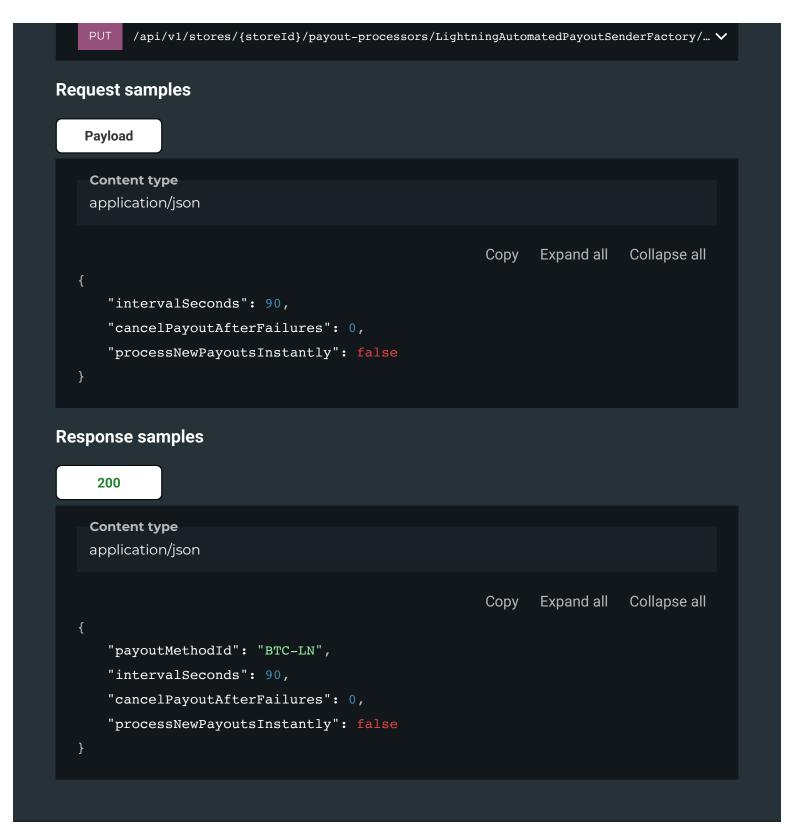
created with pre-approval)

## Responses

> 200 configured processor

- 403 If you are authenticated but forbidden to view the specified store

- 404 The key is not found for this store



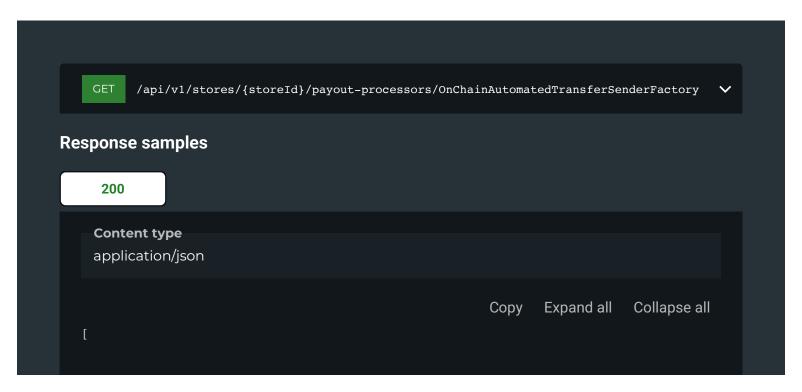
Get configured store onchain automated payout

## processors

Get configured store onchain automated payout processors

### Responses

200 configured processors
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store



```
- {
    "payoutMethodId": "BTC-LN",
    "feeTargetBlock": 0,
    "intervalSeconds": 90,
    "threshold": "0.1",
    "processNewPayoutsInstantly": false
}
```

# Update configured store onchain automated payout processors

Update configured store onchain automated payout processors

```
API_Key ( btcpay.store.canviewstoresettings ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS

→ storeId

                       string (StoreId)
  required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
REQUEST BODY SCHEMA: application/json
⊣ feeTargetBlock
                                   number Nullable
                                   How many blocks should the fee rate calculation target to
                                   confirm in. Set to 1 if not provided

→ intervalSeconds

                                   number < seconds >
                                   How often should the processor run

→ threshold

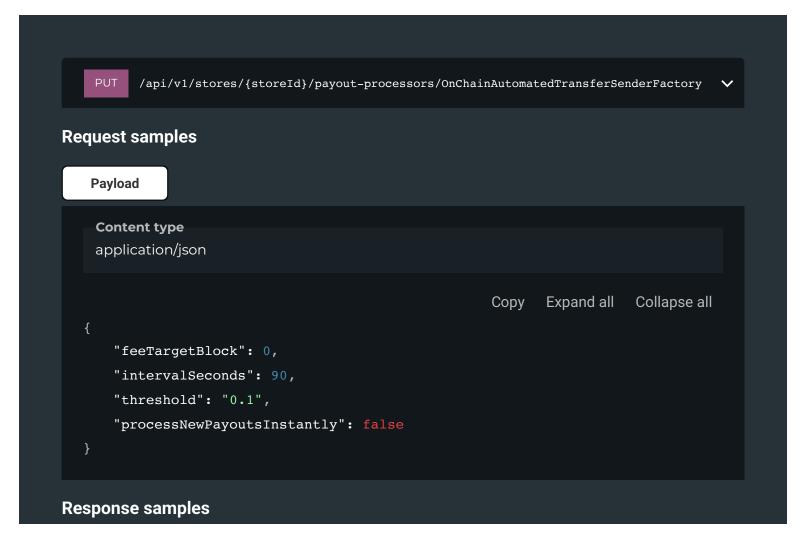
                                   string <decimal> >= 0
                                   Only process payouts when this payout sum is reached.
```

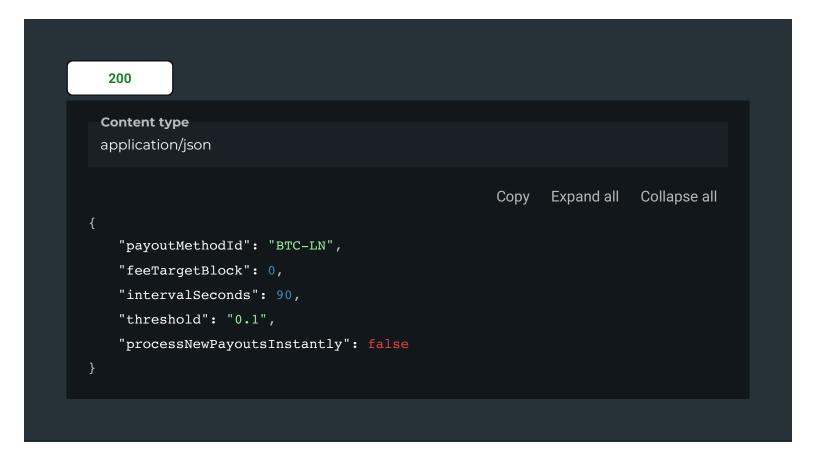
□ processNewPayoutsInstantly boolean

Default: false

Skip the interval when ane eligible payout has been approved (or created with pre-approval)

```
    200 configured processor
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store
```





# Get configured store Lightning automated payout processors

Get configured store Lightning automated payout processors

```
API_Key (| btcpay.store.canviewstoresettings |) AND Basic

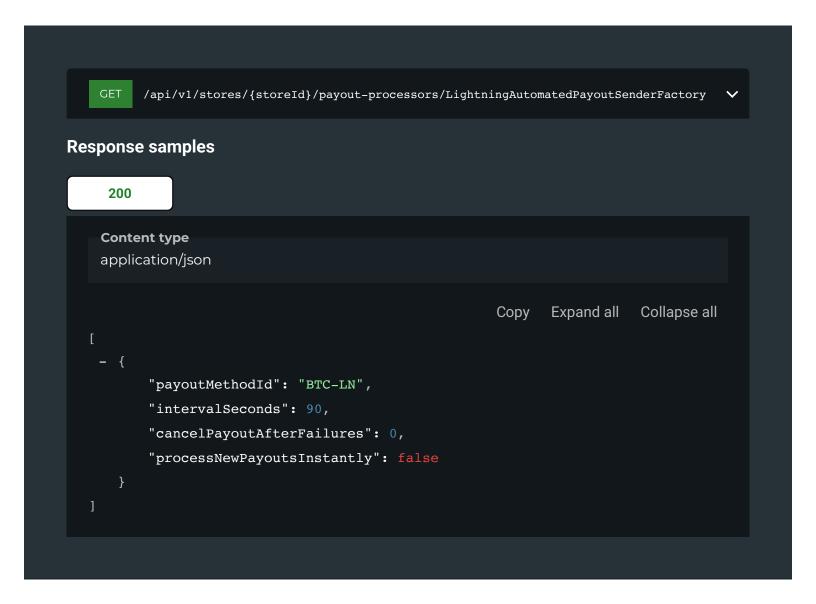
PATH PARAMETERS

— storeId string (StoreId)

required Example: 9Cinzkoanxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID
```

```
    200 configured processors
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store
```



## Users

User operations

## Get current user information

View information about the current user

```
AUTHORIZATIONS: API_Key ( btcpay.user.canviewprofile ) AND Basic
```

```
> 200 Information about the current user

— 404 The user could not be found
```

```
Response samples

200

Content type
application/json

Copy Expand all Collapse all

{
    "id": "string",
    "email": "string",
    "name": "string",
    "imageUrl": "string",
    "emailConfirmed": true,
```

```
"requiresEmailConfirmation": true,
    "approved": true,
    "requiresApproval": true,
    "created": 1592312018,
    - "roles": [
        "string"
    ]
}
```

# Update current user information

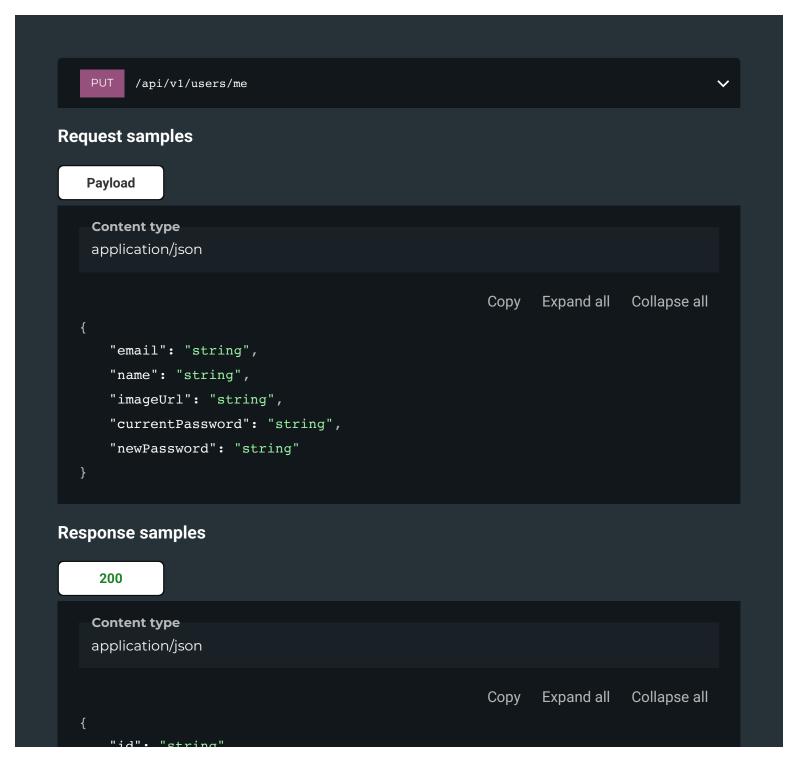
Update the current user

```
AUTHORIZATIONS:
                        API_Key (|btcpay.user.canmodifyprofile|) AND Basic
REQUEST BODY SCHEMA:
                         application/json
⊣ email
                        string Nullable
                        The email of the user
name
                        string Nullable
                        The name of the user
 ⊣ imageUrl
                        string Nullable
                        The profile picture URL of the user
→ currentPassword
                        string Nullable
                        The current password of the user
⊣ newPassword
                        string Nullable
                        The new password of the user
```

#### Responses

> 200 Information about the current user

— 404 The user could not be found



```
"email": "string",
    "name": "string",
    "imageUrl": "string",
    "requiresEmailConfirmation": true,
    "approved": true,
    "requiresApproval": true,
    "created": 1592312018,
    - "roles": [
        "string"
    ]
}
```

# Deletes user profile

Deletes user profile and associated user data for user making the request

AUTHORIZATIONS: API\_Key ( btcpay.user.candeleteuser ) AND Basic

### Responses

- 200 User and associated data deleted successfully
- 404 The user could not be found



# Uploads a profile picture for the current user

Uploads a profile picture for the current user

AUTHORIZATIONS: API\_Key ( btcpay.user.canmodifyprofile ) AND Basic

REQUEST BODY SCHEMA: multipart/form-data

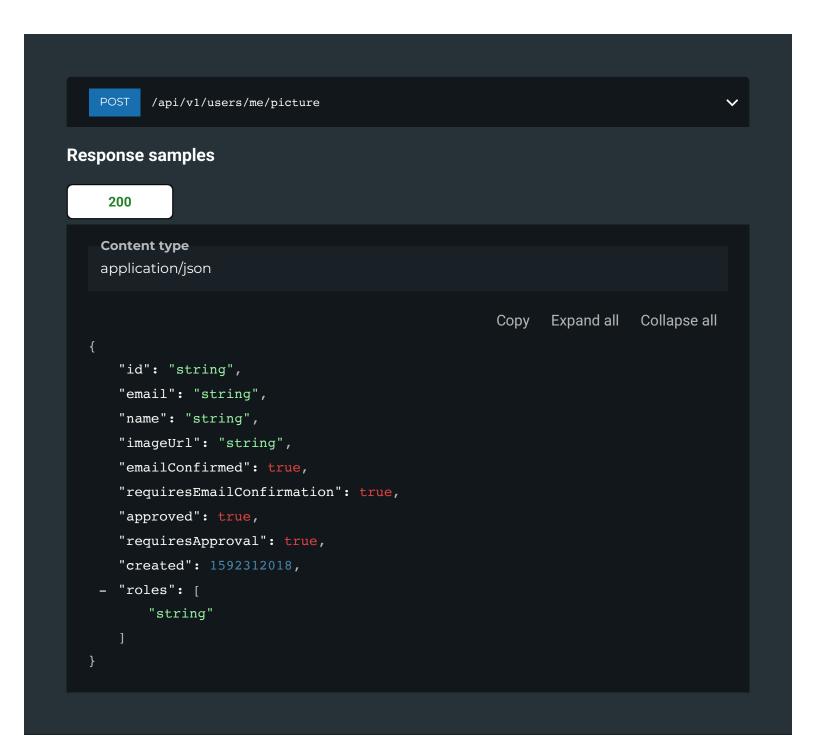
→ file string < binary >

The profile picture

#### Responses

> 200 Uploads a profile picture for the current user

- 404 The user could not be found

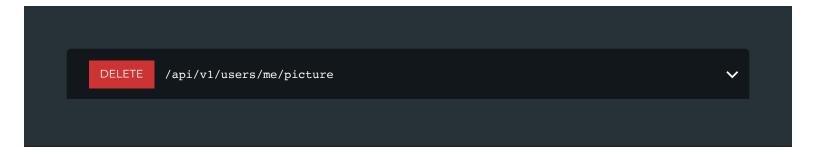


# Deletes user profile picture

Deletes the user profile picture

AUTHORIZATIONS: API\_Key ( btcpay.user.canmodifyprofile ) AND Basic

- 200 Profile picture deleted successfully
- 404 The user could not be found



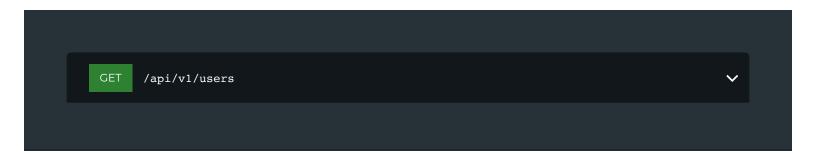
### Get all users

Load all users that exist.

AUTHORIZATIONS: API\_Key ( btcpay.server.canviewusers ) AND Basic

### Responses

200 Users found
401 Missing authorization for loading the users
403 Authorized but forbidden to load the users. You have the wrong API permissions.



## Create user

Create a new user.

This operation can be called without authentication in any of this cases:

- · There is not any administrator yet on the server,
- User registrations are not disabled in the server's policies.

If the first administrator is created by this call, user registrations are automatically disabled.

**AUTHORIZATIONS:** API\_Key (| btcpay.server.cancreateuser |) AND Basic REQUEST BODY SCHEMA: application/json email string The email of the new user → name string Nullable The name of the new user ⊣ imageUrl string Nullable The profile picture URL of the new user → password string Nullable The password of the new user (if no password is set, an email will be sent to the user requiring him to set the password) → isAdministrator boolean Nullable Default: false Make this user administrator (only if you have the unrestricted permission of a server administrator)

#### Responses

> 201 Information about the new user

> 400 A list of errors that occurred when creating the user

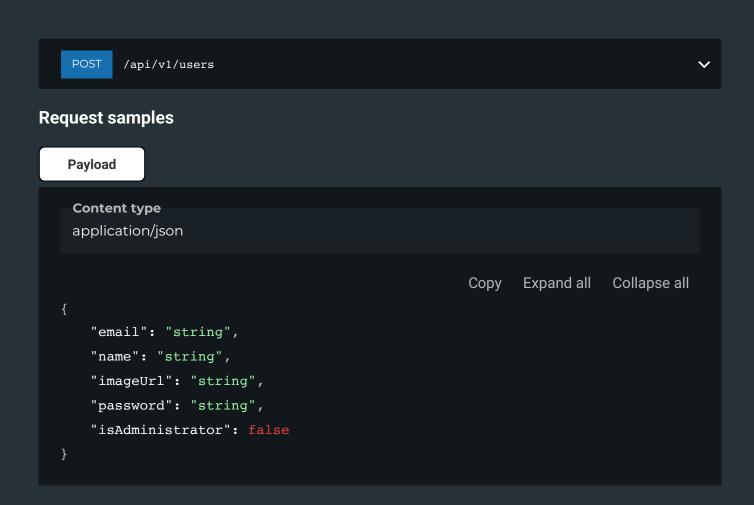
**—** 401

If you need to authenticate for this endpoint (ie. the server settings policies lock subscriptions and that an admin already exists)

**-403** 

If you are authenticated but forbidden to create a new user (ie. you don't have the unrestricted permission on a server administrator or if you are not administrator and user registrations are disabled in the server's policies)

- 429 DDoS protection if you are creating more than 2 accounts every minutes (non-admin only)



#### **Response samples**

```
Content type
application/json

Copy Expand all Collapse all

{
    "id": "string",
    "email": "string",
    "name": "string",
    "imageUrl": "string",
    "emailConfirmed": true,
    "requiresEmailConfirmation": true,
    "approved": true,
```

# Get user by ID or Email

Get 1 user by ID or Email.

```
    200 User found
    401 Missing authorization for loading the user
    403 Authorized but forbidden to load the user. You have the wrong API permissions.
    404 No user found with this ID or email
```

```
/api/v1/users/{idOrEmail}
Response samples
     200
    Content type
    application/json
                                                             Expand all Collapse all
                                                      Copy
      "id": "string",
      "email": "string",
      "name": "string",
      "imageUrl": "string",
      "emailConfirmed": true,
      "requiresEmailConfirmation": true,
      "approved": true,
      "requiresApproval": true,
      "created": 1592312018,
    - "roles": [
          "string"
```

### Delete user

Delete a user.

Must be an admin to perform this operation.

Attempting to delete the only admin user will not succeed.

All data associated with the user will be deleted as well if the operation succeeds.



## Responses

200 User has been successfully deleted
401 Missing authorization for deleting the user
403 Authorized but forbidden to delete the user. Can happen if you attempt to delete the only admin user.
404 User with provided ID was not found



# Toggle user lock out

Lock or unlock a user.

Must be an admin to perform this operation.

Attempting to lock the only admin user will not succeed.

API\_Key ( btcpay.user.canmodifyserversettings ) AND Basic

PATH PARAMETERS

I idOrEmail string The user's id or email

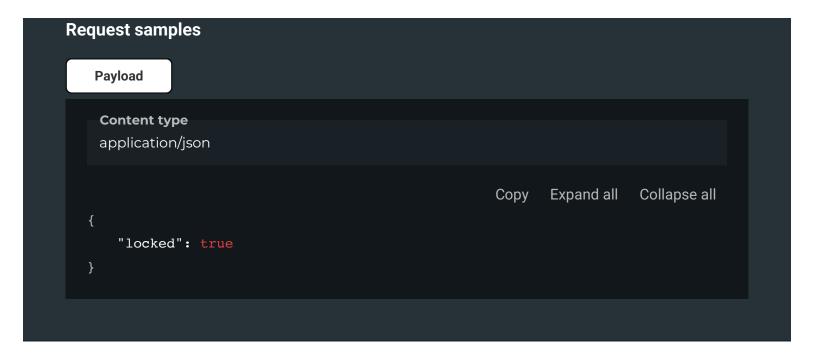
REQUEST BODY SCHEMA: application/json

I locked boolean Whether to lock or unlock the user

## Responses

- 200 User has been successfully toggled
- 401 Missing authorization for locking the user
- 403 Authorized but forbidden to lock the user. Can happen if you attempt to disable the only admin user.
- 404 User with provided ID was not found





# Toggle user approval

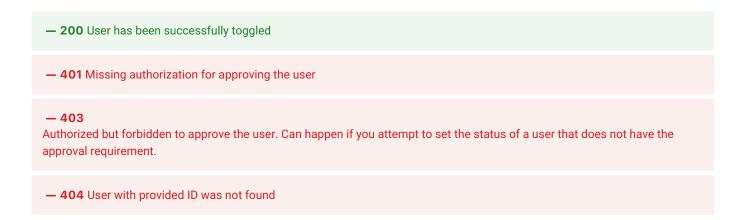
Approve or unapprove a user.

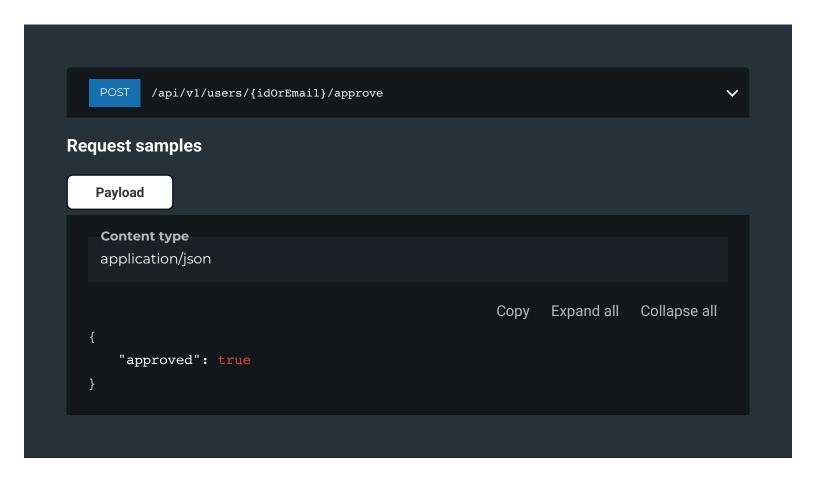
Must be an admin to perform this operation.

Attempting to (un)approve a user for which this requirement does not exist will not succeed.

AUTHORIZATIONS:	API_Key( btcpay.user.canmodifyserversettings ) AND Basic
PATH PARAMETERS	
→ idOrEmail required	string The user's id or email
REQUEST BODY SCHEMA:	application/json
→ approved	boolean Whether to approve or unapprove the user

### Responses





# Webhooks

Webhook operations

## Get webhooks of a store

View webhooks of a store

API\_Key ( btcpay.store.webhooks.canmodifywebhooks ) AND Basic

PATH PARAMETERS

string (StoreId)

required

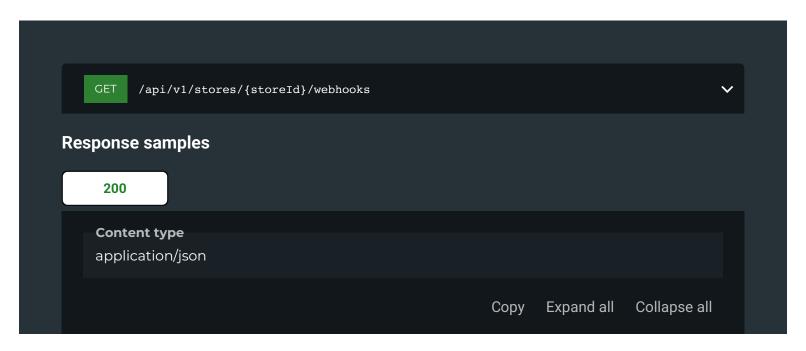
Example: 9CinzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

### Responses

> 200 List of webhooks

— 404 The key is not found for this list of webhooks



### Create a new webhook

Create a new webhook

```
AUTHORIZATIONS:
                        API_Key ( btcpay.store.webhooks.canmodifywebhooks ) AND Basic
PATH PARAMETERS

→ storeId

                        string (StoreId)
   required
                        Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                        The store ID
REQUEST BODY SCHEMA: application/json
 - enabled
                           boolean
                           Default: true
                           Whether this webhook is enabled or not
→ automaticRedelivery boolean
                           Default: true
                           If true, BTCPay Server will retry to redeliver any failed delivery after 10
                           seconds, 1 minutes and up to 6 times after 10 minutes.
 ⊣ url
                           string
```

The endpoint where BTCPay Server will make the POST request with the webhook body

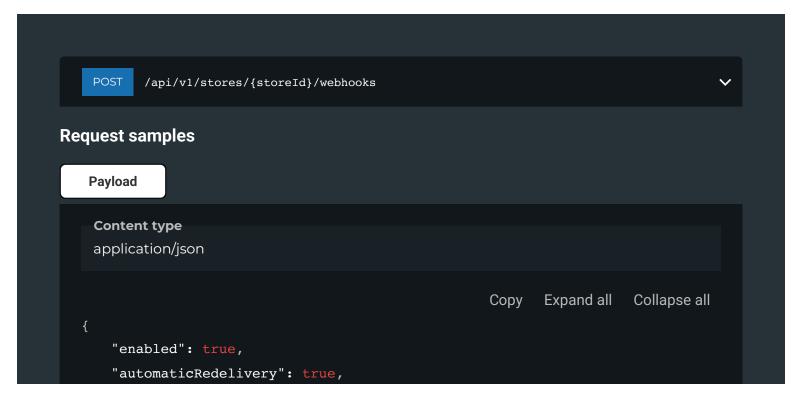
object
Which event should be received by this endpoint

secret

string Nullable
Must be used by the callback receiver to ensure the delivery comes from BTCPay Server. BTCPay Server includes the BTCPay-Sig HTTP header, whose format is sha256=HMAC256(UTF8(webhook's secret), body). The pattern to authenticate the webhook is similar to how to secure webhooks in Github. If left out, null, or empty, the secret will be auto-generated.

#### Responses

- > 200 Information about the new webhook
- > 400 A list of errors that occurred when creating the webhook



```
"url": "string",
    - "authorizedEvents": {
          "everything": true,
       + "specificEvents": [ ... ]
      },
      "secret": "string"
Response samples
  200
    Content type
    application/json
                                                             Expand all Collapse all
                                                      Copy
      "enabled": true,
      "automaticRedelivery": true,
      "url": "string",
    - "authorizedEvents": {
          "everything": true,
       + "specificEvents": [ ... ]
      },
      "id": "string",
      "secret": "string"
```

## Get a webhook of a store

View webhook of a store

### Responses

```
> 200 A webhook

— 404 The webhook has not been found
```

```
Response samples

Content type
application/json

Copy Expand all Collapse all

{
    "enabled": true,
    "automaticRedelivery": true,
    "url": "string",
```

```
- "authorizedEvents": {
        "everything": true,
        + "specificEvents": [ ... ]
    },
    "id": "string"
}
```

# Update a webhook

Update a webhook

```
API_Key ( btcpay.store.webhooks.canmodifywebhooks ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
⊣ storeId
                       string (StoreId)
  required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
 webhookId
                       string
  required
                       The webhook id
REQUEST BODY SCHEMA:
                        application/json
- enabled
                          boolean
                          Default: true
                          Whether this webhook is enabled or not
automaticRedelivery boolean
                          Default: true
                          If true, BTCPay Server will retry to redeliver any failed delivery after 10
                          seconds, 1 minutes and up to 6 times after 10 minutes.
 ⊣ url
                          string
```

The endpoint where BTCPay Server will make the POST request with the webhook body

object
Which event should be received by this endpoint

secret

string Nullable
Must be used by the callback receiver to ensure the delivery comes from BTCPay Server. BTCPay Server includes the BTCPay—Sig HTTP header, whose format is sha256=HMAC256(UTF8(webhook's secret), body). The pattern to authenticate the webhook is similar to how to secure webhooks in Github. If left out, null, or empty, the secret will not be changed.

#### Responses

- > 200 Information about the updated webhook
- > 400 A list of errors that occurred when creating the webhook



```
"url": "string",
- "authorizedEvents": {
         "everything": true,
         + "specificEvents": [ ... ]
},
         "secret": "string"
}
```

### **Response samples**

200

400

```
Content type
application/json

Copy Expand all Collapse all

{
    "enabled": true,
    "automaticRedelivery": true,
    "url": "string",

- "authorizedEvents": {
        "everything": true,
        + "specificEvents": [ ... ]
    },
    "id": "string"
}
```

## Delete a webhook

#### Delete a webhook

AUTHORIZATIONS:	API_Key( btcpay.store.webhooks.canmodifywebhooks ) AND Basic
PATH PARAMETERS	
→ storeId required	string (StoreId) Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID
→ webhookId required	string The webhook id

## Responses

```
200 The webhook has been deleted404 The webhook does not exist
```



## Get latest deliveries

List the latest deliveries to the webhook, ordered from the most recent

API\_Key (|btcpay.store.webhooks.canmodifywebhooks|) AND Basic **AUTHORIZATIONS:** PATH PARAMETERS ⊣ storeId string (StoreId) required Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID → webhookId string required The webhook id **QUERY PARAMETERS** - count string The number of latest deliveries to fetch

#### Responses

> 200 List of deliveries

— 404 The key is not found for this list of deliveries



```
Copy Expand all Collapse all

[
- {
    "id": "string",
    "timestamp": 1592312018,
    "httpCode": 0,
    "errorMessage": "string",
    "status": "string"
    }
]
```

# Get a webhook delivery

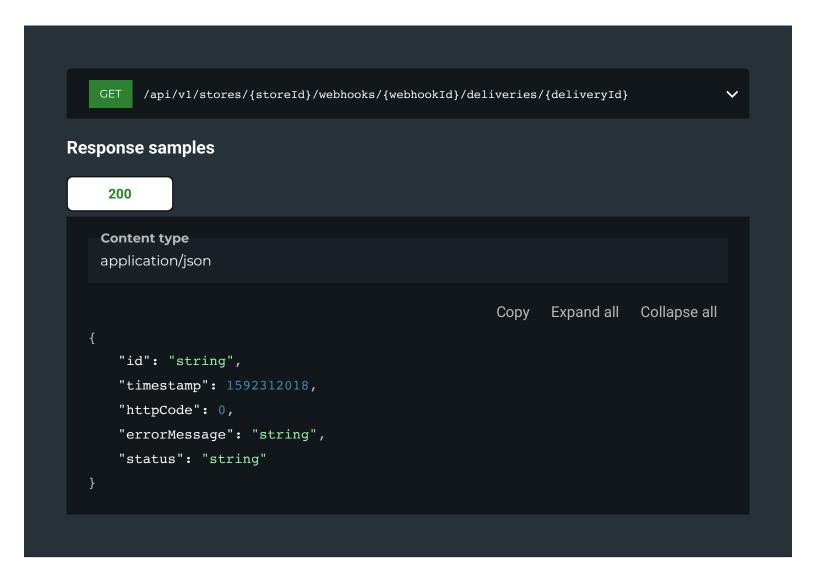
Information about a webhook delivery

```
AUTHORIZATIONS:
                       API_Key (|btcpay.store.webhooks.canmodifywebhooks|) AND Basic
PATH PARAMETERS
⊣ deliveryId
                       string
  required
                       The id of the delivery
⊣ storeId
                       string (StoreId)
   required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
 ⊣ webhookId
                       string
  required
                       The webhook id
```

## Responses

> 200 Information about a delivery

— 404 The delivery does not exists.



# Get the delivery's request

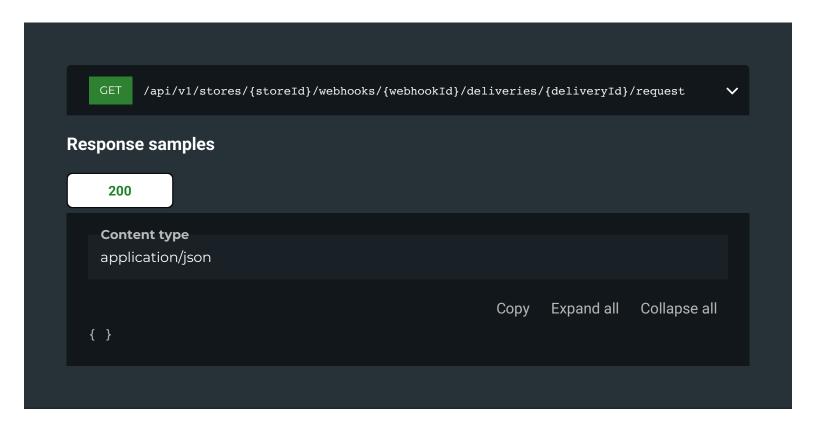
The delivery's JSON request sent to the endpoint

```
AUTHORIZATIONS: API_Key( btcpay.store.webhooks.canmodifywebhooks ) AND Basic
PATH PARAMETERS
```



### Responses

```
    200 The delivery's JSON Request
    404 The delivery does not exists.
    409 webhookdelivery-pruned: This webhook delivery has been pruned, so it can't be redelivered.
```



## Redeliver the delivery

#### Redeliver the delivery

AUTHORIZATIONS: API\_Key ( btcpay.store.webhooks.canmodifywebhooks ) AND Basic

#### **PATH PARAMETERS**

deliveryId
required
The id of the delivery

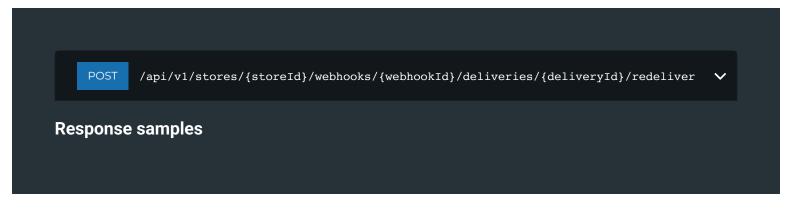
storeId
string (StoreId)
Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID

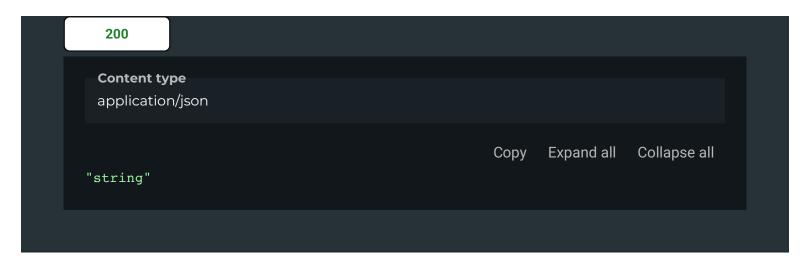
webhookId
required
The webhook id

The webhook id

## Responses

200 The new delivery id being broadcasted. (Broadcast happen asynchronously with this call)
 404 The delivery does not exists.
 409 webhookdelivery-pruned: This webhook delivery has been pruned, so it can't be redelivered.





## InvoiceCreated Webhook

A new invoice has been created

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string required Examp

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

→ deliveryId string

The delivery id of the webhook

→ webhookId string

The id of the webhook

→ originalDeliveryId string

If this delivery is a redelivery, the is the delivery id of the original delivery.

⊣ isRedelivery boolean

True if this delivery is a redelivery ⊣ type string The type of this event, current available are InvoiceCreated, InvoiceReceivedPayment , InvoiceProcessing , InvoiceExpired , InvoiceSettled, InvoiceInvalid, and InvoicePaymentSettled. → timestamp number <int32> The timestamp when this delivery has been created ⊣ storeId string The store id of the invoice's event → invoiceId string The invoice id of the invoice's event ⊣ metadata object User-supplied metadata added to the invoice at the time of its creation

### Responses

> default Unexpected error



```
"webhookId": "string",
      "originalDeliveryId": "string",
      "isRedelivery": true,
      "type": "string",
      "timestamp": 1592312018,
      "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
      "invoiceId": "string",
      "metadata": { }
Response samples
    default
    Content type
    application/json
                                                                        Collapse all
                                                             Expand all
                                                      Copy
      "code": "string",
      "message": "string"
```

## InvoiceExpired webhook

An invoice expired

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string required

#### Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

RE	REQUEST BODY SCHEMA: application/json		
Ļ	deliveryId	string The delivery id of the webhook	
H	webhookId	string The id of the webhook	
H	originalDeliveryId	string If this delivery is a redelivery, the is the delivery id of the original delivery.	
H	isRedelivery	boolean True if this delivery is a redelivery	
-	type	string The type of this event, current available are <a href="InvoiceCreated">InvoiceCreated</a> , <a href="InvoiceExpired">InvoiceExpired</a> ,	

#### Responses

> default Unexpected error

### **Request samples**

#### **Payload**

```
Content type
application/json

Copy Expand all Collapse all

{
    "deliveryId": "string",
    "webhookId": "string",
    "originalDeliveryId": "string",
    "isRedelivery": true,
    "type": "string",
    "timestamp": 1592312018,
    "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "invoiceId": "string",
    "metadata": { },
    "partiallyPaid": true
}
```

## Response samples

#### default

Content type
application/json

Copy Expand all Collapse all

```
{
    "code": "string",
    "message": "string"
}
```

## InvoiceReceivedPayment webhook

An invoice received a payment

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string required

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

The delivery id of the webhook

He webhookId string
The id of the webhook

HoriginalDeliveryId string
If this delivery is a redelivery, the is the delivery id of the original delivery.

His is Redelivery boolean
True if this delivery is a redelivery

His type string

The type of this event, current available are InvoiceCreated, InvoiceReceivedPayment, InvoiceProcessing, InvoiceExpired, InvoiceSettled, InvoiceInvalid, and InvoicePaymentSettled. → timestamp number <int32> The timestamp when this delivery has been created → storeId string The store id of the invoice's event → invoiceId string The invoice id of the invoice's event → metadata object User-supplied metadata added to the invoice at the time of its creation → afterExpiration boolean Whether this payment has been sent after expiration of the invoice → paymentMethodId string What payment method was used for this payment payment > object Details about the payment

## Responses

> default Unexpected error

# Request samples

**Payload** 

```
Content type application son
                                                                     Collapse all
                                                   Copy
                                                         Expand all
  "deliveryId": "string",
  "webhookId": "string",
  "originalDeliveryId": "string",
  "isRedelivery": true,
  "type": "string",
  "timestamp": 1592312018,
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "invoiceId": "string",
  "metadata": { },
  "afterExpiration": true,
  "paymentMethodId": "BTC-CHAIN",
- "payment": {
      "id": "string",
      "receivedDate": 1592312018,
      "value": "string",
      "fee": "string",
      "status": "Invalid",
      "destination": "string"
```

## Response samples

#### default

```
Content type
application/json

Copy Expand all Collapse all

{
    "code": "string",
    "message": "string"
}
```

## InvoicePaymentSettled Webhook

An payment relating to an invoice has settled

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string

required

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

#### REQUEST BODY SCHEMA: application/json

⊣ deliveryId string

The delivery id of the webhook

→ webhookId string

The id of the webhook

→ originalDeliveryId string

If this delivery is a redelivery, the is the delivery id of the original delivery.

⊣ isRedelivery boolean

True if this delivery is a redelivery

⊣ type string

The type of this event, current available are InvoiceCreated,

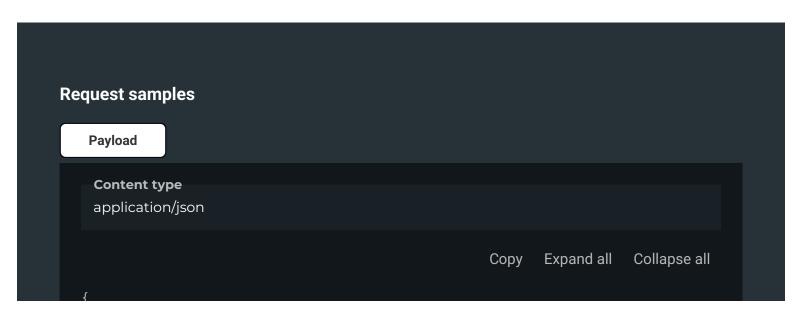
InvoiceReceivedPayment, InvoiceProcessing, InvoiceExpired, InvoiceSettled, InvoiceInvalid, and InvoicePaymentSettled.

⊣ timestamp number <int32>

The timestamp when this delivery has been created ⊣ storeId string The store id of the invoice's event → invoiceId string The invoice id of the invoice's event → metadata object User-supplied metadata added to the invoice at the time of its creation → afterExpiration boolean Whether this payment has been sent after expiration of the invoice paymentMethodId string What payment method was used for this payment payment > object Details about the payment

## Responses

> default Unexpected error



```
"deliveryId": "string",
  "webhookId": "string",
  "originalDeliveryId": "string",
  "isRedelivery": true,
  "type": "string",
  "timestamp": 1592312018,
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "invoiceId": "string",
  "metadata": { },
  "afterExpiration": true,
  "paymentMethodId": "BTC-CHAIN",
- "payment": {
      "id": "string",
      "receivedDate": 1592312018,
      "value": "string",
      "fee": "string",
      "status": "Invalid",
      "destination": "string"
```

## Response samples

#### default

```
Content type
application/json

Copy Expand all Collapse all

"code": "string",
"message": "string"
}
```

## InvoiceProcessing Webhook

Triggers when an invoice is fully paid, but doesn't have the required amount of confirmations on the blockchain yet according to your store's settings.

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string required

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

⊣ deliveryId string

The delivery id of the webhook

→ webhookId string

The id of the webhook

→ originalDeliveryId string

If this delivery is a redelivery, the is the delivery id of the original delivery.

⊣ isRedelivery boolean

True if this delivery is a redelivery

→ type string

The type of this event, current available are InvoiceCreated,

InvoiceReceivedPayment, InvoiceProcessing, InvoiceExpired, InvoiceSettled, InvoiceInvalid, and InvoicePaymentSettled.

→ timestamp number <int32>

The timestamp when this delivery has been created

⊣ storeId string

	The store id of the invoice's event
⊣ invoiceId	string
	The invoice id of the invoice's event
→ metadata	object
	User-supplied metadata added to the invoice at the time of its creation
⊣ overPaid	boolean
	Whether this invoice has received more money than expected

### Responses

> default Unexpected error

```
Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

{
  "deliveryId": "string",
  "webhookId": "string",
  "originalDeliveryId": "string",
  "isRedelivery": true,
  "type": "string",
  "timestamp": 1592312018,
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "invoiceId": "string",
```

```
"metadata": { },
    "overPaid": true
}

Response samples

default

Content type
    application/json

Copy Expand all Collapse all
{
    "code": "string",
    "message": "string"
}
```

# InvoiceInvalid webhook

An invoice became invalid

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

— BTCPay-Sig string
 required Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

H	deliveryId	string The delivery id of the webhook
H	webhookId	string The id of the webhook
H	originalDeliveryId	string If this delivery is a redelivery, the is the delivery id of the original delivery.
H	isRedelivery	boolean  True if this delivery is a redelivery
H	type	string The type of this event, current available are InvoiceCreated, InvoiceReceivedPayment, InvoiceProcessing, InvoiceExpired, InvoiceSettled, InvoiceInvalid, and InvoicePaymentSettled.
H	timestamp	number <int32> The timestamp when this delivery has been created</int32>
H	storeId	string The store id of the invoice's event
H	invoiceId	string The invoice id of the invoice's event
H	metadata	object User-supplied metadata added to the invoice at the time of its creation
4	manuallyMarked	boolean Whether the invoice have been manually marked as confirmed. If false, this invoice has received payments which could not confirm in time.

# Responses

> default Unexpected error

### **Request samples**

#### **Payload**

```
Content type
application/json

Copy Expand all Collapse all

{
    "deliveryId": "string",
    "webhookId": "string",
    "originalDeliveryId": "string",
    "isRedelivery": true,
    "type": "string",
    "timestamp": 1592312018,
    "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "invoiceId": "string",
    "metadata": { },
    "manuallyMarked": true
}
```

## Response samples

#### default

```
Content type
application/json

Copy Expand all Collapse all

"code": "string",
"message": "string"
}
```

## InvoiceSettled Webhook

Triggers when an invoice is considered settled and the merchant can proceed with the order's delivery. The invoice now has enough confirmations on the blockchain (if paid on-chain) according to your store's configuration.

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string

required

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

→ deliveryId string

The delivery id of the webhook

→ webhookId string

The id of the webhook

→ originalDeliveryId string

If this delivery is a redelivery, the is the delivery id of the original delivery.

⊣ isRedelivery boolean

True if this delivery is a redelivery

→ type string

The type of this event, current available are InvoiceCreated,

InvoiceReceivedPayment, InvoiceProcessing, InvoiceExpired, InvoiceSettled, InvoiceInvalid, and InvoicePaymentSettled.

⊣ timestamp	number <int32></int32>
	The timestamp when this delivery has been created
⊣ storeId	string
	The store id of the invoice's event
⊣ invoiceId	string
	The invoice id of the invoice's event
⊣ metadata	object
	User-supplied metadata added to the invoice at the time of its creation
⊣ manuallyMarked	boolean
	Whether the invoice have been manually marked as confirmed
⊣ overPaid	boolean
	Whether this invoice has received more money than expected
I	

## Responses

> default Unexpected error

```
Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

{
    "deliveryId": "string",
    "webhookId": "string",
```

```
"originalDeliveryId": "string",
      "isRedelivery": true,
      "type": "string",
      "timestamp": 1592312018,
      "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
      "invoiceId": "string",
      "metadata": { },
      "manuallyMarked": true,
      "overPaid": true
Response samples
    default
    Content type
    application/json
                                                                        Collapse all
                                                             Expand all
                                                      Copy
      "code": "string",
      "message": "string"
```

# PaymentRequestCreated webhook

A new payment request has been created

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string

#### required

#### Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

```
REQUEST BODY SCHEMA: application/json
```

```
The id of the payment request

string

status

string

Enum: "Completed" "Expired" "Pending"

The status of the payment request
```

## Responses

> default Unexpected error

```
Request samples

Content type
application/json

Copy Expand all Collapse all

{
    "paymentRequestId": "string",
    "status": "Completed"
}

Response samples
```

```
Content type
application/json

Copy Expand all Collapse all

"code": "string",
"message": "string"
}
```

# PaymentRequestUpdated webhook

string

A payment request has been updated

```
AUTHORIZATIONS:

API_Key AND Basic

HEADER PARAMETERS

HEADER PARAMETERS

String
required
Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9
The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

PaymentRequestId string
The id of the payment request
```

Enum: "Completed" "Expired" "Pending"

The status of the payment request

⊣ status

## Responses

> default Unexpected error

```
Request samples
   Payload
    Content type
    application/json
                                                                Expand all
                                                                            Collapse all
                                                         Copy
       "paymentRequestId": "string",
       "status": "Completed"
Response samples
    default
    Content type
    application/json
                                                                Expand all
                                                                            Collapse all
                                                         Copy
       "code": "string",
       "message": "string"
```

# PaymentRequestArchived webhook

A payment request has been archived

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string

required

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

→ paymentRequestId string

The id of the payment request

⊣ status string

Enum: "Completed" "Expired" "Pending"

The status of the payment request

## Responses

> default Unexpected error



**Payload** 

**Content type** 

application/ison

```
Expand all
                                                                             Collapse all
                                                         Copy
       "paymentRequestId": "string",
       "status": "Completed"
Response samples
    default
    Content type
    application/json
                                                                 Expand all
                                                                             Collapse all
                                                         Copy
       "code": "string",
       "message": "string"
```

# PaymentRequestStatusChanged Webhook

A payment request has had its status changed

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

BTCPay-Sig string
required
Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

# REQUEST BODY SCHEMA: application/json

```
The id of the payment request

string

string

string

Enum: "Completed" "Expired" "Pending"

The status of the payment request
```

## Responses

> default Unexpected error

# Request samples Payload Content type application/json Copy Expand all Collapse all { "paymentRequestId": "string", "status": "Completed" } Response samples default

```
Content type
application/json

Copy Expand all Collapse all

{
    "code": "string",
    "message": "string"
}
```

# PayoutCreated Webhook

A payout has been created

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string required

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

REQUEST BODY SCHEMA: application/json

```
The id of the payout

I pullPaymentId string Nullable
The id of the pull payment this payout belongs to

I payoutState string (PayoutState)
Enum: "AwaitingApproval" "AwaitingPayment" "Cancelled"
```

```
"Completed" "InProgress"

The state of the payout (AwaitingApproval, AwaitingPayment,
InProgress, Completed, Cancelled)
```

## Responses

> default Unexpected error

```
Request samples
   Payload
    Content type
    application/json
                                                                Expand all
                                                                            Collapse all
                                                         Copy
       "payoutId": "string",
       "pullPaymentId": "string",
       "payoutState": "AwaitingPayment"
Response samples
    default
    Content type
    application/json
                                                                            Collapse all
                                                         Сору
                                                                Expand all
      "code": "string"
```

```
"message": "string"
}
```

## PayoutApproved Webhook

A payout has been approved

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string required

equired Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

#### REQUEST BODY SCHEMA: application/json

— payoutId string

The id of the payout

— pullPaymentId string Nullable

The id of the pull payment this payout belongs to

→ payoutState string (PayoutState)

Enum: "AwaitingApproval" "AwaitingPayment" "Cancelled"

"Completed" "InProgress"

The state of the payout ( AwaitingApproval , AwaitingPayment ,

InProgress , Completed , Cancelled )

## Responses

> default Unexpected error

```
Request samples
   Payload
    Content type
    application/json
                                                               Expand all
                                                                           Collapse all
                                                        Copy
       "payoutId": "string",
       "pullPaymentId": "string",
       "payoutState": "AwaitingPayment"
Response samples
    default
    Content type
    application/json
                                                               Expand all
                                                                           Collapse all
                                                        Copy
       "code": "string",
      "message": "string"
```

## PayoutUpdated Webhook

#### A payout has been updated

AUTHORIZATIONS: API\_Key AND Basic

**HEADER PARAMETERS** 

→ BTCPay-Sig string required

Example:

sha256=b438519edde5c8144a4f9bcec51a9d346eca6506887c2ceeae1c0092884a97b9

The HMAC of the body's byte with the secret's of the webhook.

sha256=HMAC256(UTF8(webhook's secret), body)

#### REQUEST BODY SCHEMA: application/json

⊣ payoutId string

The id of the payout

→ pullPaymentId string Nullable

The id of the pull payment this payout belongs to

→ payoutState string (PayoutState)

Enum: "AwaitingApproval" "AwaitingPayment" "Cancelled"

"Completed" "InProgress"

The state of the payout ( AwaitingApproval , AwaitingPayment ,

InProgress , Completed , Cancelled )

## Responses

> default Unexpected error

```
Request samples
   Payload
    Content type
    application/json
                                                               Expand all
                                                                           Collapse all
                                                        Copy
       "payoutId": "string",
       "pullPaymentId": "string",
       "payoutState": "AwaitingPayment"
Response samples
    default
    Content type
    application/json
                                                               Expand all
                                                                           Collapse all
                                                        Copy
      "code": "string",
      "message": "string"
```

# Point of Sale

# Get Point of Sale app data

Returns POS app data

AUTHORIZATIONS: API\_Key AND Basic

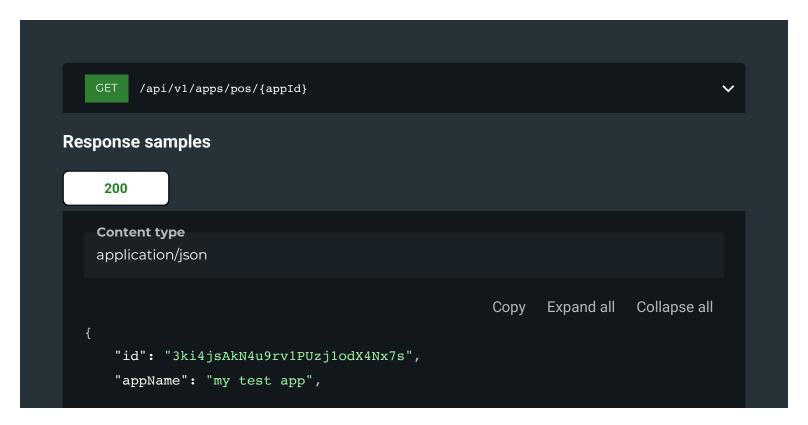
PATH PARAMETERS

→ appId string required App ID

## Responses

> 200 POS app data

— 404 POS app with specified ID was not found



```
"storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "created": 1651554744,
  "appType": "PointOfSale",
  "archived": false,
  "title": "My PoS app",
  "description": "This is my amazing PoS app",
  "defaultView": "Cart",
  "showItems": true,
  "showCustomAmount": true,
  "showDiscount": false,
  "showSearch": false,
  "showCategories": false,
  "enableTips": true,
  "currency": "BTC",
  "fixedAmountPayButtonText": "Buy for \{0\}",
  "customAmountPayButtonText": "Pay",
  "tipText": "Do you want to leave a tip?",
- "customTipPercentages": [
     15,
     18,
  ],
  "notificationUrl": "string",
  "redirectUrl": "string",
  "redirectAutomatically": true,
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string",
- "items": [
   + { ... },
```

# Crowdfund

# Get crowdfund app data

Returns crowdfund app data

AUTHORIZATIONS: API\_Key AND Basic

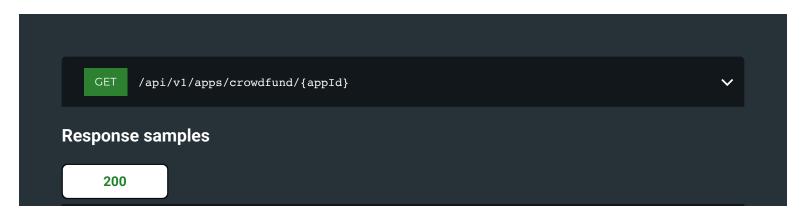
PATH PARAMETERS

→ appId string required App ID

## Responses

> 200 Crowdfund app data

- 404 Crowdfund app with specified ID was not found



# **Content type** application/json

Copy Expand all Collapse all "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s", "appName": "my test app", "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776", "created": 1651554744, "appType": "PointOfSale", "archived": false, "title": "My crowdfund app", "description": "My crowdfund description", "enabled": true, "enforceTargetAmount": false, "startDate": 768658369, "endDate": 771336769, "targetCurrency": "BTC", "targetAmount": 420.69, "mainImageUrl": "string", "notificationUrl": "string", "tagline": "I can't believe it's not butter", "disqusEnabled": true, "disqusShortname": "string", "soundsEnabled": false, "animationsEnabled": true, "resetEveryAmount": 1, "resetEvery": "Day", "displayPerksValue": false, "sortPerksByPopularity": true, - "sounds": [

"https://github.com/ClaudiuHKS/AdvancedQuakeSounds/raw/master/sound/AC

https://docs.btcpayserver.org/API/Greenfield/v1/

],

# Create a new Crowdfund app

```
API_Key ( btcpay.store.canmodifystoresettings ) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS

→ storeId

                       string (StoreId)
   required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
REQUEST BODY SCHEMA: application/json
⊣ id
                            string
                            Id of the app
→ appName
                            string
                            Name given to the app when it was created
 ⊣ storeId
                            string
```

	Id of the store to which the app belongs
⊣ created	integer UNIX timestamp for when the app was created
⊣ аррТуре	string Type of the app which was created
⊣ archived	boolean Nullable Default: false If true, the app does not appear in the apps list by default.
⊣ title	string Nullable Display title of the app
→ description	string Nullable App description
⊣ enabled	boolean Nullable Whether the app is enabled to be viewed by everyone
→ enforceTargetAmount	boolean Nullable Whether contributions over the set target amount are allowed
⊣ startDate	number <int32> Nullable UNIX timestamp for crowdfund start time (https://www.unixtimestamp.com/)</int32>
→ endDate	number <int32> Nullable UNIX timestamp for crowdfund end time (https://www.unixtimestamp.com/)</int32>
→ targetCurrency	string Nullable Target currency for the crowdfund
⊣ targetAmount	number Nullable Target amount for the crowdfund
⊣ mainImageUrl	string Nullable URL for image used as a cover image for the app

→ notificationUrl	string Nullable Callback notification url to POST to once when invoice is paid for and once when there are enough blockchain confirmations
— tagline	string Nullable Tagline for the app displayed to user
⊣ disqusEnabled	boolean Nullable Whether Disqus is enabled for the app
⊣ disqusShortname	string Nullable Disqus shortname to used for the app
→ soundsEnabled	boolean Nullable Whether sounds on new contributions are enabled
→ animationsEnabled	boolean Nullable Whether background animations on new contributions are enabled
→ resetEveryAmount	number Nullable Contribution goal reset frequency amount
⊣ resetEvery	string Nullable Contribution goal reset frequency
⊣ displayPerksValue	boolean Nullable Whether perk values are displayed
→ sortPerksByPopularity	boolean Nullable Default: true Whether perks are sorted by popularity
⊣ sounds	Array of strings Nullable Array of custom sounds which can be used on new contributions
→ animationColors	Array of strings Nullable Array of custom HEX colors which can be used for background animations on new contributions
→ htmlLang	string Nullable

	Used for SEO, the HTML Lang of the page
⊣ htmlMetaTags	string Nullable
	Used for SEO, the Meta tags of the page
⊣ formId	string Nullable
	Form ID to request customer data
— perksTemplate	string JSON of perks available in the app

## Responses

```
> 200 Created app details
> 422 Unable to validate the request
```

```
POST /api/v1/stores/{storeId}/apps/crowdfund

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

{
    "id": "3ki4jsAkN4u9rv1PUzjlodX4Nx7s",
    "appName": "my test app",
    "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
    "created": 1651554744,
```

```
"appType": "PointOfSale",
  "archived": false,
  "title": "My crowdfund app",
  "description": "My crowdfund description",
  "enabled": true,
  "enforceTargetAmount": false,
  "startDate": 768658369,
  "endDate": 771336769,
  "targetCurrency": "BTC",
  "targetAmount": 420.69,
  "mainImageUrl": "string",
  "notificationUrl": "string",
  "tagline": "I can't believe it's not butter",
  "disqusEnabled": true,
  "disqusShortname": "string",
  "soundsEnabled": false,
  "animationsEnabled": true,
  "resetEveryAmount": 1,
  "resetEvery": "Day",
  "displayPerksValue": false,
  "sortPerksByPopularity": true,
- "sounds": [
      "https://github.com/ClaudiuHKS/AdvancedQuakeSounds/raw/master/sound/AC
  ],
- "animationColors": [
      "#FF0000",
      "#00FF00",
      "#0000FF"
  ],
  "htmlLang": "en",
  "htmlMetaTags": "<meta name=\"description\" content=\"Your description\">
  "formId": "string",
  "perksTemplate": "string"
```

## **Response samples**

200

422

```
Content type
application/json
                                                        Expand all Collapse all
                                                 Copy
  "id": "3ki4jsAkN4u9rv1PUzj1odX4Nx7s",
  "appName": "my test app",
  "storeId": "9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776",
  "created": 1651554744,
  "appType": "PointOfSale",
  "archived": false,
  "title": "My crowdfund app",
  "description": "My crowdfund description",
  "enabled": true,
  "enforceTargetAmount": false,
  "startDate": 768658369,
  "endDate": 771336769,
  "targetCurrency": "BTC",
  "targetAmount": 420.69,
  "mainImageUrl": "string",
  "notificationUrl": "string",
  "tagline": "I can't believe it's not butter",
  "disqusEnabled": true,
  "disqusShortname": "string",
  "soundsEnabled": false,
  "animationsEnabled": true,
  "resetEveryAmount": 1,
  "resetEvery": "Day",
  "displayPerksValue": false,
  "sortPerksByPopularity": true,
- "sounds": [
      "https://github.com/ClaudiuHKS/AdvancedQuakeSounds/raw/master/sound/AC
  ],
```

# Stores (Payouts)

# Create Payout

Create a new payout

```
API_Key( btcpay.store.cancreatepullpayments, btcpay.store.cancreatenonapprovedpullpayments ) AND Basic

PATH PARAMETERS

— storeId string (StoreId) required
```

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776 The store ID

#### REQUEST BODY SCHEMA: application/json

destination string

The destination of the payout (can be an address or a BIP21 url)

- amount string <decimal>

The amount of the payout in the currency of the pull payment (eg. USD).

→ payoutMethodId string (PayoutMethodId)

Payout method IDs. Available payment method IDs for Bitcoin are:

• "BTC-CHAIN": Onchain

• "BTC-LN" : Lightning

→ pullPaymentId string

The pull payment to create this for. Optional.

→ approved boolean

Whether to approve this payout automatically upon creation

⊣ metadata object

Additional metadata to store with the payout

## Responses

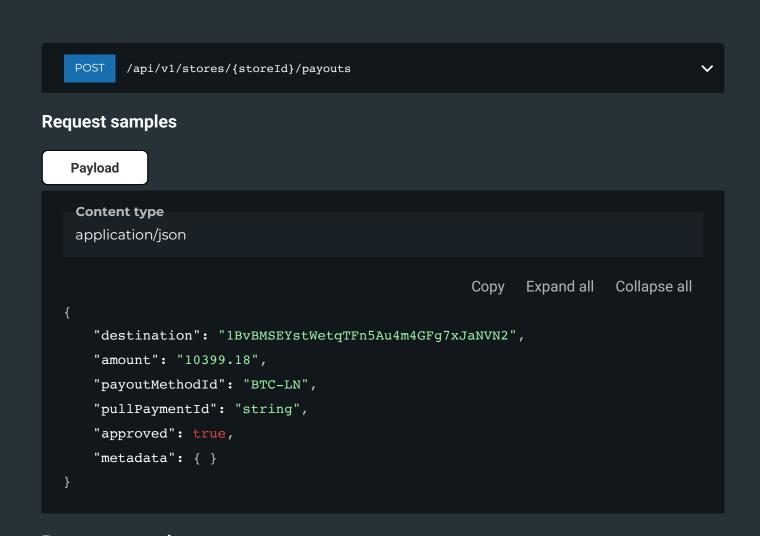
> 200 A new payout has been created

#### > 400

Wellknown error codes are: duplicate-destination , expired , not-started , archived , overdraft , amount-too-low , payment-method-not-supported

- 404 store not found

> 422 Unable to validate the request



## **Response samples**

```
Content type
application/json

Copy Expand all Collapse all

{
    "id": "string",
    "revision": 0,
    "pullPaymentId": "string",
    "date": "string",
    "date": "string",
    "date": "string",
    "date": "string",
    "date": "lbvbMSEYstWetqTFn5Au4m4GFg7xJaNVN2",
    "originalCurroper": "USD"
```

```
originalcurrency
  "originalAmount": "10399.18",
  "payoutCurrency": "BTC",
  "payoutAmount": "1.12300000",
  "payoutMethodId": "BTC-LN",
  "state": "AwaitingPayment",
- "paymentProof": {
      "id": "string"
  },
- "metadata": {
      "source": "Payout created through the API"
```

# Get Store Payouts

Get payouts

#### PATH PARAMETERS

→ storeId required string (StoreId)

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

#### **QUERY PARAMETERS**

→ includeCancelled boolean

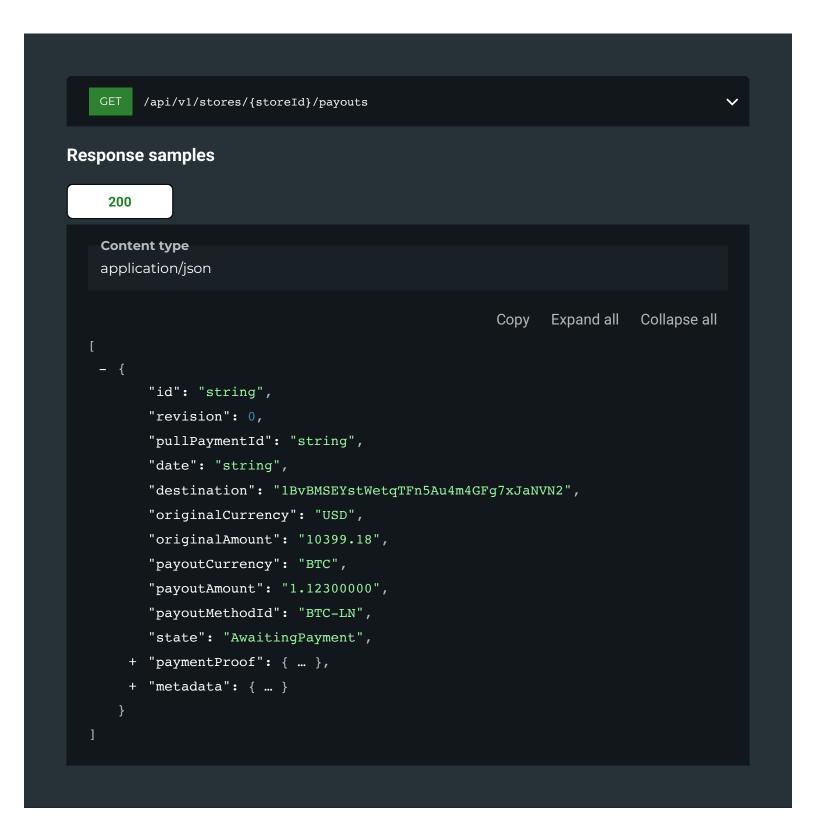
Default: false

Whether this should list cancelled payouts

## Responses

> 200 The payouts of the store

- 404 Pull payment not found



# Get Payout

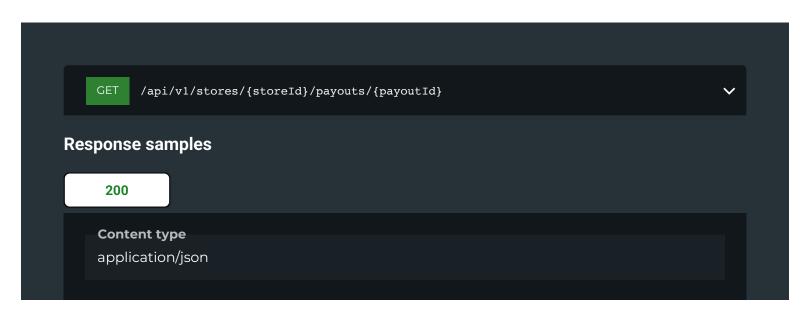
## Get payout

AUTHORIZATIONS:	API_Key( btcpay.store.canmanagepullpayments) AND Basic
PATH PARAMETERS	
⊣ storeId required	string (StoreId)  Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776  The store ID
→ payoutId required	string The ID of the payout

## Responses

> 200 A specific payout of a store

— 404 Payout not found



```
Copy
                                                        Expand all
                                                                   Collapse all
  "id": "string",
  "revision": 0,
  "pullPaymentId": "string",
  "date": "string",
  "destination": "1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2",
  "originalCurrency": "USD",
  "originalAmount": "10399.18",
  "payoutCurrency": "BTC",
  "payoutAmount": "1.12300000",
  "payoutMethodId": "BTC-LN",
  "state": "AwaitingPayment",
- "paymentProof": {
      "id": "string"
  },
- "metadata": {
      "source": "Payout created through the API"
```

# Approve Payout

Approve a payout

API\_Key (|btcpay.store.canmanagepullpayments|) AND Basic

PATH PARAMETERS

string (StoreId)

required

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

required string
The ID of the payout

REQUEST BODY SCHEMA: application/json

revision integer
The revision number of the payout being modified

rateRule string Nullable
The rate rule to calculate the rate of the payout. This can also be a fixed decimal. (if null or unspecified, will use the same rate setting as the store's settings)

## Responses





```
Copy Expand all Collapse all

{
    "revision": 0,
    "rateRule": "kraken(BTC_USD)"
}
```

## **Response samples**

```
200
 Content type
 application/json
                                                   Copy
                                                          Expand all
                                                                     Collapse all
   "id": "string",
   "revision": 0,
    "pullPaymentId": "string",
    "date": "string",
    "destination": "1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2",
    "originalCurrency": "USD",
    "originalAmount": "10399.18",
    "payoutCurrency": "BTC",
    "payoutAmount": "1.12300000",
    "payoutMethodId": "BTC-LN",
    "state": "AwaitingPayment",
 - "paymentProof": {
       "id": "string"
   },
 - "metadata": {
       "source": "Payout created through the API"
```

# Cancel Payout

#### Cancel the payout

API\_Key ( btcpay.store.canmanagepullpayments ) AND Basic

PATH PARAMETERS

storeId string (StoreId)
Example: 9CinzkoAnxxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
The store ID

payoutId string required The ID of the payout

## Responses

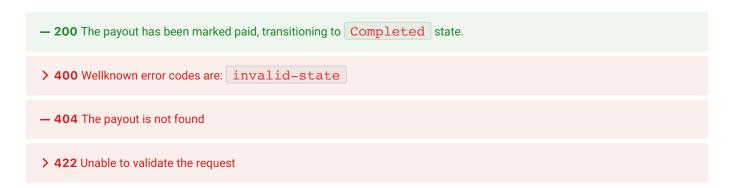
200 The payout has been cancelled404 The payout is not found



# Mark Payout as Paid

#### Mark a payout as paid

## Responses





```
{
    "code": "string",
    "message": "string"
}
```

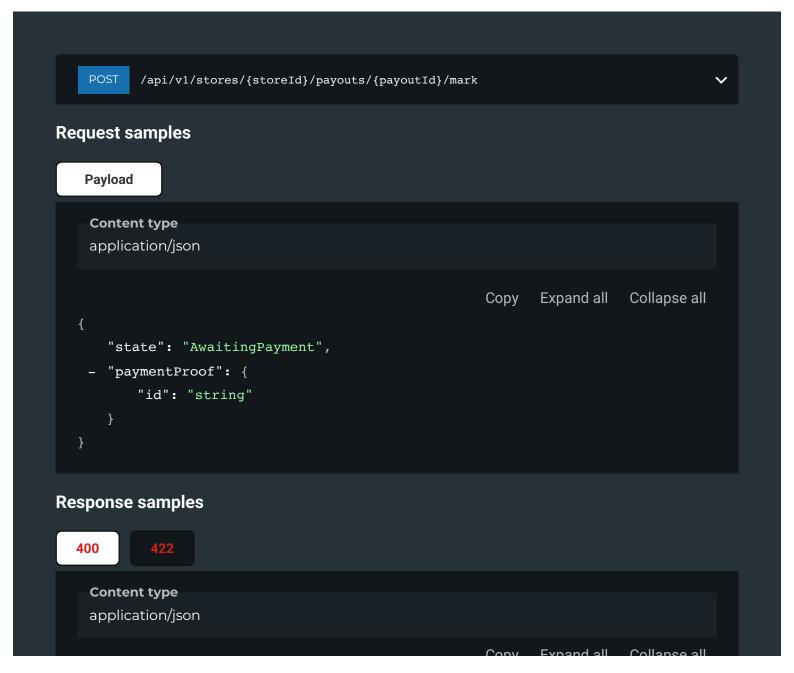
# Mark Payout

Mark a payout with a state

```
API_Key (|btcpay.store.canmanagepullpayments|) AND Basic
AUTHORIZATIONS:
PATH PARAMETERS
 ⊣ storeId
                       string (StoreId)
   required
                       Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776
                       The store ID
 ⊣ payoutId
                       string
   required
                       The ID of the payout
REQUEST BODY SCHEMA:
                        application/json
 ⊣ state
                       string (PayoutState)
                       Enum: "AwaitingApproval" "AwaitingPayment" "Cancelled"
                        "Completed" "InProgress"
                       The state of the payout ( AwaitingApproval , AwaitingPayment ,
                        InProgress , Completed , Cancelled )
 paymentProof >
                       object or object (PayoutPaymentProof)
                       Additional information around how the payout is being or has been paid out.
                       The mentioned properties are all optional (except proofType) and you can
                       introduce any json format you wish.
```

## Responses

```
200 The payout has been set to the specified state
400 Wellknown error codes are: invalid-state
404 The payout is not found
422 Unable to validate the request
```



```
"copy Expand an Conapse an

{
    "code": "string",
    "message": "string"
}
```

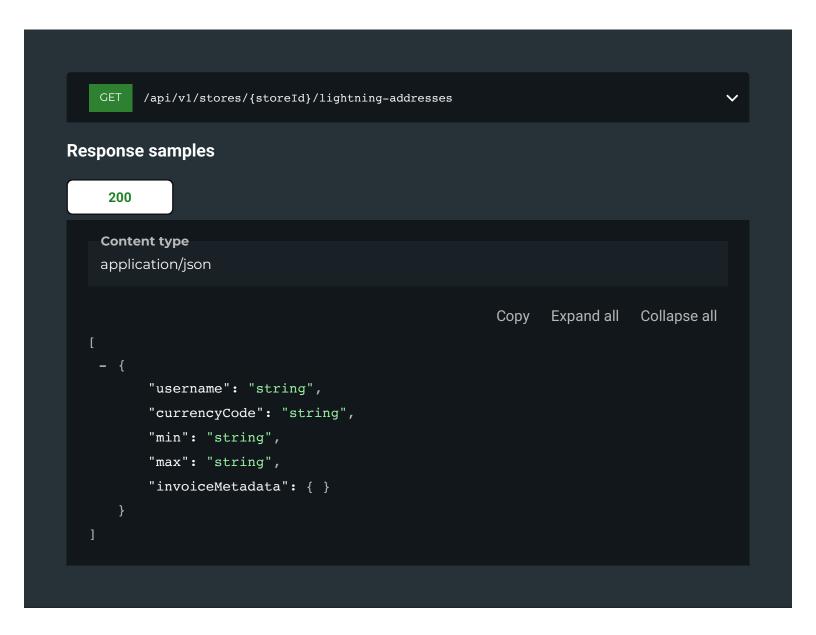
# Lightning address

# Get store configured lightning addresses

Get store configured lightning addresses

## Responses

200 The lightning addresses configured in the store
 403 If you are authenticated but forbidden to view the specified store
 404 The key is not found for this store/wallet



# Get store configured lightning address

Get store configured lightning address

```
API_Key ( btcpay.store.canviewstoresettings ) AND Basic

PATH PARAMETERS

string (StoreId)
```

```
required

Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776

The store ID

username string
required

The lightning address username
```

## Responses

```
    200 The lightning address configured in the store
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store/wallet
```

```
Response samples

Content type
application/json

Copy Expand all Collapse all

{
  "username": "string",
  "currencyCode": "string",
  "min": "string",
  "max": "string",
  "invoiceMetadata": { }
}
```

# Add or update store configured lightning address

Add or update store configured lightning address

AUTHORIZATIONS:	API_Key( btcpay.store.canmodifystoresettings ) AND Basic
PATH PARAMETERS	
→ storeId required	string (StoreId)  Example: 9CiNzKoANXxmk5ayZngSXrHTiVvvgCrwrpFQd4m2K776  The store ID
→ username  required	string the lightning address username
REQUEST BODY SCHEMA:	application/json
⊣ username	string The username of the lightning address
⊣ currencyCode	string Nullable The currency to generate the invoices for this lightning address in. Leave null Ito use the store default.
→ min	string Nullable The minimum amount in sats this In address allows
⊣ max	string Nullable The maximum amount in sats this In address allows
⊣ invoiceMetadata	object Nullable The invoice metadata as JSON.

## Responses

```
    200 The lightning address configured in the store
    403 If you are authenticated but forbidden to view the specified store
    404 The key is not found for this store/wallet
```

