



# **CUBITS**

API DOCUMENTATION

Version 1.8

# Table of Contents

Table of Contents	1
Introduction	2
Request and Response Format	4
Authentication	7
Callbacks	10
User language selection	14
Test	15
Invoices	17
Channels	25
Quote Channels	38
Accounts	52
Send Money	54
Quotes	56
Buy	59
Buysend	62
Sell	65
Appendices	68

# Introduction

---

This document describes the Cubits REST API and everything that is necessary to access its resources. In order to use the API, you first need to have a verified Cubits merchant account. If you did not receive your credentials yet, please sign up and verify your merchant account at [Cubits](#) or contact [support@cubits.com](mailto:support@cubits.com).

## Configuration

---

The base URL host for all API requests documented below is:

```
https://api.cubits.com
```

All API requests are performed over HTTPS and follow [JSON API](#) conventions. All data is sent and received as JSON with the content type `application/vnd.api+json`.

## Versioning

---

The Cubits REST API follows a major.minor versioning scheme. All request formats, headers and semantics are backwards compatible within the same major version, which can also be identified by looking at the request path. Newer minor versions may add additional headers or fields in the responses or allow for more valid parameter combinations in the requests. It is safe to use implementations for a given major version of the API with any future minor versions. The major version of the REST API (also visible in the request path) is reserved for non backwards-compatible changes and in general you cannot use an implementation of the API for different major versions.

## Changelog

---

Version	Date	Changes
v1.0	2014-11-03	Initial version
v1.1	2015-01-09	Added <code>send_money</code> call
v1.2	2015-02-09	Added <code>accounts</code> , <code>quotes</code> , <code>buy</code> and <code>sell</code> calls
v1.3	2015-05-20	Added <code>channels</code> and signed callbacks
v1.4	2015-06-03	Introduce new request header format
v1.5	2015-10-26	Additional <code>quotes</code> functionality, restructure documentation
v1.6	2016-01-20	Added <code>quote_channels</code>
v1.7	2016-11-07	Added <code>buysend</code> APIs. Extends channels options with <code>share_to_keep_in_btc</code> and <code>amount_before_fees</code>
v1.8	2017-10-12	Changed invoices defaults, added language selection section

# Request and Response Format

---

All API requests are performed over HTTPS and follow [JSON API](#) conventions. All data is sent and received as JSON with the content type `application/vnd.api+json` .

## Authentication Headers

---

All requests to the API *MUST* include the following headers for authentication:

```
X-Cubits-Key: ...  
X-Cubits-Nonce: ...  
X-Cubits-Signature: ...
```

Using curl:

```
curl -H "X-Cubits-Key: ..." -H "X-Cubits-Nonce: ..." -H "X-Cubits-Signature: ..." ...
```

See [Authentication](#) section for a detailed description.

## GET Requests

---

It is recommended to set the "Accept" header as follows:

```
Accept: application/vnd.api+json
```

Using curl:

```
curl -H "Accept: application/vnd.api+json" ...
```

## POST Requests

---

POST requests *MUST* send data in JSON format within request body and have a header `Content-Type: application/vnd.api+json` :

```
POST /api/v1/test  
Content-Type: application/vnd.api+json  
Accept: application/vnd.api+json  
  
{  
  "foobar": "1x Foo + 1x Bar",  
  "foo": 12345,  
  "bar": 0.75  
}
```

Using curl:

```
curl -H "Content-Type: ..." -H "Accept: ..." -X POST \  
-d '{"foobar": "1x Foo + 1x Bar", "foo": 12345, "bar": 0.75}' ...
```

## Responses

All API server responses are in JSON format with a Content-Type `application/vnd.api+json` :

```
HTTP/1.1 200 OK  
Content-Type: application/vnd.api+json  
  
{  
  "status": "success"  
}
```

Appropriate HTTP status codes are used both for successful processing and in case of errors.

## Success Responses

Successful responses have one of the following HTTP status codes:

Situation	HTTP status code
Request was accepted, validated and processed	200 OK
As above, plus a resource was created as a result	201 Created

## Error Responses

In case of any error the server responds with an appropriate HTTP status code and a JSON body:

```
HTTP/1.1 403 Forbidden  
Content-Type: application/vnd.api+json  
  
{  
  "message": "Invalid signature"  
}
```

All error responses have the following format:

key	type	description
message	String	Human readable description of the error

## Client Errors (4xx)

Situation	HTTP status code	message
Key is not present	400 Bad Request	Key is missing
Signature is not present	400 Bad Request	Signature is missing
Nonce is not present	400 Bad Request	Nonce is missing
Nonce is already used	400 Bad Request	Invalid nonce
Invalid request parameters	400 Bad Request	Invalid parameters
for some requests, more specific messages	400 Bad Request	<i>message</i>
Key or signature is invalid	403 Forbidden	Invalid signature
Invalid/unknown path	404 Not Found	Not found
Unsupported HTTP method	404 Not Found	Not found
Content-Type not application/vnd.api+json	415 Unsupported Media Type	Invalid Content-Type

### Server Errors (5xx)

Situation	HTTP status code	message
Any internal error/exception	500 Internal Server Error	Internal Server Error

# Authentication

---

To use the Cubits API you have to first create an API token (key + secret) in your [Cubits merchant account](#). All requests to the API have to be authenticated using this token information in the headers.

Each valid authenticated request has to include the following HTTP headers:

```
X-Cubits-Key: ...  
X-Cubits-Nonce: ...  
X-Cubits-Signature: ...
```

## Cubits API key: X-Cubits-Key

---

`X-Cubits-Key` is the API access key which you receive when you generate an API access token. It is a sequence of hex-digits represented as a string, randomly generated when a new key is created in the web interface.

API access keys are case-sensitive.

### Example

```
X-Cubits-Key: 549653887407b9b8ad66d4b47093eb9f
```

## Nonce: X-Cubits-Nonce

---

`X-Cubits-Nonce` is a 64-bit **integer** number, which you must generate for every request you make to the API. This *nonce* ("number used once") has to meet the following two requirements:

1. *nonce* must be unique for every request you make with the same API access key, ever. If you make a request with the same API access key and nonce again, it will be rejected.
2. Every *nonce* that you generate for the request, has to be greater than any of the previous nonces that you used to make requests to the API. There is no way to reset the nonce value for a given API key but you can always just generate a new API key.

One way to generate nonces is to use the UNIX epoch timestamp of the request. Be sure, though, that you use enough precision: if you use only the *seconds* part of the timestamp and you send two requests to the API within the same second, one of them will be rejected. It is recommended that you use the UNIX epoch to microsecond resolution. The nonce value must be representable as an unsigned 64-bit integer therefore it has to be within the range [0..18446744073709551615]

### Example

```
X-Cubits-Nonce: 1411754081462609
```

## Cubits API secret: CUBITS\_SECRET

---



The secret is a randomly generated 64 character long string from the Base62 character set:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789 .
```

Upon creation of a new API access token, the secret is displayed in the web-interface. Since it is only displayed once, you should write the secret down or copy it into your application code immediately. You can however create new API keys at every point in time.

## Example

```
CUBITS_SECRET: 0J7C3DBkMScVk89fJ11pKFujspSP9aa4KVnGa3DGVQXUA51TaBK4eWtONQEg5pAX
```

## Signature: X-Cubits-Signature

The signature is derived from the API access secret (CUBITS\_SECRET), the nonce (X-Cubits-Nonce) and the request body. Its format is a **hex-string** representation of the result of the following hash calculation:

```
HMAC-SHA512(k, msg)
```

Where:

`k` is your API access secret (CUBITS\_SECRET) as UTF-8 string

`msg` is a UTF-8 string, constructed by string concatenation of `uri_path` + `nonce_s` + `SHA256(request_data)`

`uri_path` is the *path* part of the request URI as UTF-8 string

`nonce_s` is the *nonce* (X-Cubits-Nonce) of the request, converted to a UTF-8 string

`SHA256(request_data)` is the hex-encoded SHA256 digest of `request_data`, as a UTF-8 string, **downcased**

`request_data` is either the **JSON** encoded request body in case of POST requests, or the URL-encoded query in case of a GET request as specified in [RFC3986](#)

## Example 1

Your API access key is `7287ba0902461025b01d5b99e4679018`

Your API access secret is

```
93yJJ8LBDe3zNSewHBdX1XIQDjCMDIn0EKNnXrd3kfzL72fvLz99uKnXFLYuCfkt
```

You are making a POST request to `/api/v1/test`, the nonce is `123` and the POST request body is

```
{"attr1": 123, "attr2": "hello"}
```

`msg` is then

```
/api/v1/test123947753ba472927154c534cf2e4e11de27ed7a9560dc033e77d6cc24ee950ea56
```

*X-Cubits-Signature* is

```
d3cb2a18b754994ea7dcdc4d46cb89cb538d6533155a48f6953296680a1dc2cf7476ce7c194b2cb38231fe7!  
fa14799b976ea61b0190afadaffe53434ea56bf
```

## Example 2

Your API access key is `3cd7a0db76ff9dca48979e24c39b408c`

Your API access secret is

`M2NkN2EwZGI3NmZm0WRjYTQ40Tc5ZTI0YzM5YjQwOGMgIC0KM2NkN2EwZGI3NmZm`

You are making a GET request to `/api/v1/info`, the nonce is `4711` and the URL-encoded query part of the GET request URI is `first=this+is+a+field&second=was+it+clear+%28already%29%3F`

`msg` is then

`/api/v1/info471121638dfe9dd465f4eb5e31be96cebc0e1baf0966b6378949cf3653c04ad8de00`

`X-Cubits-Signature` is

`24c2a83c15581c85de5b180716bd8e86467c089665d6ab51bd6e979815e9e740a74a265d9b2aaee3db91467c583254d64280b1fbdf1e8cf91bf98ef09aff114`

## Reference Implementations

You can find open source code samples as well as full library implementations on our [Cubits github page](#).

### Ruby 1.9+

```
require 'openssl'

CUBITS_SECRET = '93yJJ8LBDe3zNSewHBdX1XIQDjCMDIn0EKNnXrd3kfzL72fvLz99uKnXFLYuCfkt'

def generate_signature(path, nonce, request_data)
  msg = path + nonce.to_s + OpenSSL::Digest::SHA256.hexdigest(request_data)
  OpenSSL::HMAC.hexdigest('sha512', CUBITS_SECRET, msg)
end

generate_signature('/api/v1/test', 123, '{"attr1": 123, "attr2": "hello"}') # => d3cb2a18
```

### Python 2.7

```
import hmac, hashlib

CUBITS_SECRET = '93yJJ8LBDe3zNSewHBdX1XIQDjCMDIn0EKNnXrd3kfzL72fvLz99uKnXFLYuCfkt'

def generate_signature(path, nonce, request_data=''):
  msg = path + str(nonce) + hashlib.sha256(request_data).hexdigest()
  return hmac.new(CUBITS_SECRET, msg=msg, digestmod=hashlib.sha512).hexdigest()

generate_signature('/api/v1/test', 123, '{"attr1": 123, "attr2": "hello"}') # => d3cb2a18
```

# Callbacks

## General

Specifying a `callback_url` when creating a resource supporting callbacks (e.g. an `invoice`, `channel` or `quote_channel`), will trigger a POST request to this URL upon any change of an attribute of the respective resource. The request body of the callback will contain the same output that a GET request to the updated resource would return right after the event that triggered the update.

## Example

Invoice creation:

```
POST /api/v1/invoices HTTP/1.1

{
  "currency": "EUR",
  "price": "266.45",
  "name": "Guatemala Joe",
  "description": "Twelve ounces of high quality, organic Guatemalan coffee",
  "reference": "d0cef2",
  "callback_url": "https://my.shop.online/callback"
}
```

Callback after payment:

```
{
  "id": "378d8ec6e305f469b009cb4e2deedf93",
  "status": "completed",
  "address": "34MajouDKKL5DKKwy6Lk5zkLf84B88TPzv",
  "merchant_currency": "EUR",
  "merchant_amount": "266.45",
  "invoice_currency": "BTC",
  "invoice_amount": "0.88613839",
  "paid_currency": "BTC",
  "paid_amount": "0.88613839",
  "pending_currency": "BTC",
  "pending_amount": "0.00000000",
  "name": "Guatemala Joe",
  "description": "Twelve ounces of high quality, organic Guatemalan coffee",
  "reference": "d0cef2",
  "invoice_url": "https://pay.cubits.com/invoice/378d8ec6e305f469b009cb4e2deedf93",
  "callback_url": "https://my.shop.online/callback",
  "success_url": "https://my.shop.online/success",
  "cancel_url": "https://my.shop.online/cancel",
  "notify_email": "payments@my.shop.online",
  "create_time": 1398871897.0,
  "valid_until_time": 1398872497.0
}
```

## Authenticating Callbacks

To provide authentication for the callback, the Cubits API signs the POST request body with the API key and

secret, which were used to create or last update the resource. The signature and API key is then passed in the callback headers, together with a callback ID.

## POST Request Headers

Header	Data type	Description
X-Cubits-Callback-Id	string(8)	Unique identifier of the callback request
X-Cubits-Key	string(32)	<i>(optional)</i> API key which was used to create the resource
X-Cubits-Signature	string(128)	<i>(optional)</i> Signature of the request body

If the resource was not created using an API key (for example invoices created by *payment buttons*), the Cubits API does not sign the callback and therefore the headers `X-Cubits-Key` and `X-Cubits-Signature` are omitted.

## X-Cubits-Signature

The signature is derived from the API access secret (CUBITS\_SECRET) that was used to create the resource, a unique identifier of the callback (X-Cubits-Callback-Id) and the request body. It is represented as a **hex-string** of the result of the following hash calculation:

```
HMAC-SHA512(k, msg)
```

Where:

`k` is your API access secret (CUBITS\_SECRET) as UTF-8 string

`msg` is a UTF-8 string, constructed by string concatenation of `X-Cubits-Callback-Id` + `SHA256(request-data)`

`X-Cubits-Callback-Id` is a UTF-8 string value of the callback unique identifier passed in the request headers

`SHA256(request-data)` is the hex-encoded SHA256 digest of `request-data`, as a UTF-8 string, **downcased**

`request-data` is the **JSON** encoded request body

## Example

You have created an invoice using your API access key `7287ba0902461025b01d5b99e4679018`

API access secret for this key is

```
93yJJ8LBDe3zNSewHBdX1XIQDjCMDIn0EKNnXrd3kfzL72fvLz99uKnXFLYuCfkt
```

You have received the callback with `X-Cubits-Callback-Id` being `ABCDEFGH` and the POST request body is `{"attr1": 123, "attr2": "hello"}`

`msg` is then `ABCDEFGH947753ba472927154c534cf2e4e11de27ed7a9560dc033e77d6cc24ee950ea56`

`X-Cubits-Signature` for this callback should then be

```
7d89c35c2e0840867f63b77ea575050db21a134b674d4a38f1e255518efb5b81383442cd9a888dca86dfe3e  
a0769525088aac3efed3102a6b14bd1446f14a1
```

It is very important that you verify the callback authenticity and ensure that the callback was actually posted by Cubits. Otherwise it is quite simple for some attacker to forge a callback and to trick you into believing that the payment was made, whereas it was not.

There are two ways to achieve this:

1. Check the callback signature, provided in the callback headers
2. After receiving a callback, get the resource `id` and retrieve the actual resource details with a `GET` request.

## Example

```
POST /your/callback/path HTTP/1.1
Content-Type: application/vnd.api+json
X-Cubits-Callback-Id: A7PPGKYM
X-Cubits-Key: 0086bf7149ac69e05ec1808b9f187a10
X-Cubits-Signature: 941923cb09ec43b7b2453ccd4d7196794b9c699157a4c69166d6476108e75bf3...

{
  "merchant_currency": "EUR",
  "merchant_amount": "12.34",
  "invoice_currency": "BTC",
  "invoice_amount": "0.05348733",
  "paid_currency": "BTC",
  "paid_amount": "0.00000000",
  "pending_currency": "BTC",
  "pending_amount": "0.00000000",
  "share_to_keep_in_btc": 0,
  "name": null,
  "description": null,
  "reference": null,
  "callback_url": "https://my.shop.online/your/callback/path",
  "success_url": null,
  "cancel_url": null,
  "notify_email": null,
  "id": "a5fe404f40d628e65ff2d739a63a80f7",
  "status": "pending",
  "invoice_url": "https://pay.cubits.com/invoices/a5fe404f40d628e65ff2d739a63a80f7",
  "address": "34MDjoUDkKL5DKKwy6Lk5zkLf84B88TPzv",
  "valid_until_time": 1426520056.0,
  "create_time": 1426519456.0
}
```

## Callback Retry

The Cubits callback API will only consider a callback as successfully sent when it receives a 2xx HTTP status code (200-299) from your application that is processing the callback POST request. In case your application is currently unable to process the callback or if there is some error, it should return with a different status code, letting Cubits know that it should retry the callback. By that we make sure that updates to the status of a resource is not lost even in case of some downtime of your application.

The callback retries will happen in the following increasing time intervals:

<b>Retry attempt</b>	<b>Interval</b>	<b>Seconds after initial callback</b>
1	1 second	1
2	5 seconds	6
3	10 seconds	16
4	30 seconds	46
5	2 minutes	166
6	15 minutes	1066
7	1 hour	4666
8	2 hours	11866
9	12 hours	55066
10	24 hours	141466
11	7 days	746266
12	14 days	1955866

# User language selection

English is the default language for the payment screen. If you would like to change the language that is displayed to customers, you can modify the redirect URL with the lang parameter, as below:

```
# Notice '?lang=es' part  
https://pay.cubits.com/invoice/378d8ec6e305f469b009cb4e2deedf93?lang=es
```

Value of `lang` parameter should be from [ISO 639-1](#) list.

Currently supported following languages

Parameter value	Language
en	English
ja	Japanese
ko	Korean
zh	Chinese
es	Spanish
ru	Russian
no	Norwegian
sv	Swedish
fi	Finnish
tr	Turkish

# Test

---

## GET /api/v1/test

---

This request is intended to be used to test if your application is configured properly and can access the Cubits API using GET requests.

### Request

Any

### Response

On success API responds with HTTP status `200 OK` and following attributes:

Attribute	Data type	Description
status	string(7)	"success"

### Errors

On error, the API responds with standard [error responses](#).

### Example

Request:

```
GET /api/v1/test HTTP/1.1
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
}
```

---

## POST /api/v1/test

---

This request is intended to be used to test if your application is configured properly and can access the Cubits API using POST requests.



## Request

There are no required parameters, but you can pass any parameters to test if your signature calculation is correct.

## Response

On success API responds with HTTP status `200 OK` and following attributes:

Attribute	Data type	Description
status	string(7)	"success"

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
POST /api/v1/test HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "foobar": 123.0,
  "foo": null,
  "bar": true
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "status": "success"
}
```

# Invoices

Cubits invoices are a convenient way to accept Bitcoin payments using Cubits. An invoice is created specifying a certain amount and a currency you want to receive. This currency can be any of the supported fiat currencies or Bitcoin. In any case, a quote is created and the corresponding price in Bitcoin is calculated. Each invoice has a unique Bitcoin address and a payment screen URL with a QR code and additional features that can be used to show to a customer. The Cubits payment screen also supports logging in to a Cubits account directly to facilitate single-click payments for Cubits users. Each invoice has a limited validity period (for fiat invoices: 15 minutes) during which they can be paid and the (optional) conversion to fiat will be performed. Bitcoin payments arriving after the validity period will be treated as regular deposits.

Our system operates with two types of invoices: refundable and non-refundable. Non-refundable invoices always deposit any funds that are left after conversion. This includes underpaid unsuccessful invoices as well as overpaid amounts. Refundable invoices are different in a way that such excess funds are claimable by customer who received invoice link.

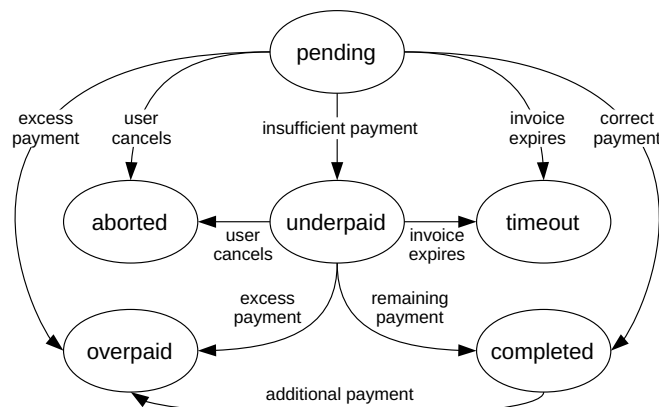
Since version 1.8 of API, newly created merchant accounts will have invoices refundable by default. If your account was created before and you wish to have switch refundable invoices, please contact you account manager.

## Invoice status

Status	Explanation
pending	Initial state, invoice has been created but no payment was received yet
completed	Success state, invoice has been fully paid
overpaid	Success state, invoice has been fully paid but additional funds were received
underpaid	Intermediary state, insufficient payment was received
aborted	Failure state, payment was cancelled by the customer
timeout	Failure state, no sufficient payment was received before the invoice expired

All invoices with `underpaid` status can be completed, overpaid or timeout depending on user actions.

## State transitions



## GET /api/v1/invoices/{invoice\_id}

---

Get information about an existing invoice.

### Request

None

### Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the invoice
status	string(11)	"pending", "completed", "overpaid", "aborted", "timeout" or "underpaid"
address	string(34)	Bitcoin address displayed to the customer
merchant_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that the merchant wants to receive (e.g. "EUR")
merchant_amount	string(16)	Amount of currency the merchant wants to receive, as a decimal floating point number, converted to string (e.g. "123.05")
invoice_currency	string(3)	ISO code of the currency that the customer was requested to pay (e.g. "BTC")
invoice_amount	string(16)	Amount of the invoice currency that the customer was requested to pay
paid_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was paid by the customer
paid_amount	string(16)	Amount of currency that was paid by the customer
pending_currency	string(3)	<a href="#">ISO 4217</a> code of the pending amount currency
pending_amount	string(16)	Total amount of the unconfirmed incoming payments that were sent by the customer
share_to_keep_in_btc	number	Percentage of the invoice amount to be kept in BTC, as an integer number from 0 to 100
name	string(256)	Name of the item displayed to the customer
description	string(512)	Description of the item displayed to the customer
reference	string(512)	Individual free-text field stored in the invoice as-is
invoice_url	string(512)	Unique URL of the invoice payment page on Cubits
callback_url	string(512)	URL that is called on invoice status updates
success_url	string(512)	URL to redirect the customer to after a successful payment
cancel_url	string(512)	URL to redirect the customer to if the invoice expires
notify_email	string(256)	e-mail address to be notified about payments
create_time	number	(float) Unix-epoch time of the invoice creation
valid_until_time	number	(float) Unix-epoch timestamp after which the invoice will expire

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
GET /api/v1/invoices/378d8ec6e305f469b009cb4e2deedf93 HTTP/1.1
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "id": "378d8ec6e305f469b009cb4e2deedf93",
  "status": "completed",
  "address": "1AeMbkpHia8FVuKczQKUrV9uMzv7uC1HZi",
  "merchant_currency": "EUR",
  "merchant_amount": "266.45",
  "invoice_currency": "BTC",
  "invoice_amount": "0.88613839",
  "paid_currency": "BTC",
  "paid_amount": "0.88613839",
  "pending_currency": "BTC",
  "pending_amount": "0.00000000",
  "share_to_keep_in_btc": 0,
  "name": "Guatemala Joe",
  "description": "Twelve ounces of high quality, organic Guatemalan coffee",
  "reference": "d0cef2",
  "invoice_url": "https://pay.example.org/invoice/378d8ec6e305f469b009cb4e2deedf93",
  "callback_url": "https://my.shop.online/callback",
  "success_url": "https://my.shop.online/success",
  "cancel_url": "https://my.shop.online/cancel",
  "notify_email": "payments@my.shop.online",
  "create_time": 1398871897.0,
  "valid_until_time": 1398872497.0
}
```

---

## POST /api/v1/invoices

Creates a new invoice.

### Request

Attribute	Data type	Description
currency	string(3)	ISO 4217 code of the currency that the merchant wants to receive (e.g. "EUR")
price	string(16)	Price of the invoice that the merchant wants to receive, as a decimal floating point number, converted to string (e.g. "123.05")
share_to_keep_in_btc	number	(optional) Percentage of the invoice amount to be kept in BTC, as an integer number from 0 to 100. If not specified, a default value is used from the Cubits Pay / Payouts / Percentage Kept in BTC
name	string(256)	(optional) Name of the item displayed to the customer
description	string(512)	(optional) Description of the item displayed to the customer
reference	string(512)	(optional) Individual free-text field stored in the invoice as-is
callback_url	string(512)	(optional) URL that is called on invoice status updates
success_url	string(512)	(optional) URL to redirect the customer to after a successful payment
cancel_url	string(512)	(optional) URL to redirect the customer to if the invoice expires
notify_email	string(256)	(optional) e-mail address to be notified about payments

If any of the optional URLs are omitted in the request, the values set in the merchant profile will be used as defaults.

## Response

On success, a response with HTTP status `201 Created` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the created invoice
status	string(11)	"pending"
address	string(34)	Bitcoin address displayed to the customer
merchant_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that the merchant wants to receive (e.g. "EUR")
merchant_amount	string(16)	Amount of currency the merchant wants to receive, as a decimal floating point number, converted to string (e.g. "123.05")
invoice_currency	string(3)	ISO code of the currency that the customer was requested to pay (e.g. "BTC")
invoice_amount	string(16)	Amount of the invoice currency that the customer was requested to pay
paid_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was paid by the customer
paid_amount	string(16)	Amount of currency that was paid by the customer
pending_currency	string(3)	<a href="#">ISO 4217</a> code of the pending amount currency
pending_amount	string(16)	Total amount of the unconfirmed incoming payments that were sent by the customer
share_to_keep_in_btc	number	Percentage of the invoice amount to be kept in BTC, as an integer number from 0 to 100
name	string(256)	Name of the item displayed to the customer
description	string(512)	Description of the item displayed to the customer
reference	string(512)	Individual free-text field stored in the invoice as-is
invoice_url	string(512)	Unique URL of the invoice payment page on Cubits
callback_url	string(512)	URL that is called on invoice status updates
success_url	string(512)	URL to redirect the customer to after a successful payment
cancel_url	string(512)	URL to redirect the customer to if the invoice expires
notify_email	string(256)	e-mail address to be notified about payments
create_time	number	(float) Unix-epoch time of the invoice creation
valid_until_time	number	(float) Unix-epoch timestamp after which the invoice will expire

## Errors

On error API responds with standard [error responses](#) and with one specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Requested currency is not supported	400 Bad Request	Unsupported currency

## Example

Request:

```
POST /api/v1/invoices HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "currency": "EUR",
  "price": "266.45",
  "name": "Guatemala Joe",
  "description": "Twelve ounces of high quality, organic Guatemalan coffee",
  "reference": "d0cef2"
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "id": "378d8ec6e305f469b009cb4e2deedf93",
  "status": "pending",
  "address": "1AeMbkpHia8FVuKczQKUrV9uMzv7uC1HZi",
  "merchant_currency": "EUR",
  "merchant_amount": "266.45",
  "invoice_currency": "BTC",
  "invoice_amount": "0.88613839",
  "paid_currency": null,
  "paid_amount": null,
  "pending_currency": null,
  "pending_amount": null,
  "share_to_keep_in_btc": 0,
  "name": "Guatemala Joe",
  "description": "Twelve ounces of high quality, organic Guatemalan coffee",
  "reference": "d0cef2",
  "invoice_url": "https://pay.cubits.com/invoice/378d8ec6e305f469b009cb4e2deedf93",
  "callback_url": "https://my.shop.online/callback",
  "success_url": "https://my.shop.online/success",
  "cancel_url": "https://my.shop.online/cancel",
  "notify_email": "payments@my.shop.online",
  "create_time": 1398871897.0,
  "valid_until_time": 1398872497.0
}
```

## Callbacks

When specifying a `callback_url`, invoices will send callbacks to inform you about changes to any of the



invoice fields. See [Callbacks](#) for a general description of the Cubits callback mechanism and format.

Usually we will send a callback within a few seconds of a Bitcoin transaction being propagated on the Bitcoin network even though it is not yet confirmed. In some rare cases we might determine that the risk for a double spend attack is higher and wait until we accept a Bitcoin payment transaction. In those cases you will receive the callback with some delay of the actual payment.

# Channels

---

Cubits channels are a way to receive and potentially convert Bitcoin payments of variable amounts and without a set validity period. Bitcoins sent to a channel can be automatically converted to any of the supported fiat currencies at the spot rate at the time of the payment. Channels have a fixed payment address and support callbacks. A payment screen is also available to provide a convenient way to pay to this address to users. In case a channel is created with a fiat conversion, Bitcoin payments for less than the smallest possible unit for that fiat currency will not be converted but transferred as-is to the Cubits Wallet. Channels also provide a way of tracking individual payments made to them. A list of transactions is recorded with each channel and returned upon querying a channels transactions as well as with each callback. Possible use cases for Cubits channels are accepting donations on a website or just keeping track of payments to a certain address.

## GET /api/v1/channels/{channel\_id}

---

Get information about an existing channel.

### Request

None

### Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the channel
address	string(34)	Bitcoin address associated with this channel
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
name	string(256)	Name of the channel, displayed to the customer on the payment screen
description	string(512)	Description of the item displayed to the customer on the payment screen
reference	string(512)	Individual free-text field stored in the channel as-is
channel_url	string(512)	Unique URL of the channel payment screen on Cubits
callback_url	string(512)	URL that is called on channel updates
txs_callback_url	string(512)	URL that is called on channel transaction updates
success_url	string(512)	URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")
created_at	number	(float) Unix-epoch timestamp of the channel creation
updated_at	number	(float) Unix-epoch timestamp when the channel data was last updated

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
GET /api/v1/channels/af9c1f5ae2c51e105e05e75ab716a2067 HTTP/1.1
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
  "receiver_currency": "EUR",
  "name": "Donation payment channel",
  "description": null,
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "txs_callback_url": "https://example.com/txs_callback",
  "success_url": "https://example.com/thank_you.html",
  "created_at": 1427137017.0,
  "updated_at": 1427138200.0,
  "id": "af9c1f5ae2c51e105e05e75ab716a206",
  "channel_url": "https://pay.cubits.com/channels/af9c1f5ae2c51e105e05e75ab716a206",
  "address": "34MDjoUDkK35DKKwy6Lk5zkLf84B88TPzv",
  "share_to_keep_in_btc": "0"
}
```

## GET /api/v1/channels/{channel\_id}/txs

Get information about all transactions of an existing channel.

### Request

Attribute	Data type	Description
page	number	(optional) Page number to return (default: 1)
per_page	number	(optional) Number of transactions to return per page (max: 1000, default: 100)

### Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
pagination	object	<i>Pagination</i> object for the returned txs array
txs	array	Array of <i>transaction</i> objects of this channel sorted by descending <i>updated_at</i> field

### Pagination Object

Attribute	Data type	Description
page	number	Index of the current page (1 <= page <= page_count)
page_count	number	Number of total pages in the result
per_page	number	Number of entries per page (1 <= per_page <= 1000)
total_count	number	Number of total entries in the result

## Transaction Object

Attribute	Data type	Description
tx_ref_code	string(6)	Unique character string to reference this transaction
channel_id	string(32)	Id of the channel this transaction belongs to
state	string(9)	<code>pending</code> , <code>completed</code> or <code>cancelled</code>
created_at	number	(float) Unix-epoch timestamp of the transaction creation
updated_at	number	(float) Unix-epoch timestamp when the transaction data was last updated
sender	object	Information about the <i>sender</i> part of this transaction
receiver	object	Information about the <i>receiver</i> part of this transaction

Unconfirmed transactions will stay in the `pending` state until Cubits deems the risk to accept them as low enough for them to go in the `completed` state. A transaction that was a double-spend or one that will not confirm for other reasons (e.g. because of the Bitcoin dust limit), will eventually go to `cancelled` state.

## Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was sent
amount	string(17)	Amount that was sent
bitcoin_txid	string(64)	Bitcoin transaction id of this transaction or <code>null</code> in case of a Cubits internal transfer

## Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was received. Note that this will be <code>BTC</code> even for fiat channels in case less than the equivalent of the smallest fiat currency unit was sent.
amount	string(17)	Amount that was received. Note that this amount is preliminary in case of <code>pending</code> transactions and might change once the transaction goes to <code>completed</code> state.
amount_plus_fees	string(17)	Amount that was received plus fees paid. Note that this amount is preliminary in case of <code>pending</code> transactions and might change once the transaction goes to <code>completed</code> state.

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
GET /api/v1/channels/af9c1f5ae2c51e105e05e75ab716a2067/txs HTTP/1.1
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "pagination": {
    "page": 1,
    "page_count": 1,
    "per_page": 100,
    "total_count": 3
  },
  "txs": [
    {
      "tx_ref_code": "MH5AD",
      "channel_id": "af9c1f5ae2c51e105e05e75ab716a2067",
      "state": "completed",
      "created_at": 1427137517.0,
      "updated_at": 1427137623.0,
      "sender": {
        "currency": "BTC",
        "amount": "0.00010000",
        "bitcoin_txid": "9ae311aae58c9150ca22f5b3fa7b64e32bc4e25cf9db11ba6bd28de50bcd1f73"
      },
      "receiver": {
        "currency": "EUR",
        "amount": "0.02",
        "amount_plus_fees": "0.03"
      }
    },
    {
      "tx_ref_code": "375X4",
      "channel_id": "af9c1f5ae2c51e105e05e75ab716a2067",
      "state": "pending",
      "created_at": 1427138127.0,
      "updated_at": 1427138127.0,
      "sender": {
        "currency": "BTC",
        "amount": "0.00010000",
        "bitcoin_txid": "d25100ff30af7ad87f5da8a509c4b6a51df22a27b02bcb4a5567b66d7913d47f"
      },
      "receiver": {
        "currency": "EUR",
        "amount": "0.02",
        "amount_plus_fees": "0.03"
      }
    },
    {
      "tx_ref_code": "B5CU5",
      "channel_id": "af9c1f5ae2c51e105e05e75ab716a2067",
      "state": "pending",
      "created_at": 1427138200.0,
      "updated_at": 1427138200.0,
      "sender": {
```

```

    "currency": "BTC",
    "amount": "0.00005430",
    "bitcoin_txid": "a4521e87cb2e610f79e2a4e55bdb07d416a5844d0b58ad02be57bf2fdee06778"
  },
  "receiver": {
    "currency": "BTC",
    "amount": "0.00005430",
    "amount_plus_fees": "00005431"
  }
}
]
}

```

## GET /api/v1/channels/{channel\_id}/txs/{tx\_ref\_code}

Get information about an individual transactions of a channel.

### Request

None

### Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
tx_ref_code	string(6)	Unique character string to reference this transaction
channel_id	string(32)	Id of the channel this transaction belongs to
state	string(9)	<code>pending</code> , <code>completed</code> or <code>cancelled</code>
created_at	number	(float) Unix-epoch timestamp of the transaction creation
updated_at	number	(float) Unix-epoch timestamp when the transaction data was last updated
sender	object	Information about the <i>sender</i> part of this transaction
receiver	object	Information about the <i>receiver</i> part of this transaction

Unconfirmed transactions will stay in the `pending` state until Cubits deems the risk to accept them as low enough for them to go in the `completed` state. A transaction that was a double-spend or one that will not confirm for other reasons (e.g. because of the Bitcoin dust limit), will eventually go to `cancelled` state.

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was sent
amount	string(17)	Amount that was sent
bitcoin_txid	string(64)	Bitcoin transaction id of this transaction or <code>null</code> in case of a Cubits internal transfer

## Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was received. Note that this will be <code>BTC</code> even for fiat channels in case less than the equivalent of the smallest fiat currency unit of the fiat currency was sent.
amount	string(17)	Amount that was received. Note that this amount is preliminary in case of <code>pending</code> transactions and might change once the transaction goes to <code>completed</code> state.

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
GET /api/v1/channels/af9c1f5ae2c51e105e05e75ab716a2067/txs/MH5AD HTTP/1.1
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:



```

HTTP/1.1 200 OK
Content-Type: application/json

{
  "tx_ref_code": "MH5AD",
  "channel_id": "af9c1f5ae2c51e105e05e75ab716a2067",
  "state": "completed",
  "created_at": 1427137517.0,
  "updated_at": 1427137623.0,
  "sender": {
    "currency": "BTC",
    "amount": "0.00010000",
    "bitcoin_txid": "9ae311aae58c9150ca22f5b3fa7b64e32bc4e25cf9db11ba6bd28de50bcd1f73"
  },
  "receiver": {
    "currency": "EUR",
    "amount": "0.02"
  }
}

```

## POST /api/v1/channels

Creates a new channel.

### Request

Attribute	Data type	Description
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
name	string(256)	<i>(optional)</i> Name of the channel, displayed to the customer on the payment screen
description	string(512)	<i>(optional)</i> Description of the item displayed to the customer on the payment screen
reference	string(512)	<i>(optional)</i> Individual free-text field stored in the channel as-is
callback_url	string(512)	<i>(optional)</i> URL that is called on channel status updates
txs_callback_url	string(512)	<i>(optional)</i> URL that is called on channel transaction updates
success_url	string(512)	<i>(optional)</i> URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")

### Response

On success, a response with HTTP status `201 Created` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the channel
address	string(34)	Bitcoin address associated with this channel
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
name	string(256)	Name of the channel, displayed to the customer on the payment screen
description	string(512)	Description of the item displayed to the customer on the payment screen
reference	string(512)	Individual free-text field stored in the channel as-is
channel_url	string(512)	Unique URL of the channel payment screen on Cubits
callback_url	string(512)	URL that is called on channel status updates
txs_callback_url	string(512)	URL that is called on channel transaction updates
success_url	string(512)	URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")
created_at	number	(float) Unix-epoch timestamp of the channel creation
updated_at	number	(float) Unix-epoch timestamp when the channel data was last updated

## Errors

On error, the API responds with standard [error responses](#) and with one specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Requested currency is not supported	400 Bad Request	Unsupported currency

## Example

Request:

```
POST /api/v1/channels HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "receiver_currency": "EUR",
  "name": "Donation payment channel",
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "success_url": "https://example.com/thank_you.html",
  "share_to_keep_in_btc": "0"
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "receiver_currency": "EUR",
  "name": "Donation payment channel",
  "description": null,
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "txs_callback_url": null,
  "success_url": "https://example.com/thank_you.html",
  "id": "f5790652fb26b62bb48e65941bec06f8",
  "channel_url": "https://pay.cubits.com/channels/f5790652fb26b62bb48e65941bec06f8",
  "address": "3AvsD1FSJwUwrnXweHVJJ2Av6P4JP1sVyxa",
  "share_to_keep_in_btc": "0",
  "created_at": 1427217218.0,
  "updated_at": 1427217218.0
}
```

## POST /api/v1/channels/{channel\_id}

---

Updates an existing channel.

### Request

Attribute	Data type	Description
receiver_currency	string(3)	<i>(optional)</i> <a href="#">ISO 4217</a> code of the new currency that you want to receive (e.g. "EUR")
name	string(256)	<i>(optional)</i> New name of the channel, displayed to the customer on the payment screen
description	string(512)	<i>(optional)</i> New description of the item displayed to the customer on the payment screen
reference	string(512)	<i>(optional)</i> New free-text field stored in the channel as-is
callback_url	string(512)	<i>(optional)</i> New URL that is called on channel status updates
txs_callback_url	string(512)	<i>(optional)</i> URL that is called on channel transaction updates
success_url	string(512)	<i>(optional)</i> New URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")

## Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the channel
address	string(34)	Bitcoin address associated with this channel
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
name	string(256)	Name of the channel, displayed to the customer on the payment screen
description	string(512)	Description of the item displayed to the customer on the payment screen
reference	string(512)	Individual free-text field stored in the channel as-is
channel_url	string(512)	Unique URL of the channel payment screen on Cubits
callback_url	string(512)	URL that is called on channel status updates
txs_callback_url	string(512)	URL that is called on channel transaction updates
success_url	string(512)	URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")
created_at	number	(float) Unix-epoch timestamp of the channel creation
updated_at	number	(float) Unix-epoch timestamp when the channel data was last updated

## Errors

On error, the API responds with standard [error responses](#) and with one specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Requested currency is not supported	400 Bad Request	Unsupported currency

## Example

Request:

```
POST /api/v1/channels/f5790652fb26b62bb48e65941bec06f8 HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "receiver_currency": "BTC",
  "share_to_keep_in_btc": "10"
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "receiver_currency": "BTC",
  "name": "Donation payment channel",
  "description": null,
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "txs_callback_url": null,
  "success_url": "https://example.com/thank_you.html",
  "id": "f5790652fb26b62bb48e65941bec06f8",
  "channel_url": "https://pay.cubits.com/channels/f5790652fb26b62bb48e65941bec06f8",
  "address": "3AvsD1FSJwUwrnXweHVJJ2Av6P4JP1sVyxa",
  "share_to_keep_in_btc": "10",
  "created_at": 1427217218.0,
  "updated_at": 1427217219.0
}
```

## Callbacks

When specifying a `callback_url`, the server will send callbacks to inform you about changes to any of the channel fields. The POSTed request body will correspond to the result of a GET request to `/api/v1/channels/{channel_id}`.

When specifying a `txs_callback_url`, the server will send callbacks to inform you about changes to any of the channel's transactions. The POSTed request body will correspond to the result of a GET request to

/api/v1/channels/{channel\_id}/txs/{tx\_ref\_code}.

See [Callbacks](#) for a general description of the Cubits callback mechanism and format.

Usually we will send a callback within a few seconds of a Bitcoin transaction being propagated on the Bitcoin network even though it is not yet confirmed and you will see those transactions as `pending` .

# Quote Channels

---

Quote channels provide a method to receive and potentially convert Bitcoin payments of variable amounts just as with regular channels. The difference is that quote channels have a fixed exchange rate that is used for the Bitcoin conversion within the validity time of the quote but at the same time still convert incoming funds to the requested *receiver\_currency* at the spot price if they are paid after the validity time. Quote channels have to be created for a certain expected amount in any of the supported fiat currencies. They have a fixed payment address and support callbacks. Bitcoin payments for less than the smallest possible unit for the chosen fiat currency will not be converted but transferred as-is to the Cubits Wallet. Quote channels also provide a way of tracking individual payments made to them. A list of transactions is recorded with each quote channel and returned upon querying its transaction endpoint as well as with each transaction callback.

## GET /api/v1/quote\_channels/{channel\_id}

---

Get information about an existing quote channel.

### Request

None

### Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the quote channel
address	string(34)	Bitcoin address associated with this quote channel
sender_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that the quote channel can accept (currently only "BTC" is supported)
sender_amount	string(16)	Amount of currency ("BTC") the quote channel expects
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
receiver_amount	string(16)	Amount of currency that should be received (used to calculate the quote) as a decimal floating point number, converted to string (e.g. "123.05")
name	string(256)	Name of the quote channel
description	string(512)	Description of the quote channel
reference	string(512)	Individual free-text field stored in the quote channel as-is
callback_url	string(512)	URL that is called on quote channel updates
txs_callback_url	string(512)	URL that is called on quote channel transaction updates
success_url	string(512)	URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")
created_at	number	(float) Unix-epoch timestamp of the quote channel creation
updated_at	number	(float) Unix-epoch timestamp when the quote channel data was last updated
valid_until	number	(float) Unix-epoch timestamp after which the quote will expire

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
GET /api/v1/quote_channels/ae9641776a94456a8067970b9765e396 HTTP/1.1
Accept: application/vnd.api+json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:



```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```

```
{
  "sender_currency": "BTC",
  "sender_amount": "0.40623221",
  "receiver_currency": "EUR",
  "receiver_amount": "164.25",
  "name": "Order XYZ",
  "description": null,
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "txs_callback_url": "https://example.com/txs_callback",
  "success_url": "https://example.com/thank_you.html",
  "created_at": 1427234017.0,
  "updated_at": 1427235200.0,
  "id": "ae9641776a94456a8067970b9765e396",
  "address": "35xv5napaKi4sLuYZWT9xcp7aaj6YZxpKc",
  "share_to_keep_in_btc": "0",
  "valid_until": 1427234917.0
}
```

## GET /api/v1/quote\_channels/{channel\_id}/txs

Get information about all transactions of an existing quote channel.

### Request

Attribute	Data type	Description
page	number	<i>(optional)</i> Page number to return (default: 1)
per_page	number	<i>(optional)</i> Number of transactions to return per page (max: 1000, default: 100)

### Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
pagination	object	<i>Pagination</i> object for the returned txs array
txs	array	Array of <i>transaction</i> objects of this quote channel sorted by descending <i>updated_at</i> field

### Pagination Object

Attribute	Data type	Description
page	number	Index of the current page (1 <= page <= page_count)
page_count	number	Number of total pages in the result
per_page	number	Number of entries per page (1 <= per_page <= 1000)
total_count	number	Number of total entries in the result

## Transaction Object

Attribute	Data type	Description
tx_ref_code	string(6)	Unique character string to reference this transaction
quote_channel_id	string(32)	Id of the quote channel this transaction belongs to
state	string(9)	pending , completed or cancelled
created_at	number	(float) Unix-epoch timestamp of the transaction creation
updated_at	number	(float) Unix-epoch timestamp when the transaction data was last updated
sender	object	Information about the <i>sender</i> part of this transaction
receiver	object	Information about the <i>receiver</i> part of this transaction

Unconfirmed transactions will stay in the `pending` state until Cubits deems the risk to accept them as low enough for them to go in the `completed` state. A transaction that was a double-spend or one that will not confirm for other reasons (e.g. because of the Bitcoin dust limit), will eventually go to `cancelled` state.

## Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was sent
amount	string(17)	Amount that was sent
bitcoin_txid	string(64)	Bitcoin transaction id of this transaction or <code>null</code> in case of a Cubits internal transfer

## Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was received.
amount	string(17)	Amount that was received, with fees deducted. Note that this amount is preliminary in case of <code>pending</code> transactions and might change once the transaction goes to <code>completed</code> state.
amount_plus_fees	string(17)	Amount that was received, without fees deducted. Note that this amount is preliminary in case of <code>pending</code> transactions and might change once the transaction goes to <code>completed</code> state.

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
GET /api/v1/quote_channels/ae9641776a94456a8067970b9765e396/txs HTTP/1.1
Accept: application/vnd.api+json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:

HTTP/1.1 200 OK  
Content-Type: application/vnd.api+json

```
{
  "pagination": {
    "page": 1,
    "page_count": 1,
    "per_page": 100,
    "total_count": 3
  },
  "txs": [
    {
      "tx_ref_code": "G3H4W",
      "quote_channel_id": "ae9641776a94456a8067970b9765e396",
      "state": "completed",
      "created_at": 1427234023.0,
      "updated_at": 1427234023.0,
      "sender": {
        "currency": "BTC",
        "amount": "0.00020000",
        "bitcoin_txid": "dd0b3099a7bb981d6ac08ec80d805d74260c8d83649a9466671018144408765c"
      },
      "receiver": {
        "currency": "EUR",
        "amount": "0.04",
        "amount_plus_fees": "0.05"
      }
    },
    {
      "tx_ref_code": "284W9",
      "quote_channel_id": "ae9641776a94456a8067970b9765e396",
      "state": "pending",
      "created_at": 1427234123.0,
      "updated_at": 1427235200.0,
      "sender": {
        "currency": "BTC",
        "amount": "0.00050000",
        "bitcoin_txid": "67d2de1c07129c482c88db51563c252bc52267976d4b060f8f78ceff1bb419b3"
      },
      "receiver": {
        "currency": "EUR",
        "amount": "0.12",
        "amount_plus_fees": "0.13"
      }
    }
  ]
}
```

## GET

**/api/v1/quote\_channels/{channel\_id}/txs/{tx\_ref\_code}**

Get information about an individual transactions of a quote channel.

## Request

None

## Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
tx_ref_code	string(6)	Unique character string to reference this transaction
quote_channel_id	string(32)	Id of the quote channel this transaction belongs to
state	string(9)	<code>pending</code> , <code>completed</code> or <code>cancelled</code>
created_at	number	(float) Unix-epoch timestamp of the transaction creation
updated_at	number	(float) Unix-epoch timestamp when the transaction data was last updated
sender	object	Information about the <i>sender</i> part of this transaction
receiver	object	Information about the <i>receiver</i> part of this transaction

Unconfirmed transactions will stay in the `pending` state until Cubits deems the risk to accept them as low enough for them to go in the `completed` state. A transaction that was a double-spend or one that will not confirm for other reasons (e.g. because of the Bitcoin dust limit), will eventually go to `cancelled` state.

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was sent
amount	string(17)	Amount that was sent
bitcoin_txid	string(64)	Bitcoin transaction id of this transaction or <code>null</code> in case of a Cubits internal transfer

### Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was received.
amount	string(17)	Amount that was received, with fees deducted. Note that this amount is preliminary in case of <code>pending</code> transactions and might change once the transaction goes to <code>completed</code> state.
amount_plus_fees	string(17)	Amount that was received, without fees deducted. Note that this amount is preliminary in case of <code>pending</code> transactions and might change once the transaction goes to <code>completed</code> state.

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```
GET /api/v1/quote_channels/ae9641776a94456a8067970b9765e396/txs/XGHW HTTP/1.1
Accept: application/vnd.api+json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json

{
  "tx_ref_code": "XGHW",
  "quote_channel_id": "ae9641776a94456a8067970b9765e396",
  "state": "completed",
  "created_at": 1427137517.0,
  "updated_at": 1427137623.0,
  "valid_until": 1427138417.0,
  "sender": {
    "currency": "BTC",
    "amount": "0.00010000",
    "bitcoin_txid": "9ae311aae58c9150ca22f5b3fa7b64e32bc4e25cf9db11ba6bd28de50bcd1f73"
  },
  "receiver": {
    "currency": "EUR",
    "amount": "0.02",
    "amount_plus_fees": "0.03"
  }
}
```

---

## POST /api/v1/quote\_channels

Creates a new quote channel.

### Request

Attribute	Data type	Description
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
receiver_amount	string(16)	Amount of currency that should be received (used to calculate the quote) as a decimal floating point number, converted to string (e.g. "123.05")
name	string(256)	<i>(optional)</i> Name of the quote channel
description	string(512)	<i>(optional)</i> Description of the quote channel
reference	string(512)	<i>(optional)</i> Individual free-text field stored in the quote channel as-is
callback_url	string(512)	<i>(optional)</i> URL that is called on quote channel status updates
txs_callback_url	string(512)	<i>(optional)</i> URL that is called on quote channel transaction updates
success_url	string(512)	<i>(optional)</i> URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")

## Response

On success, a response with HTTP status `201 Created` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the quote channel
address	string(34)	Bitcoin address associated with this quote channel
sender_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that the quote channel can accept (currently only "BTC" is supported)
sender_amount	string(16)	Amount of currency ("BTC") the quote channel expects
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
receiver_amount	string(16)	Amount of currency that should be received (used to calculate the quote) as a decimal floating point number, converted to string (e.g. "123.05")
name	string(256)	Name of the quote channel
description	string(512)	Description of the quote channel
reference	string(512)	Individual free-text field stored in the quote channel as-is
callback_url	string(512)	URL that is called on quote channel status updates
txs_callback_url	string(512)	URL that is called on quote channel transaction updates
success_url	string(512)	URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")
created_at	number	(float) Unix-epoch timestamp of the quote channel creation
updated_at	number	(float) Unix-epoch timestamp when the quote channel data was last updated
valid_until	number	(float) Unix-epoch timestamp after which the quote will expire

## Errors

On error, the API responds with standard [error responses](#) and with one specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Requested currency is not supported	400 Bad Request	Unsupported currency

## Example

Request:



```
POST /api/v1/quote_channels HTTP/1.1
Content-Type: application/vnd.api+json
Accept: application/vnd.api+json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "receiver_currency": "EUR",
  "receiver_amount": "164.25",
  "name": "name": "Order XYZ",
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "success_url": "https://example.com/thank_you.html",
  "share_to_keep_in_btc": "0"
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/vnd.api+json

{
  "sender_currency": "BTC",
  "sender_amount": "0.40623221",
  "receiver_currency": "EUR",
  "receiver_amount": "164.25",
  "name": "Order XYZ",
  "description": null,
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "txs_callback_url": null,
  "success_url": "https://example.com/thank_you.html",
  "id": "f5790652fb26b62bb48e65941bec06f8",
  "address": "3AvsD1FSJwUwrnXweHVJJ2Av6P4JP1sVyxa",
  "share_to_keep_in_btc": "0",
  "created_at": 1427342218.0,
  "updated_at": 1427342218.0,
  "valid_until": 1427343118.0
  "share_to_keep_in_btc": "0"
}
```

## POST /api/v1/quote\_channels/{channel\_id}

---

Updates an existing quote channel.

### Request

Attribute	Data type	Description
description	string(512)	<i>(optional)</i> New description of the quote channel
reference	string(512)	<i>(optional)</i> New free-text field stored in the quote channel as-is
name	string(256)	<i>(optional)</i> New name of the quote channel
callback_url	string(512)	<i>(optional)</i> New URL that is called on quote channel status updates
txs_callback_url	string(512)	<i>(optional)</i> URL that is called on quote channel transaction updates
success_url	string(512)	<i>(optional)</i> New URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")

## Response

On success, a response with HTTP status `200 OK` is returned containing the following attributes:

Attribute	Data type	Description
id	string(32)	Unique hex-string identifier of the quote channel
address	string(34)	Bitcoin address associated with this quote channel
sender_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that the quote channel can accept (currently only "BTC" is supported)
sender_amount	string(16)	Amount of currency ("BTC") the quote channel expects
receiver_currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
receiver_amount	string(16)	Amount of currency that should be received (used to calculate the quote)
name	string(256)	Name of the quote channel
description	string(512)	Description of the quote channel
reference	string(512)	Individual free-text field stored in the quote channel as-is
callback_url	string(512)	URL that is called on quote channel status updates
txs_callback_url	string(512)	URL that is called on quote channel transaction updates
success_url	string(512)	URL to redirect the user to after a successful payment
share_to_keep_in_btc	string(16)	Per cent of the each transaction to be kept in BTC, as a decimal number, converted to string (e.g. "20")
created_at	number	(float) Unix-epoch timestamp of the quote channel creation
updated_at	number	(float) Unix-epoch timestamp when the quote channel data was last updated
valid_until	number	(float) Unix-epoch timestamp after which the quote will expire

## Errors

On error, the API responds with standard [error responses](#).

## Example

Request:

```

POST /api/v1/quote_channels/f5790652fb26b62bb48e65941bec06f8 HTTP/1.1
Content-Type: application/vnd.api+json
Accept: application/vnd.api+json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "success_url": "https://example.com/thank_you2.html",
  "share_to_keep_in_btc": "10"
}

```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/vnd.api+json
```

```
{
  "sender_currency": "BTC",
  "sender_amount": "0.40623221",
  "receiver_currency": "EUR",
  "receiver_amount": "164.25",
  "name": "Order XYZ",
  "description": null,
  "reference": "xyz",
  "callback_url": "https://example.com/callback",
  "txs_callback_url": null,
  "success_url": "https://example.com/thank_you2.html",
  "id": "f5790652fb26b62bb48e65941bec06f8",
  "address": "3AvsD1FSJwUwrnXweHVJJ2Av6P4JP1sVyxa",
  "share_to_keep_in_btc": "10",
  "created_at": 1427217218.0,
  "updated_at": 1427217219.0,
  "valid_until": 1427218118.0
}
```

## Callbacks

---

When specifying a `callback_url`, the server will send callbacks to inform you about changes to any of the quote channel fields. The POSTed request body will correspond to the result of a GET request to `/api/v1/quote_channels/{channel_id}`.

When specifying a `txs_callback_url`, the server will send callbacks to inform you about changes to any of the quote channel's transactions. The POSTed request body will correspond to the result of a GET request to `/api/v1/quote_channels/{channel_id}/txs/{tx_ref_code}`.

See [Callbacks](#) for a general description of the Cubits callback mechanism and format.

Usually we will send a callback within a few seconds of a Bitcoin transaction being propagated on the Bitcoin network even though it is not yet confirmed and you will see those transactions as `pending`.

# Accounts

## GET /api/v1/accounts

Retrieves a list of your Cubits wallet accounts. Each wallet can have accounts in different currencies. With this call you can get a complete overview of all your balances on Cubits.

### Request

None.

### Response

On success, the API responds with HTTP status `200 OK` and the following attributes:

Attribute	Data type	Description
accounts	array	Array of <i>account</i> objects

### Account Objects

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the account currency (e.g. "EUR")
balance	string(32)	Current balance of the account, decimal number as a string (e.g. "12.50")

### Errors

On error, the API responds with standard [error responses](#).

### Example

Request:

```
GET /api/v1/accounts HTTP/1.1
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

Response:

HTTP/1.1 200 OK  
Content-Type: application/json

```
{
  "accounts": [
    {
      "currency": "EUR",
      "balance": "128.45"
    },
    {
      "currency": "BTC",
      "balance": "0.35090347"
    }
  ]
}
```

# Send Money

## POST /api/v1/send\_money

Creates a transaction to send bitcoins from your Cubits wallet to an external bitcoin address.

### Request

Attribute	Data type	Description
amount	string(32)	Amount in BTC to be sent, decimal number as a string (e.g. "0.12500000")
address	string(64)	Bitcoin address the amount is to be sent to
reference	string(512)	<i>(optional)</i> Individual free-text field stored in the tx as-is

### Response

On success, the API responds with HTTP status `201 Created` and the following attributes:

Attribute	Data type	Description
tx_ref_code	string(32)	Reference code of the created transaction

### Errors

On error, the API responds with standard [error responses](#) and with some specific to this request:

#### Client errors (4xx)

Situation	HTTP status code	message
Amount has invalid format	400 Bad Request	Invalid amount
Requested amount exceeds balance	400 Bad Request	Insufficient funds
Bitcoin address is invalid	400 Bad Request	Invalid address

### Example

Request:

```
POST /api/v1/send_money HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "amount": "0.12340000",
  "address": "1AeMbkpHia8FVuKczQKUrV9uMzv7uC1HZi"
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
  "tx_ref_code": "6NFXB"
}
```



# Quotes

Quotes provide an estimation of the exchange rate for a given amount and operation type. They can be used to get information about the result of a subsequent *buy* or *sell* operation.

## POST /api/v1/quotes

Requests a quote for a *buy* or *sell* operation.

### Request

Attribute	Data type	Description
operation	string(32)	Type of the transaction: <code>buy</code> or <code>sell</code>
sender	object	<i>Sender</i> object specifying the spending part of the transaction
receiver	object	<i>Receiver</i> object specifying the receiving part of the transaction

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to spend (e.g. "EUR")
amount	string(32)	<i>(optional)</i> Amount in specified currency to be spent, decimal number as a string (e.g. "12.50")

### Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")
amount	string(32)	<i>(optional)</i> Amount in specified currency to be received, decimal number as a string (e.g. "12.50")

### Required Attributes

Exactly one amount, either `sender.amount` or `receiver.amount` must be specified.

### Response

On success, the API responds with HTTP status `201 Created` and the following attributes:

Attribute	Data type	Description
operation	string(32)	Type of the transaction: buy or sell
sender	object	<i>Sender</i> object specifying the spending part of the transaction
receiver	object	<i>Receiver</i> object specifying the receiving part of the transaction

## Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency to spend (e.g. "EUR")
amount	string(32)	Amount in specified currency to be spent, decimal number as a string (e.g. "12.50")

## Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency to receive (e.g. "EUR")
amount	string(32)	Amount in specified currency to be received, decimal number as a string (e.g. "12.50")

## Errors

On error, the API responds with standard [error responses](#) and with some specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Requested operation is not supported	400 Bad Request	Invalid operation
Amount has invalid format	400 Bad Request	Invalid amount
Requested currency is not supported	400 Bad Request	Unsupported currency

## Example

Request:

```
POST /api/v1/quotes HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

```
{
  "operation": "buy",
  "sender": {
    "currency": "EUR"
  },
  "receiver": {
    "amount": "0.15000000",
    "currency": "BTC"
  }
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```
{
  "operation": "buy",
  "sender": {
    "amount": "29.89",
    "currency": "EUR"
  },
  "receiver": {
    "amount": "0.15000000",
    "currency": "BTC"
  }
}
```

# Buy

## POST /api/v1/buy

Creates a transaction to buy bitcoins using funds from your Cubits account. Bought bitcoins will be credited to your Cubits wallet.

The exact exchange rate will be calculated at the transaction execution time.

### Request

Attribute	Data type	Description
sender	object	<i>Sender</i> object defining the spending part of the transaction

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to spend (e.g. "EUR")
amount	string(17)	Amount in specified currency to be spent, decimal number as a string (e.g. "12.50")
amount_minus_fees	string(17)	Amount in specified currency to be sent after all fees paid, decimal number as a string (e.g. "12.50")

### Response

On success, the API responds with HTTP status `201 Created` and the following attributes:

Attribute	Data type	Description
tx_ref_code	string(32)	Reference code of the created transaction
sender	object	Information about the <i>sender</i> part of this transaction
receiver	object	Information about the <i>receiver</i> part of this transaction

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was spent
amount	string(17)	Amount that was spent

### Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was received
amount	string(17)	Amount that was received

## Errors

On error, the API responds with standard [error responses](#) and with some specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Amount has invalid format	400 Bad Request	Invalid amount
Requested amount exceeds balance	400 Bad Request	Insufficient funds
Requested currency is not supported	400 Bad Request	Unsupported currency

## Example

Request to buy using amount parameter:

Request:

```

POST /api/v1/buy HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "sender": {
    "currency": "EUR",
    "amount": "12.50"
  }
}

```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```
{
  "tx_ref_code": "6NFXB",
  "sender": {
    "currency": "EUR",
    "amount": "12.50",
    "amount_minus_fees": "12.49"
  },
  "receiver": {
    "currency": "BTC",
    "amount": "0.05750000"
  }
}
```

Request to buy using `amount_minus_fees` parameter:

Request:

```
POST /api/v1/buy HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****
```

```
{
  "sender": {
    "currency": "EUR",
    "amount_minus_fees": "12.45"
  }
}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
```

```
{
  "tx_ref_code": "6NFXB",
  "sender": {
    "currency": "EUR",
    "amount": "12.50"
  },
  "receiver": {
    "currency": "BTC",
    "amount": "0.05750000"
  }
}
```

# Buysend

## POST /api/v1/buysend

Creates a transaction to buy bitcoins using funds from your Cubits account. Bought bitcoins will be sent to address provided.

The exact exchange rate will be calculated at the transaction execution time.

### Request

Attribute	Data type	Description
sender	object	<i>Sender</i> object defining the spending part of the transaction
address	string(64)	Bitcoin address the amount is to be sent to
reference	string(512)	<i>(optional)</i> Individual free-text field stored in the transaction as-is

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to spend (e.g. "EUR")
amount	string(17)	Amount in specified currency to be spent, decimal number as a string (e.g. "12.50")
amount_minus_fees	string(17)	Amount after fees in specified currency to be spent, decimal number as a string (e.g. "12.50")

Note: only one `amount` parameter should be passed. Error will be returned otherwise.

### Response

On success, the API responds with HTTP status `201 Created` and the following attributes:

Attribute	Data type	Description
tx_ref_code	string(32)	Reference code of the created transaction
sender	object	Information about the <i>sender</i> part of this transaction
receiver	object	Information about the <i>receiver</i> part of this transaction

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was spent
amount	string(17)	Amount that was spent
amount_minus_fees	string(17)	Total amount that was send in chosen currency

## Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was received
amount	string(17)	Amount that was received
address	string(64)	Address where money were transferred

## Errors

On error, the API responds with standard [error responses](#) and with some specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Amount has invalid format	400 Bad Request	Invalid amount
Requested amount exceeds balance	400 Bad Request	Insufficient funds
Requested currency is not supported	400 Bad Request	Unsupported currency
Bitcoin address is invalid	400 Bad Request	Invalid address

## Example

Request:

```
POST /api/v1/buysend HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "sender": {
    "currency": "EUR",
    "amount": "12.50"
  },
  "address": "1AeMbkpHia8FVuKczQKUrV9uMzv7uC1HZi"
}
```

Response:



HTTP/1.1 201 Created  
Content-Type: application/json

```
{
  "tx_ref_code": "6NFXB",
  "sender": {
    "currency": "EUR",
    "amount": "12.50",
    "amount_minus_fees": "12.40",
  },
  "receiver": {
    "currency": "BTC",
    "amount": "0.05750000",
    "address": "1AeMbkpHia8FVuKczQKUrV9uMzv7uC1HZi"
  }
}
```

# Sell

## POST /api/v1/sell

Creates a transaction to sell bitcoins from your Cubits wallet and receive amount in specified fiat currency. Fiat funds will be credited to your Cubits account.

The exact exchange rate will be calculated at the transaction execution time.

### Request

Attribute	Data type	Description
sender	object	<i>Sender</i> object defining the spending part of the transaction
receiver	object	<i>Receiver</i> object defining the receiving part of the transaction

### Sender Object

Attribute	Data type	Description
amount	string(17)	Amount in BTC to be spent, decimal number as a string (e.g. "0.01250000")

### Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that you want to receive (e.g. "EUR")

### Response

On success, the API responds with HTTP status `201 Created` and the following attributes:

Attribute	Data type	Description
tx_ref_code	string(32)	Reference code of the created transaction
sender	object	Information about the <i>sender</i> part of this transaction
receiver	object	Information about the <i>receiver</i> part of this transaction

### Sender Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was spent
amount	string(17)	Amount that was spent

## Receiver Object

Attribute	Data type	Description
currency	string(3)	<a href="#">ISO 4217</a> code of the currency that was received
amount	string(17)	Amount that was received

## Errors

On error, the API responds with standard [error responses](#) and with some specific to this request:

### Client errors (4xx)

Situation	HTTP status code	message
Amount has invalid format	400 Bad Request	Invalid amount
Requested amount exceeds balance	400 Bad Request	Insufficient funds
Requested currency is not supported	400 Bad Request	Unsupported currency

## Example

Request:

```
POST /api/v1/sell HTTP/1.1
Content-Type: application/json
Accept: application/json
X-Cubits-Key: *****
X-Cubits-Nonce: *****
X-Cubits-Signature: *****

{
  "sender": {
    "amount": "0.15000000"
  },
  "receiver": {
    "currency": "EUR"
  }
}
```

Response:

HTTP/1.1 201 Created  
Content-Type: application/json

```
{
  "tx_ref_code": "6NFXB",
  "sender": {
    "currency": "BTC",
    "amount": "0.15000000"
  },
  "receiver": {
    "currency": "EUR",
    "amount": "32.63"
  }
}
```

# Appendices

---

## List of supported currencies

---

This table lists the ISO 4217 codes of currencies supported by Cubits

<b>ISO 4217</b>	<b>Currency</b>
ARS	Argentine Peso
AUD	Australian Dollar
BRL	Brazilian Real
CAD	Canadian Dollar
CHF	Swiss Franc
CNY	Chinese Yuan Renminbi
CZK	Czech Republic Koruna
DKK	Danish Krone
EUR	Euro
GBP	British Pound Sterling
HRK	Croatian Kuna
HUF	Hungarian Forint
INR	Indian Rupee
JPY	Japanese Yen
KRW	South Korean Won
MXN	Mexican Peso
MYR	Malaysian Ringgit
NOK	Norwegian Krone
PLN	Polish Zloty
RSD	Serbian Dinar
RUB	Russian Ruble
SEK	Swedish Krona
SGD	Singapore Dollar
THB	Thai Baht
TRY	Turkish Lira
USD	US Dollar
VND	Vietnamese Dong
ZAR	South African Rand