ChainCloud Documentation

Release 0.0.1

WH

Apr 25, 2018

Contents

ChainCloud - Cloud Your Chain!

Contents:

CHAPTER 1

Overview

1.1 What is ChainCloud?

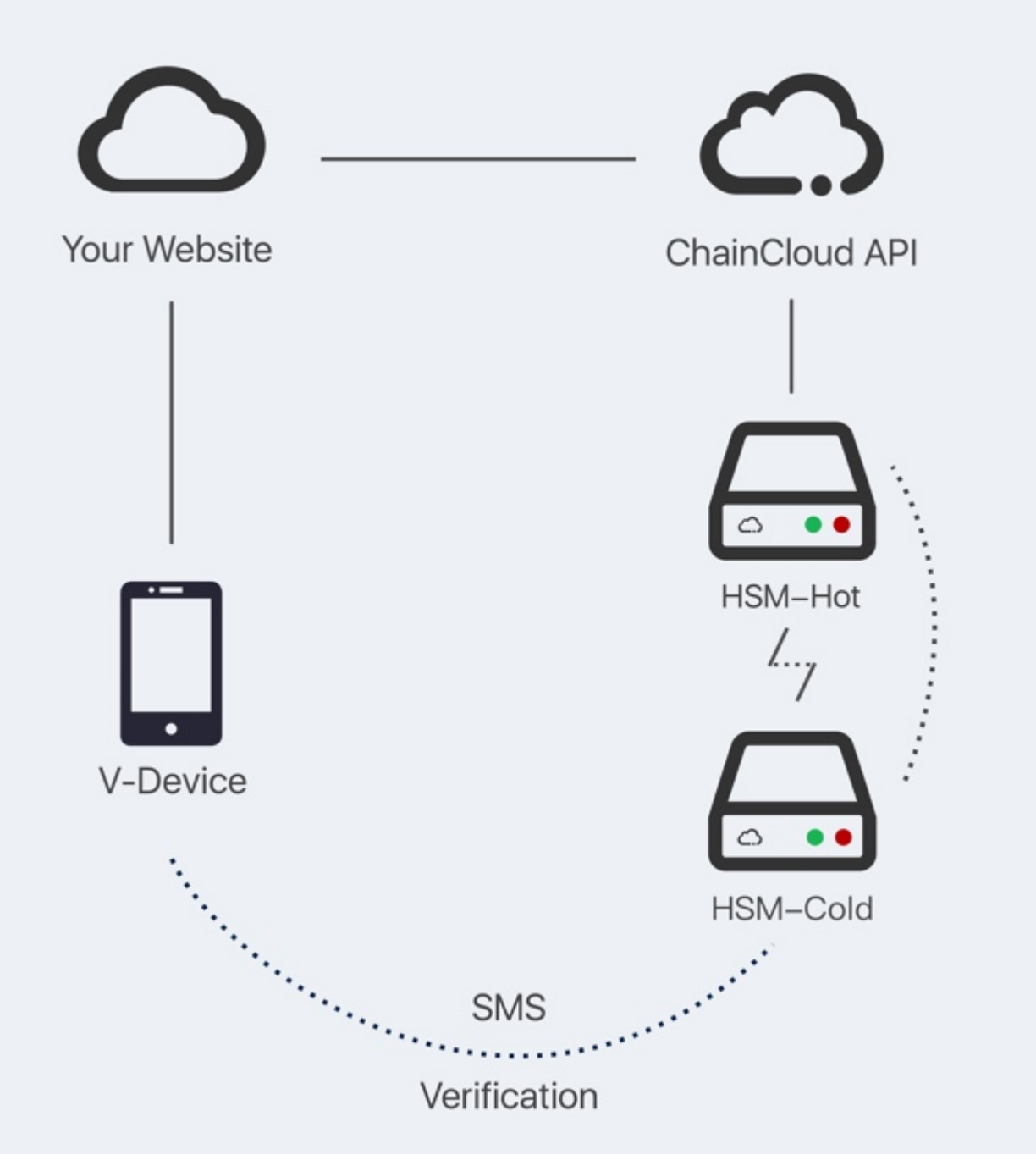
ChainCloud is a platform that can help you to connect your website or application with the most popular blockchain assets (Bitcoin, Ethereum, Litecoin, Dogecoin and so on).

1.1.1 Hello

Hello

1.2 Architecture

1.2.1 Diagram



1.2.2 HSM

HSM means Hardware Security Module. We have specially designed two HSM modules to construct unbreakable secured blockchain cloud platform:

- HSM-Hot
- HSM-Cold

1.2.3 V-Device

V-Device is like a "2-Step Verification" Device for you. It is an Android device running with ChainCloud-V (an Open Source App developped by us).

CHAPTER 2

API

2.1 General Usage

How to use ChainCloud API?

2.1.1 Access Method

Domain: **chaincloud-api.getcai.com** (api.chaincloud.com in the future)

All API should be accessed with `token` (in HTTP header), and you can apply for your API token here.

2.1.2 Access Samples

Python:

```
domain = 'chaincloud-api.getcai.com'
url = 'https://' + domain + '/api/v1/open/time'
token = '3333bd0af3a28d0c18b32206627ac1af68dc63e304b9975f08917c53831a6666'
request = urllib2.Request(url)
request.add_header('token', token)
response = urllib2.urlopen(request, timeout=global_timeout)
```

Curl:

```
curl -H "token:3333bd0af3a28d0c18b32206627ac1af68dc63e304b9975f08917c53831a6666"␣
↪https://chaincloud-api.getcai.com/api/v1/open/time
```

2.1.3 API call rate limit

We will set limits in the future.

2.2 Basic API

All basic APIs.

2.2.1 Time

GET /api/v1/open/time

Returns Server Time.

Example response:

```
{
    "time": "2016-08-02T12:51:34.460394"
}
```

2.2.2 User

GET /api/v1/open/user

Returns Current User's Information.

Example response:

```
{
    "can_otc_ad": 0,
    "user_id": 40000,
    "gender": 0,
    "user_type": 4,
    "real_name": null,
    "receiving_index": 1,
    "avatar": null,
    "register_at": "2016-07-11T14:19:55",
    "address": "1Mzvo2UkBnEpQC8xkui23mYQQeL8NREqrJ",
    "balance": 60000,
    "user_name": "bitpie-hot-40000",
    "can_otc_order": 0
}
```

2.2.3 Address Batch

GET /api/v1/open/address/batch/ *(int: batch_no)*

Returns a batch of receiving addresses. You should use this API to retrieve addresses (both Cold Receiving and Hot Sending) each time for a batch, and after that you should use V Device to verify the data of this batch.

Example response:

```
[
    {
        "index": 0,
        "address": "1B4qK6KTzWCnbNcx4WDYR99KfVRsU2zDcN"
    },
    {
```

```
        "index": 1,
        "address": "1Mzvo2UkBnEpQC8xkui23mYQQeL8NREqrJ"
    },
    {
        "index": 2,
        "address": "1n566cWbw6c3KLBJ8X1vvo8vi4G8BMmEV"
    },
    {"...": "..."},
    {
        "index": 997,
        "address": "1Hc6MffGiYPLJdwAnR42JpTYdGonfWEJ7N"
    },
    {
        "index": 998,
        "address": "1KqPMBXtmvKkLe9TCHzVR4New3c5xkVgge"
    },
    {
        "index": 999,
        "address": "1BN69Mnr48BDVgwXV7cKBBsCJ2iB6EHxUa"
    }
]
```


Parameters:

- `batch_no` *(required) (int)* - batch no.

Note: batch no begins from 0, and each batch has 1,000 addresses (address index begins from 0 too).

2.2.4 Address History

GET /api/v1/open/address/history/ *(int: path)*

Returns history addresses of certain path.

Example response:


```
[
    {
        "index": 2,
        "address": "1P4HA6XFs1j8S9FZtakhbCrT8JcZzWwH9h"
    },
    {
        "index": 1,
        "address": "1F1BnSuNwidFvmBmCHcqiWGePkzBfBTtgF"
    },
    {
        "index": 0,
        "address": "15qbwdkwUzUkk3PQ7HVSwJXJfNashY51S2"
    }
]
```


Parameters:

- `path` *(required) (int)* - path for receiving or change addresses.
- `since_address` *(optional) (string)* - from which address to list.

Note:

- `path` 0 means receiving addresses, and path 1 means change addresses.
- `since_address` null represents that start from the beginning.

2.2.5 Address Next

GET /api/v1/open/address/next

Returns next unused receiving address.

Example response:

```
{
    "index": 1,
    "address": "1Mzvo2UkBnEpQC8xkui23mYQQeL8NREqrJ"
}
```

2.2.6 Tx List

GET /api/v1/open/tx

Returns all related txs, and support pages.

Example response:

```
[
    {
        "tx_hash":
↪"d79deb9419b3cf62f08badef456a75396400bfe78cc38ed5ee6cbba27caf57e6",
        "confirmation": 3716,
        "confirm_at": "2016-08-04T12:32:27",
        "value": 500000,
        "tx_at": "2016-08-04T12:22:20",
        "inputs": [
            {
                "is_mine": false,
                "value": 620008,
                "prev_out_sn": 1,
                "sn": 0,
                "address": "1AdGeNK9aorEBmsgFtbu1ENZbPizxnHfeZ",
                "prev_tx_hash":
↪"00ba3370e8c030e42cb43e9861d3d393dfa3e4e157c1ac2e8e5c5f69bda7e4c4"
            }
        ],
        "outputs": [
            {
                "status": 1,
                "is_mine": true,
                "sn": 0,
                "value": 500000,
                "address": "1BhtAhTmXu98JYQW9V4vvKAa6NJ1A7npJ1"
            },
            {
                "status": 1,
```

```
                "is_mine": false,
                "sn": 1,
                "value": 100008,
                "address": "1KU9Vh4HzfYJMBo4QW3ytLdrd9bz6ghJSj"
            }
        ]
    },

    {
        "tx_hash":
↪"576aa53f24b2aaa12e35583caf7fe32f9de8569e440058b1468cb42b417cc48f",
        "confirmation": 3715,
        "confirm_at": "2016-08-04T12:37:07",
        "value": -110000,
        "tx_at": "2016-08-04T12:24:34",
        "inputs": [
            {
                "is_mine": true,
                "value": 500000,
                "prev_out_sn": 0,
                "sn": 0,
                "address": "1BhtAhTmXu98JYQW9V4vvKAa6NJ1A7npJ1",
                "prev_tx_hash":
↪"d79deb9419b3cf62f08badef456a75396400bfe78cc38ed5ee6cbba27caf57e6"
            }
        ],
        "outputs": [
            {
                "status": 0,
                "is_mine": false,
                "sn": 0,
                "value": 100000,
                "address": "1Bitpie7nzdqcsHWYMVi4ePuY88hF2jr7E"
            },
            {
                "status": 1,
                "is_mine": true,
                "sn": 1,
                "value": 390000,
                "address": "1M5CKnUywtfQBicDWFfbRQU7fVUACc71oT"
            }
        ]
    },

    {
        "tx_hash":
↪"f4f30ddf3379b30b49d1128989949e7e8d2754430fef56cb9e78e0d006371933",
        "confirmation": 3715,
        "confirm_at": "2016-08-04T12:37:07",
        "value": -110000,
        "tx_at": "2016-08-04T12:26:08",
        "inputs": [
            {
                "is_mine": true,
                "value": 390000,
                "prev_out_sn": 1,
                "sn": 0,
                "address": "1M5CKnUywtfQBicDWFfbRQU7fVUACc71oT",
```

```
                "prev_tx_hash":
↪"576aa53f24b2aaa12e35583caf7fe32f9de8569e440058b1468cb42b417cc48f"
            }
        ],
        "outputs": [
            {
                "status": 0,
                "is_mine": false,
                "sn": 0,
                "value": 100000,
                "address": "1Bitpie7nzdqcsHWYMVi4ePuY88hF2jr7E"
            },
            {
                "status": 1,
                "is_mine": true,
                "sn": 1, "value": 280000,
                "address": "13iUXsqRuuEXJGUtr9KKopsSnDpBQtqEQM"
            }
        ]
    }
]
```


Arguments:

- `tx_hash` *(optional) (str)* - from which tx_hash to retrieve txs.

Note: txs are ordered by tx_time in asc. if tx_hash is not provided, it means to retrieve the earlist 20 txs. if provided, then returns 20 txs since this tx_hash. The is_mine field in inputs and outputs list should be used to determine whether this input's or output's address belongs to the user.

2.2.7 Tx Detail

GET /api/v1/open/tx/detail/ *(str: tx_hash)*

Returns tx details for specific tx.

Example response:


```
{
    "tx_hash":
↪"1c56ac562ad216f3fe68e8142a7f28f4889886065cc77745af4c9ddaee17c6e1",
    "inputs": [
        {
            "is_mine": false,
            "value": 2510000,
            "prev_out_sn": 1,
            "sn": 0,
            "address": "19EbsdkWfihtQHn4CoN5Eoyzh7cqqPNSLR",
            "prev_tx_hash":
↪"6aac1207ced737dc5a576dc5812a1c38687a760e2e148e6e6b1f12c1bf121dc0"
        }
    ],
    "confirmation": 908,
    "outputs": [
        {
```

```
            "status": 1,
            "is_mine": true,
            "sn": 0,
            "value": 10000,
            "address": "1B4qK6KTzWCnbNcx4WDYR99KfVRsU2zDcN"
        },
        {
            "status": 1,
            "is_mine": false,
            "sn": 1,
            "value": 2490000,
            "address": "1BU1p3PU9MRcPrvggBmDEXokigZy5cxQso"
        }
    ],
    "confirm_at": "2016-07-27T02:31:34",
    "value": 10000,
    "tx_at": "2016-07-27T02:21:53"
}
```


Parameters:

- `tx_hash` *(required) (str)* - for which tx_hash to retrieve tx detail.

2.3 Cold Receiving API

2.3.1 Tx List

GET /api/v1/open/tx

Returns all related txs, and support pages.

Example response:


```
[
    {
        "tx_hash":
↪"d79deb9419b3cf62f08badef456a75396400bfe78cc38ed5ee6cbba27caf57e6",
        "confirmation": 3716,
        "confirm_at": "2016-08-04T12:32:27",
        "value": 500000,
        "tx_at": "2016-08-04T12:22:20",
        "inputs": [
            {
                "is_mine": false,
                "value": 620008,
                "prev_out_sn": 1,
                "sn": 0,
                "address": "1AdGeNK9aorEBmsgFtbu1ENZbPizxnHfeZ",
                "prev_tx_hash":
↪"00ba3370e8c030e42cb43e9861d3d393dfa3e4e157c1ac2e8e5c5f69bda7e4c4"
            }
        ],
        "outputs": [
            {
                "status": 1,
                "is_mine": true,
```

```
                "sn": 0,
                "value": 500000,
                "address": "1BhtAhTmXu98JYQW9V4vvKAa6NJ1A7npJ1"
            },
            {
                "status": 1,
                "is_mine": false,
                "sn": 1,
                "value": 100008,
                "address": "1KU9Vh4HzfYJMBo4QW3ytLdrd9bz6ghJSj"
            }
        ]
    },

    {
        "tx_hash":
↪"576aa53f24b2aaa12e35583caf7fe32f9de8569e440058b1468cb42b417cc48f",
        "confirmation": 3715,
        "confirm_at": "2016-08-04T12:37:07",
        "value": -110000,
        "tx_at": "2016-08-04T12:24:34",
        "inputs": [
            {
                "is_mine": true,
                "value": 500000,
                "prev_out_sn": 0,
                "sn": 0,
                "address": "1BhtAhTmXu98JYQW9V4vvKAa6NJ1A7npJ1",
                "prev_tx_hash":
↪"d79deb9419b3cf62f08badef456a75396400bfe78cc38ed5ee6cbba27caf57e6"
            }
        ],
        "outputs": [
            {
                "status": 0,
                "is_mine": false,
                "sn": 0,
                "value": 100000,
                "address": "1Bitpie7nzdqcsHWYMVi4ePuY88hF2jr7E"
            },
            {
                "status": 1,
                "is_mine": true,
                "sn": 1,
                "value": 390000,
                "address": "1M5CKnUywtfQBicDWFfbRQU7fVUACc71oT"
            }
        ]
    },

    {
        "tx_hash":
↪"f4f30ddf3379b30b49d1128989949e7e8d2754430fef56cb9e78e0d006371933",
        "confirmation": 3715,
        "confirm_at": "2016-08-04T12:37:07",
        "value": -110000,
        "tx_at": "2016-08-04T12:26:08",
        "inputs": [
```

```
            {
                "is_mine": true,
                "value": 390000,
                "prev_out_sn": 1,
                "sn": 0,
                "address": "1M5CKnUywtfQBicDWFfbRQU7fVUACc71oT",
                "prev_tx_hash":
↪"576aa53f24b2aaa12e35583caf7fe32f9de8569e440058b1468cb42b417cc48f"
            }
        ],
        "outputs": [
            {
                "status": 0,
                "is_mine": false,
                "sn": 0,
                "value": 100000,
                "address": "1Bitpie7nzdqcsHWYMVi4ePuY88hF2jr7E"
            },
            {
                "status": 1,
                "is_mine": true,
                "sn": 1, "value": 280000,
                "address": "13iUXsqRuuEXJGUtr9KKopsSnDpBQtqEQM"
            }
        ]
    }
]
```

Arguments:

- `tx_hash` *(optional) (str)* - from which tx_hash to retrieve txs.

Note: txs are ordered by tx_time in asc. if tx_hash is not provided, it means to retrieve the earlist 20 txs. if provided, then returns 20 txs since this tx_hash. The is_mine field in inputs and outputs list should be used to determine whether this input's or output's address belongs to the user.

2.4 Hot Sending API

2.4.1 Hot Sending Tx Request

POST /api/v1/open/tx/request

Post Hot Sending Tx Request.

Example response:

```
{
    "result": true
}
```

Parameters:

- `coin_code` *(optional) (string)* - coin type, for example *(BTC)*.

- `user_tx_no` *(required) (string)* - tx id in vweb.
- `outs` *(required) (string)* - transfer to address and value.
- `vc_code` *(required) (string)* - the sign of unsigntx.
- `is_dynamic_fee` *(optional) (int)* - whether to charge the dynamic fee.
- `c_id` *(required) (int)* - the channel id of sms to sign.

Note:

- `coin_code` default *(BTC)*.
- `user_tx_no` unique in vweb.
- `outs` support multiple addresses.for example "1NbbvxBYxGGCBhaM8mow1HFWA7dB5yukmY,2000;1XaavxBYxGGCBhaM8mow1HFWA7dB5yukmY,3000";
- `is_dynamic_fee` 0 no 1 yes default 1.
- `vc_code` and `c_id` is use to ensure security .

2.4.2 Hot Sending Tx Detail

GET /api/v1/open/tx/ *(int: user_tx_no)*

Returns tx details for your specific tx_no.

Example response:

```
{
    "tx_hash":
↪"897a7eb61ea4e9b5a72afa95573ccdc67d4edb3984509a11316664cb69a18065",
    "c_id": 2,
    "user_id": 40000,
    "send_request": {
        "coin_code": "BTC",
        "user_tx_no": "8",
        "is_dynamic_fee": 1,
        "outs": "1NbbvxBYxGGCBhaM8mow1HFWA7dB5yukmY,2000;
↪1XaavxBYxGGCBhaM8mow1HFWA7dB5yukmY,3000"
    },
    "vc_code":
↪"HxAK9Q4CdcKzypai9Wk4gjYwC8jeuHq9UWunAvyzRbO1a4PyZecmYF0WS5kdtBH80/
↪0EtSETjurHyRctkCFsxVk=",
    "request_at": "2016-07-31T04:52:13",
    "hot_wallet_tx_id": 26,
    "tx_at": "2016-07-31T05:31:07",
    "hot_wallet_tx_status": 1
}
```

Parameters:

- `user_tx_no` *(required) (int)* - for which user_tx_no to retrieve tx detail.

Note: user_tx_no is the unique transaction id in your own system, you should use an auto incremented primary key as user_tx_no.

2.5 Report API

Hello

2.5.1 Hello

Hello

CHAPTER 3

ChainCloud-V

3.1 What is V-Device?

V-Device is like a "2-Step Verification" Device for you.

V-Device is an Android device running with ChainCloud-V (an Open Source App developped by us). With this App, you can establish a secured channel between your V-Device and ChainCloud's HSM-Cold with SMS. All data transfered on this channel should be verified to ensure the security.

There are two different verifications between V-Device and ChainCloud HSM-Cold:

- All Hot-Sending requests (from your website/service) shoud be signed by your V Device, and the signature will pass through ChainCloud API to HSM-Hot Device, and then to HSM-Cold. After that, the HSM-Cold will verify the signature with data. If the signature is correct, HSM-Cold will acknowlege the Hot-Sending request, otherwise, it will close the channel and notify the administrator.
- When you request the Cold-Receiving/Hot-Sending addresses from ChainCloud API, and plan to use these addresses in your business logic, you can verify each batch of these addresses (1000 address per batch) with the security channel between V Device and HSM-Cold. You should verify all addresses before use, to protect your blockchain assets.

You can also develop your own version of ChainCloud-V (because it is Open Source), and add other safe strategies, for example :

- Limitation of destination addresses;
- Strategies of sending amount;
- Strategies of sending frequency;
- Special encryption strategy between your website/service and your V Device;

V-Device should always be powered on (connect to the charger), and you should keep chaincloud-v running. The chaincloud-v will periodically establish new channel with HSM-Cold, and all new data will be secured by this newly created channel. If you have powered off the V-Device or exit the chaincloud-v, next time when you want to use V-Device for verification, you have to wait for the new channel to be created (normally midnight). If you want to use V-Device immediately, you can try to contact our technical support.

3.2 App: chaincloud-v

The chaincloud-v is an open source software running on the V-Device, and automatically ensures the security.

Source Code

Latest Release

3.3 V-Device Sample

In following docs, we will describe how to build up a web-service that can be used by V-Device

3.3.1 V-Test

Get A Tx To Be Sended

GET /api/v1/open/vtest

Get A Tx To Be Sended.

Example response:

```
{
    "vtest_info":
↪"B23A7DDF7AF953D4344ED8B190E9424C2A8D85CAA00B4358D58C5341E36A194097EA21F7D457E770E2C951283
↪166EAB021AFB6244191EEE75340F8109/D267F338A21B487D",
    "vtest_at": "2016-08-18T06:00:53",
    "vtest_id": 11
}
```

Note:

- `vtest_info` encrypt with aes. decryption is {“outs”:”1NbbvxBYxGGCBhaM8mow1HFWA7dB5yukmY,10000;1MCs9SZwLg9JvLo6pzvVBWtmSV1dakwyM1,1 passwd is “20160721”. The unit of value is in satoshi, the minimum is 10000 satoshi.
- returns NULL if data is empty.

Update A Tx Status

POST /api/v1/open/vtest

Update A Tx Status

Example response:

```
{
    "result": true
}
```

Parameters:

- `vtest_id` *(required) (string)* - vtest_id.

- `tx_hash` *(optional) (string)* - tx hash.

Note:

- `tx_hash` if hash is null, it is failed, else success.

Get A Batch Addresses To Be Checked

GET /api/v1/open/batch

Get A Batch Address To Be Checked.

Example response:

```
{
    "batch_no": 0,
    "address_list": [
        {
            "index": 0,
            "address": "1B4qK6KTzWCnbNcx4WDYR99KfVRsU2zDcN"
        },
        {
            "index": 1,
            "address": "1Mzvo2UkBnEpQC8xkui23mYQQeL8NREqrJ"
        },
        "......",
        {
            "index": 998,
            "address": "1KqPMBXtmvKkLe9TCHzVR4New3c5xkVgge"
        },
        {
            "index": 999,
            "address": "1BN69Mnr48BDVgwXV7cKBBsCJ2iB6EHxUa"
        }
    ],
    "address_type": 1
}
```

Note:

- `batch_no` batch number that start from 0.
- `address_list` to be checked address, 1000 in all.
- `address_type` 1->HOT SEND ADDRESS 2->COLD RECEIVE ADDRESS.
- returns NULL if data is empty.

Update A Batch Addresses Status

POST /api/v1/open/batch

Update A Batch Addresses Status

Example response:

```
{
    "result": true
}
```

Parameters:

- `batch_no` *(required) (int)* - batch number that start from 0.
- `status` *(required) (int)* - 2->SUCCESS 3->FAILED
- `type` *(required) (int)* - 1->HOT SEND ADDRESS 2->COLD RECEIVE ADDRESS.

Note:

- `status` check result
- `type` address type

3.4 SMS Verification

V-Device can interact verifaction data with ChainCloud's HSM-Cold through SMS messages.

HSM-Cold has a built-in SMS module, so it can communicate verification data with V-Device through SMS messages.

Currently, we support two different SIM cards, and can be used with two different carriers for Fail-Safe configuration.

CHAPTER 4

Pricing

4.1 ChainCloud Pricing

ChainCloud has a flexible pricing model suitable from small business to large enterprise.

4.1.1 Basic Pricing

The basic pricing model is designed to scale with your business :

- **0.10% per incoming tx**
  - Truncate to mBTC
  - Pay only for what you use
  - Real-time fee reporting
- **Volume based discounts available**
  - ChainCloud offers everything needed to run a business at scale
  - You can contact our sales for more details

4.1.2 Account Pricing

The account pricing model is fixed. You can pay once, and use always :

- **10 mBTC per account**
  - One Cold Receiving Token
  - One Hot Sending Token
  - One ChainCloud-V account

4.1.3 Address Pricing

The address pricing model is based on how many addresses used by your business :

- **1 mBTC per batch of addresses**
  - Each batch has 1,000 addresses
  - Both Hot/Cold addresses are using this pricing model
  - Actually Hot addresses are not used as much as Cold addresses, because you only need Hot addresses when the fund in Hot wallet is not enough
  - Change addresses are not counted in this pricing model

4.1.4 Others

Other pricing models :

- **10mBTC for basic information modification**
  - Modify V-Device phone no
  - Modify Cold-Receiving Settlement HD Account (Bitpie Account)
- You should buy an Android phone (with 2 sim-cards support), and pay the phone/sms bills yourself.
- **Currently, 10 BTCs(nonrefundable) are needed to be paid by the enterprise user to activate the ChainCloud account, and**
  - If you decided to start your ChianCloud service after contacting our managers, you should pay 10 BTCs to the following address: 1Bitpie7nzdqcsHWYMVi4ePuY88hF2jr7E.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search