

比特币中国交易 API 文档

- 版本
- 交易 API 访问认证
 - 详细步骤
 - API 认证示例代码
- 交易 API v1
 - 错误代码
 - 返回对象
 - profile
 - balance
 - frozen
 - order
 - withdrawal
 - deposit
 - API 方法
 - buyOrder
 - 参数
 - 返回
 - cancelOrder
 - 参数
 - 返回
 - getAccountInfo
 - 参数
 - 返回
 - getDeposits
 - 参数
 - 返回
 - getMarketDepth
 - 参数
 - 返回
 - getOrder
 - 参数
 - 返回
 - getOrders
 - 参数
 - 返回
 - getWithdrawal
 - 参数
 - Returns
 - getWithdrawals
 - 参数
 - 返回
 - requestWithdrawal
 - 参数
 - 返回
 - sellOrder
 - 参数
 - 返回

版本

日期	描述
2013-09-02	初始版本

交易 API 访问认证

要进行交易 API 的访问认证，您需要将公开的访问密钥（Access Key）以及通过秘密密钥（Secret Key）签名得到的一个哈希字符串传递给交易 API。签名所采用的算法是 HMAC（Hash-based Message Authentication Code）。除了访问密钥和秘密密钥以外，您还需要传递一个“Tonce”参数。Tonce 是指以毫秒为单位的当前时间的时间戳（Timestamp）。访问密钥和秘密密钥生成的签名通过 HTTP 基本身份验证（HTTP Basic Authentication）传递，Tonce 通过 HTTP 头（HTTP Header）传递。

详细步骤

1. 使用如下强制的参数创建签名字符串。将键值对使用“&”符号按照下面列表的顺序连接在一起。请注意连接顺序很重要。所有的键名必须添加，但是键值可以为空（例如 params）。
 - a. tonce（以毫秒为单位的时间戳，请确保您的系统时间准确）
 - b. accesskey（访问密钥，您可以在您的账户管理页面申请）
 - c. method（HTTP 请求方法，目前仅支持“post”）
 - d. id（JSON-RPC 请求 id）
 - e. method（JSON-RPC 方法名称）
 - f. params（JSON-RPC 方法参数）

```
### 示例 1 ###
tonce=1377743828095093
&accesskey=1d87effa-e84d-48c1-a172-0232b86305dd
&requestmethod=post
&id=1
&method=getAccountInfo
&params=

### 示例 2 ###
tonce=1377743828095093
&accesskey=1d87effa-e84d-48c1-a172-0232b86305dd
&requestmethod=post
&id=1
&method=buyOrder
&params=500,1
```

2. 使用您的秘密密钥用 HMAC 生成哈希签名。使用 sha1 作为哈希算法。

```
### PHP ###
hash_hmac('sha1', $signature, $secretkey)
```

3. 使用 HTTP 基本身份验证传递身份验证信息。您可以通过如下两种方式实现：
 - a. 在网址中传递。您可以在网址中直接传递访问密钥和哈希签名

```
https://<accesskey>:<hash>@btcchina.com/api_trade_v1.php
```

- b. HTTP 认证头。计算并创建认证头（Authorization），并和其他请求一起传递到 API。身份信息为用冒号（:）连接的访问密钥和哈希签名。最后的认证头的值需要用 Base64 编码。

```
### PHP ###
base64_encode(<accesskey>:<hash>) //PGFjY2Vzc2tleT46PGhhc2g+

# HTTP HEADER
Authorization: Basic PGFjY2Vzc2tleT46PGhhc2g+
```

- 最后您需要将 Tonce 也作为 HTTP 头传递给 API。Tonce 的值必须和您在第 1 步生成哈希签名时使用的值一致。

```
# HTTP HEADER
Json-Rpc-Tonce: 1377743828095093
```

- 发起 API 请求。如果有错误，API 会返回 HTTP 401 Unauthorized。

API 认证示例代码

Python 代码

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import time
import hmac
import hashlib
import base64
import httplib
import logging

# 在这类设置您的个人 API 访问密钥和秘密密钥
access_key="<access key>"
secret_key="<secret key>"

tonce=str(int(time.time()*1000000))
params='tonce='+tonce+'&accesskey='+access_key+'&requestmethod=post&id=1&method=getAccountInfo&params='
params_hash = hmac.new(secret_key, params, hashlib.sha1).hexdigest()
base64string = base64.b64encode(access_key+'_'+params_hash)

headers = {
    'Content-type': 'application/json-rpc',
    'Authorization': 'Basic '+base64string,
    'Json-Rpc-Tonce': tonce
}

postdata = '{"method": "getAccountInfo", "params": [], "id": 1}'

conn=httplib.HTTPSConnection("btcchina.com")
conn.request("POST", '/api_trade_v1.php', postdata, headers)
response = conn.getresponse()

print "\nresponse: ", response.read()
```

交易 API v1

交易 API 是基于 JSON-RPC 2.0 实现的。

要了解更多关于 JSON-RPC 2.0 的规范，请参考其官方文档。

<http://www.jsonrpc.org/specification>

各种流行编程语言的 JSON-RPC 实现可以在如下网址找到。

<http://json-rpc.org/wiki/implementations>

错误代码

所有 API 方法调用在请求失败或遇到未知错误时会返回 JSON-RPC 错误对象。

代码	消息
-32000	内部错误
-32003	人民币账户余额不足
-32004	比特币账户余额不足
-32005	挂单没有找到
-32006	无效的用户
-32007	无效的货币
-32008	无效的金额
-32009	无效的钱包地址
-32010	没有找到提现记录
-32011	没有找到充值记录

错误响应示例

```
{
  "error": {
    "code": -32003,
    "message": "Insufficient CNY balance",
    "id": 1
  }
}
```

返回对象

成功的方法调用会返回如下对象。这些对象的示例会在方法文档中列出。

profile

名称	类型	描述
username	string	账户用户名
trade_password_enabled	boolean	指示该账户是否已设置交易密码
otp_enabled	boolean	指示是否已启用双重认证
trade_fee	number	账户交易手续费
daily_btc_limit	number	账户每日比特币提现限额
btc_deposit_address	string	比特币充值地址
btc_withdrawal_address	string	比特币提现地址

balance

名称	类型	描述
currency	string	货币代码
Symbol	string	货币符号
amount	number	账户余额（小数精度）
amount_integer	string	账户余额（整数精度）
amount_decimal	integer	在使用“amount_integer”时小数点的位置

frozen

名称	类型	描述
currency	string	货币代码
Symbol	string	货币符号
amount	number	账户冻结金额（小数精度）
amount_integer	string	账户冻结金额（整数精度）
amount_decimal	integer	在使用“amount_integer”时小数点的位置

order

名称	类型	描述
id	integer	挂单 ID
type	string	挂单类型。可能值：bid 或 ask
price	number	价格
currency	string	货币代码。可能值：CNY 或 BTC
amount	number	挂单剩余数量。如果此值小于 amount_original，说明此挂单仅有部分成交
amount_original	number	初始挂单数量
date	integer	Unix 时间戳。自1970年1月1日以来的秒数

status	string	挂单状态。可能值：open、closed 或 cancelled
--------	--------	----------------------------------

withdrawal

名称	类型	描述
id	integer	提现 ID
address	string	比特币提现地址
currency	string	货币代码。可能值：CNY 或 BTC
amount	number	提现数量
date	integer	Unix 时间戳。自1970年1月1日以来的秒数
transaction	string	Bitcoin 交易 ID
status	string	提现状态。可能值：pending 或 completed

deposit

名称	类型	描述
id	integer	存款 ID
address	string	比特币充值地址
currency	string	货币代码。可能值：CNY 或 BTC
amount	number	充值数量
date	integer	Unix 时间戳。自1970年1月1日以来的秒数
status	string	充值状态。可能值：pending 或 completed

API 方法

buyOrder

下比特币买单。

参数

名称	类型	必选？	描述
price	number	是	买 1 比特币所用人民币的价格，最多支持小数点后 5 位精度
amount	number	是	要买的比特币数量，最多支持小数点后 8 位精度

JSON 请求示例

```
{"method": "buyOrder", "params": [500, 1], "id": 1}
```

返回

名称	类型	描述
result	boolean	如果下单成功，返回 true

JSON 响应示例

```
{"result":true,"id":"1"}
```

cancelOrder

取消一个还未完全成交的挂单，其状态应该为“open”。

参数

名称	类型	必选？	描述
id	number	是	要取消的挂单的 ID

JSON 请求示例

```
{"method":"cancelOrder","params":[2],"id":1}
```

返回

名称	类型	描述
result	boolean	如果取消挂单成功，返回 true

JSON 响应示例

```
{"result":true,"id":"1"}
```

getAccountInfo

获取账户信息和余额。

参数

无

JSON 请求示例

```
{"method": "getAccountInfo", "params": [], "id": 1}
```

返回

名称	类型	描述
result	object[]	包含如下对象: profile, balance, frozen

JSON 响应示例

```
{
  "result": {
    "profile": {
      "username": "btc",
      "trade_password_enabled": true,
      "otp_enabled": true,
      "trade_fee": 0.003,
      "daily_btc_limit": 10,
      "btc_deposit_address": "123myZyM9jBYGw5EB3wWmfgJ4Mvqnu7gEu",
      "btc_withdrawal_address": "123GzXJnfugniyy7ZDw3hSjkm4tHPHzHba"
    },
    "balance": {
      "btc": {
        "currency": "BTC",
        "symbol": "\u00e3f",
        "amount": 9999.4997,
        "amount_integer": "999949970000",
        "amount_decimal": 8
      },
      "cny": {
        "currency": "CNY",
        "symbol": "\u00a5",
        "amount": 998999.99339,
        "amount_integer": "99899999339",
        "amount_decimal": 5
      }
    },
    "frozen": {
      "btc": {
        "currency": "BTC",
        "symbol": "\u00e3f",
        "amount": 0,
        "amount_integer": "",
        "amount_decimal": 8
      },
      "cny": {
        "currency": "CNY",
        "symbol": "\u00a5",
        "amount": 0,
        "amount_integer": "",
        "amount_decimal": 5
      }
    }
  }
}
```

getDeposits

获得用户全部充值记录。

参数

名称	类型	必选？	描述
currency	string	是	目前仅支持“BTC”
pendingonly	boolean	否	默认为“true”。如果为“true”，仅返回尚未入账的比特币充值

JSON 请求示例

```
{ "method": "getDeposits", "params": ["BTC"], "id": 1 }
```

返回

名称	类型	描述
result	object[]	包含对象: deposit

JSON 响应示例

```
{ "result": [ { "deposit": [ { "id": 49751, "address": "mufUAWHCius1jZpjB4zCUwzaRbYuwXCupC", "currency": "BTC", "amount": 1, "date": 1376910685, "status": "pending" }, { "id": 49749, "address": "mkrmyZyM9jBYGw5EB3wWmfgJ4Mvqnu7gEu", "currency": "BTC", "amount": 2, "date": 1376906645, "status": "completed" } ] } ] }
```

getMarketDepth

获得完整的市场深度。返回全部尚未成交的买单和卖单。

参数

无

JSON 请求示例

```
{"method": "getMarketDepth", "params": [], "id": 1}
```

返回

名称	类型	描述
result	object[]	对象数组: market_depth

JSON 响应示例

```
{
  "result": [
    {
      "market_depth": [
        {
          "price": "500.00",
          "totalamount": "2.00000000",
          "type": "bid"
        }
      ]
    }
  ],
  "id": "1"
}
```

getOrder

获得挂单状态。

参数

名称	类型	必选?	描述
id	number	是	挂单 ID

JSON 请求示例

```
{"method": "getOrder", "params": [2], "id": 1}
```

返回

名称	类型	描述
result	object	返回对象: order

JSON 响应示例

```
{
  "result": {
    "order": {
      "id": 2,
      "type": "bid",
      "price": 500,
      "currency": "cny",
      "amount": 0.9,
      "amount_original": 0.9,
      "date": 1377671476,
      "status": "cancelled"
    }
  },
  "id": "1"
}
```

getOrders

获得全部挂单的状态。

参数

名称	类型	必选？	描述
openonly	boolean	否	默认为“true”。如果为“true”，仅返回还未完全成交的挂单。

JSON 请求示例

```
{
  "method": "getOrder",
  "params": [2],
  "id": 1
}
```

返回

名称	类型	描述
result	object[]	对象数组：order

JSON 响应示例

```
{
  "result": {
    "order": [
      {
        "id": 2,
        "type": "bid",
        "price": 500,
        "currency": "cny",
        "amount": 0.9,
        "amount_original": 0.9,
        "date": 1377671476,
        "status": "cancelled"
      }, {
        "id": 3,
        "type": "bid",
        "price": 501,
        "currency": "cny",
        "amount": 0.8,
        "amount_original": 0.8,
        "date": 1377671475,
        "status": "cancelled"
      }
    ],
    "id": "1"
  }
}
```

getWithdrawal

获取提现状态。

参数

名称	类型	必选?	描述
id	number	是	提现 ID

JSON 请求示例

```
{"method": "getWithdrawal", "params": [1], "id": 1}
```

Returns

名称	类型	描述
result	object	返回对象: withdrawal

JSON 响应示例

```
{
  "result": {
    "withdrawal": {
      "id": "20351",
      "address": "15MGzXJnfugniyy7ZDw3hSjkm4tHPHzHba",
      "currency": "BTC",
      "amount": 0.1,
      "date": 1376891209,
      "transaction": null,
      "status": "pending"
    }
  },
  "id": "1"
}
```

getWithdrawals

获取全部提现记录。

参数

名称	类型	必选？	描述
currency	string	是	目前仅支持“BTC”
pendingonly	boolean	否	默认为“true”。如果为“true”，仅返回尚未处理的提现记录

JSON 请求示例

```
{"method": "getWithdrawals", "params": ["BTC"], "id": 1}
```

返回

名称	类型	描述
result	object[]	对象数组：withdrawal

JSON 响应示例

```
{
  "result": {
    "withdrawal": [
      {
        "id": "20351",
        "address": "15MGzXJnfugniyy7ZDw3hSjkm4tHPHzHba",
        "currency": "BTC",
        "amount": 0.1,
        "date": 1376891209,
        "transaction": null,
        "status": "pending"
      },
      {
        "id": "20352",
        "address": "15MGzXJnfugniyy7ZDw3hSjkm4tHPHzHba",
        "currency": "BTC",
        "amount": 0.1,
        "date": 1376891268,
        "transaction": null,
        "status": "pending"
      }
    ],
    "id": "1"
  }
}
```

requestWithdrawal

发起比特币提现请求。

参数

名称	类型	必选?	描述
currency	string	是	货币代码。可能值: BTC 或 CNY
amount	number	是	提现金额

JSON 请求示例

```
{"method": "requestWithdrawal", "params": ["BTC", 0.1], "id": 1}
```

返回

名称	类型	描述
result	string	返回提现 ID

JSON 响应示例

```
{"result": {"id": "20362"}, "id": "1"}
```

sellOrder

下比特币卖单。

参数

名称	类型	必选?	描述
price	number	是	卖 1 比特币所用人民币的价格，最多支持小数点后 5 位精度
amount	number	是	要卖的比特币数量，最多支持小数点后 8 位精度

JSON 请求示例

```
{"method":"sellOrder","params":[500,1],"id":1}
```

返回

名称	类型	必选?
result	boolean	如果下单成功，返回 true。

JSON 响应示例

```
{"result":true,"id":"1"}
```