

API 文档

- 1. 介绍(参考[jsonrpc](#))
 - 1.1. 统一的请求格式
 - 1.2. 统一的结果格式
 - 1.3. 错误代码表
 - 1.4. 名词解释
- 2. Asset API
 - 2.1. [balance.query](#)
 - 2.2. [asset.list](#)
 - 2.3. [asset.max_amount](#)
 - 2.4. [asset.min_amount](#)
 - 2.5. [asset.price_range](#)
 - 2.6. [balance.query_all](#)
- 3. Trade API
 - 3.1. [order.book](#)
 - 3.2. [order.put_limit](#)
 - 3.3. [order.put_market](#)
 - 3.4. [order.cancel](#)
 - 3.5. [order.deals](#)
 - 3.6. [order.depth](#)
 - 3.7. [order.pending](#)
 - 3.8. [order.finished](#)
 - 3.9. [order.finished_detail](#)
 - 3.10. [order.pending_detail](#)
 - 3.11. [order.history](#)
- 4. Market API
 - 4.1. [market.deals](#)
 - 4.2. [market.kline](#)
 - 4.3. [market.status](#)
 - 4.4. [market.list](#)
 - 4.5. [market.user_deals](#)
 - 4.6. [market.last](#)

1. 介绍(参考[jsonrpc](#))

1.1. 统一的请求格式

- * **method**: 想要调用的方法的名称, `String`
- * **params**: 调用方法所需要的参数, `Array`
- * **id**: Request id, `Integer`

1.2. 统一的结果格式

- * result: Json object, null for failure
- * error: Json object, null for success, non-null for failure
 - 1. code: error code
 - 2. message: error message
- * id: Request id, Integer

1.3. 错误代码表

- * 1: invalid argument
- * 2: internal error
- * 3: service unavailable
- * 4: method not found
- * 5: service timeout

1.4. 名词解释

1.4.1. limit order:

限价单，就是用户自己设价格挂的单

1.4.2. market order:

市价单，市价单是按市场当时最优价或市价立即购买或出售一定数量期货合约的指令(区别限价单，市价单的价格是根据市场自动选择的)

1.4.3. market

交易市场，如"BTCUSDT"代表的是以"BTC"与"USDT"交换的市场

1.4.4. offset

查询列表，起始位置(请和[limit](#limit)一起看)

1.4.5. limit

查询列表，最多几笔(请和[offset](#offset)一起看)

1.4.6. ctime

订单被created的时间

1.4.7. deal_stock

订单被created的时间

1.4.8. template

解释

2. Asset API

2.1. balance.query

2.1.1. 解释

查询现在账户里的余额

2.1.2. 参数

1. **asset**: 货币的名称, **String**

2.1.3. 返回值

用户的资产状态

2.1.4. 示例

```
{
  "params": ["BTC", "ETH"],
  "result": {
    "BTC": {
      "available": "277795.43924609",
      "freeze": "1",
```

```
        "CNY": "999999.999"
      },
      "ETH": {
        "available": "277.09",
        "freeze": "1",
        "CNY": "999999.999",
      },
    }
  }
}
```

2.2. asset.list

2.2.1. 解释

得到所有交易所可交易的产品(?)

2.2.2. 参数:

[]

2.2.3. 返回值

可交易的产品的名字和精度(?)

2.2.4. 示例

```
{
  "params": [],
  "result": [
    {
      "name": "ETH",
      "prec": 8
    },
    {
      "name": "BTC",
      "prec": 8
    },
    {
      "name": "BHB",
      "prec": 8
    },
    {
      "name": "USDT",
      "prec": 8
    }
  ]
}
```

```
    ]  
  }
```

2.3. asset.max_amount

2.3.1. 解释

得到某个货币的每单最大交易量

2.3.2. 参数:

1. **asset**: 货币名; **String**

2.3.3. 返回值

货币的每单最大交易量

2.3.4. 示例

```
{  
  "params": ["BTC"],  
  "result": "10"  
}
```

2.4. asset.min_amount

2.4.1. 解释

得到某个货币的每单最小交易量

2.4.2. 参数:

1. **asset**: 货币名; **String**

2.4.3. 返回值

货币的每单最小交易量

2.4.4. 示例

```
{
  "params": ["BTC"],
  "result": "0.0001"
}
```

2.5. asset.price_range

2.5.1. 解释

得到某个货币离当前价格的极端偏离值(比当前最低价低多少,比当前最高价高多少)

2.5.2. 参数:

1. **asset**: 货币名; **String**

2.5.3. 返回值

[下限比例, 上限比例]

2.5.4. 示例

```
{
  "params": ["BTC"],
  "result": ["0.7", "1.3"]
}
```

2.6. balance.query_all

2.6.1. 解释

查询指定用户账户里所有资产的余额

2.6.2. 参数

None

2.6.3. 返回值

用户的所有资产列表状态

2.6.4. 示例

```
{
  "params": [],
  "result": {
    "BTC": {
      "available": "277795.43924609",
      "freeze": "1"
    },
    "ETH": {
      "available": "277795.43924609",
      "freeze": "1"
    }
  }
}
```

3. Trade API

3.1. order.book

3.1.1. 解释

查看交易委托账本(`order.depth`应该是整理了这个?)

交易委托账本就是送到交易所的限价委托单, 如果当时无法成交, 就会自动纪录到交易所的交易委托账本里, 等待成交机会

3.1.2. 参数

1. `market`: 见Dictionary; String
2. `side`: 挂何类型的单; Integer; 1代表卖单, 2代表买单
3. `offset`: 见Dictionary; Integer
4. `limit`: 见Dictionary; Integer

3.1.3. 返回值

result: 交易委托账的详细

3.1.4. 示例

```
{
  "params": ["BTCUSDT", 2, 0, 5],
  "result": {
    "offset": 0,
    "limit": 5,
    "total": 2,
    "orders": [
      {
        "id": 5311964,
        "market": "BTCUSDT",
        "type": 1,
        "side": 2,
        "user": 6,
        "ctime": 1532674506.886209,
        "mtime": 1534245115.528764,
        "price": "81",
        "amount": "2222333",
        "taker_fee": "0.002",
        "maker_fee": "0.001",
        "left": "2218775",
        "deal_stock": "3558",
        "deal_money": "283393",
        "deal_fee": "6.863"
      },
      {
        "id": 5311961,
        "market": "BTCUSDT",
        "type": 1,
        "side": 2,
        "user": 6,
        "ctime": 1532674409.836122,
        "mtime": 1532674409.836122,
        "price": "79",
        "amount": "23",
        "taker_fee": "0.002",
        "maker_fee": "0.001",
        "left": "23",
        "deal_stock": "0e-8",
        "deal_money": "0e-16",
        "deal_fee": "0e-12"
      }
    ]
  }
}
```


3.2. order.put_limit

3.2.1. 解释

挂一个限价单

3.2.2. 参数

1. market: 见Dictionary; String
2. side: 挂何类型的单; Integer; 1代表卖单, 2代表买单
3. amount: 挂单的数量; String
4. price: 挂单的价格; String
5. taker_fee_rate: 吃单人所承担的手续费; String
6. maker_fee_rate: 挂单人所承担的手续费; String
7. source: 订单的客户端来源; String

3.2.3. 返回值

result: 成立的订单的详细资讯

error: 10 - "balance not enough" - 余额不足以完成这一笔交易

3.2.4. 示例

```
{
  "params": ["BTCCNY", 1, "10", "8000", "0.002", "0.001"],
  "result": {
    "id": 5348586,
    "market": "BTCUSDT",
    "source": "web",
    "type": 1,
    "side": 1,
    "user": 6,
    "ctime": 1534828248.857088,
    "mtime": 1534828248.857088,
    "price": "8000",
    "amount": "10",
    "taker_fee": "0.002",
    "maker_fee": "0.001",
    "left": "10",
    "deal_stock": "0",
    "deal_money": "0",
    "deal_fee": "0"
  }
}
```

3.3. order.put_market

3.3.1. 解释

挂一个市价单

3.3.2. 参数

1. market: 见Dictionary; String
2. side:
 - 1: 卖; Integer
 - 2: 买; Integer
3. amount: 交易的数量; String
4. taker_fee_rate: 吃单的手续费; String
5. source: 来源(e.g. web); String

3.3.3. 返回值

result: 订单的详细资料

error: "balance not enough" - 余额不足以完成这一笔交易

3.3.4. 示例

```
{
  "params": ["BTCUSDT", 1, "10", "0.002", "web"],
  "result": {
    "id": 86105,
    "market": "BTCETH",
    "source": "web",
    "type": 2,
    "side": 2,
    "user": 6,
    "ctime": 1531452433.020948,
    "mtime": 1531452433.020954,
    "price": "0",
    "amount": "10",
    "taker_fee": "0.002",
    "maker_fee": "0",
    "left": "0.00008067",
    "deal_stock": "0.00118021",
    "deal_money": "9.99991933",
    "deal_fee": "0.00000236042"
  }
}
```

```
}  
}
```

3.4. order.cancel

3.4.1. 解释

取消一个订单

3.4.2. 参数

1. market: 见Dictionary; String
2. order_id: 订单的ID

3.4.3. 返回值

result: 订单的详细资料

error:

1. 10 - "order not found" - 找不到订单
2. 11 - "user not match" - 取消的订单并不属于这个用户

3.4.4. 示例

```
{  
  "params": ["BTCCNY", 172938712567],  
  "error": {"code": 10, "message": "order not found"}  
}
```

3.5. order.deals

3.5.1. 解释

查看挂的单被成交的详情(?), 例如卖了5个单位的BTC, 可能2个被A买, 3个被B买

3.5.2. 参数

1. order_id: 订单的ID; Integer
2. offset: 见Dictionary; Integer

3. limit: 见Dictionary; Integer

3.5.3. 返回值

result: 订单的详细(其records是array)

3.5.4. 示例

```
{
  "params": [5312054, 0, 10],
  "result": {
    "offset": 0,
    "limit": 10,
    "records": [
      {
        "time": 1533023956.499642, // timestamp
        "user": 6,
        "id": 2716635, // executed ID
        "role": 2, // int 1: maker, 2:taker
        "price": "8249",
        "amount": "0.02821226",
        "deal": "232.72293274",
        "fee": "0.00005642452",
        "deal_order_id": 5303089
      },
      {
        "time": 1533023956.499437,
        "user": 6,
        "id": 2716634,
        "role": 2,
        "price": "8249",
        "amount": "6",
        "deal": "49494",
        "fee": "0.012",
        "deal_order_id": 5303042
      }
    ]
  }
}
```

3.6. order.depth

3.6.1. 解释

查看现在的order-book(和API的order.book区分)

3.6.2. 参数

1. market: 见Dictionary; Integer
2. limit: 见Dictionary; Integer
3. interval: 不懂; String; e.g. "1" for 1 unit interval, "0" for no interval

3.6.3. 返回值

result

3.6.4. 示例

```
{
  "params": ["BTCETH", 20, "0"],
  "result": {
    "asks": [
      [
        "8000.00",
        "9.6250"
      ]
    ],
    "bids": [
      [
        "7000.00",
        "0.1000"
      ]
    ]
  ]
}
```

3.7. order.pending

3.7.1. 解释

查看还没被完成的订单(例如买价挂低了)

3.7.2. 参数

1. market: 见Dictionary; Integer
2. offset: 见Dictionary; Integer
3. limit: 见Dictionary; Integer

3.7.3. 返回值

result

3.7.4. 示例

```
{
  "params": ["BTCETH", 0, 20],
  "result": {
    "offset": 0,
    "limit": 100,
    "total": 1,
    "records": [
      {
        "id": 2,
        "ctime": 1492616173.355293,
        "mtime": 1492697636.238869,
        "market": "BTCCNY",
        "user": 2,
        "type": 1, // 1: limit order, 2: market order
        "side": 2, // 1: sell, 2: buy
        "amount": "1.0000",
        "price": "7000.00",
        "taker_fee": "0.0020",
        "maker_fee": "0.0010",
        "source": "web",
        "deal_money": "6300.0000000000",
        "deal_stock": "0.9000000000",
        "deal_fee": "0.0009000000"
      },
      {
        "id": 5348582,
        "market": "BTCETH",
        "source": "web",
        "type": 1,
        "side": 2,
        "user": 6,
        "ctime": 1534667686.270791,
        "mtime": 1534667686.270791,
        "price": "8231",
        "amount": "1",
        "taker_fee": "0.002",
        "maker_fee": "0.001",
        "left": "1",
        "deal_stock": "0",
        "deal_money": "0",
        "deal_fee": "0"
      }
    ]
  }
}
```

```
}  
}
```

3.8. order.finished

3.8.1. 解释

查看已经完成的订单

3.8.2. 参数

1. market: 见Dictionary; String
2. start_time: 起始时间(时间戳); Integer; 0代表从起始开始
3. end_time: 截至时间(时间戳); Integer; 0代表至今
4. offset: 见Dictionary; Integer
5. limit: 见Dictionary; Integer
6. side: 交易类型; Integer; 0代表所有类型, 1代表卖单, 2代表买单, 3代表取消卖, 4代表取消买

3.8.3. 返回值

result

3.8.4. 示例

```
{  
  "params": ["BTCETH", 0, 0, 0, 10, 0],  
  "result": {  
    "offset": 0,  
    "limit": 10,  
    "records": [  
      {  
        "id": 5348587,  
        "ctime": 1534828511.103844,  
        "ftime": 1534828511.103852,  
        "user": 6,  
        "market": "BTCETH",  
        "source": "web",  
        "type": 2,  
        "side": 2,  
        "price": "0",  
        "amount": "10",  
        "taker_fee": "0.02",  
      }  
    ]  
  }  
}
```

```
        "maker_fee": "0",
        "deal_stock": "0.00119189",
        "deal_money": "9.9999571",
        "deal_fee": "0.0000238378"
    },
    {
        "id": 5348583,
        "ctime": 1534731563.565035,
        "ftime": 1534731563.56504,
        "user": 6,
        "market": "BTCETH",
        "source": "web",
        "type": 2,
        "side": 2,
        "price": "0",
        "amount": "10",
        "taker_fee": "0.002",
        "maker_fee": "0",
        "deal_stock": "0.00119189",
        "deal_money": "9.9999571",
        "deal_fee": "0.00000238378"
    }
]
}
```

3.9. order.finished_detail

3.9.1. 解释

查看某一个订单的详细资讯

3.9.2. 参数

1. order_id: order ID, Integer

3.9.3. 返回值

result: 订单详细资讯

3.9.4. example:

```
{
  "params": [5312015],
```



```
"result": {  
  "id": 5312015,  
  "ctime": 1532834831.840904,  
  "ftime": 1532928509.165711,  
  "user": 15,  
  "market": "BTCETH",  
  "source": "web",  
  "type": 1,  
  "side": 1,  
  "price": "8246",  
  "amount": "1",  
  "taker_fee": "0.002",  
  "maker_fee": "0.001",  
  "deal_stock": "1",  
  "deal_money": "8246",  
  "deal_fee": "8.246"  
}
```

3.10. order.pending_detail

3.10.1. 解释

查看未完成订单的详细

3.10.2. 参数

1. market: 见Dictionary; String
2. order_id: 订单ID, Integer

3.10.3. 返回值

订单的详细资讯

3.10.4. 示例

```
{  
  "params": ["BTCETH", 5312287],  
  "result": {"error": null, "result": null, "id": 100}  
}
```

3.11. order.history

3.11.1. 解释

查看用户指定个数委托记录

3.11.2. 参数

1. market: 见Dictionary; String
2. offset: 见Dictionary; Integer
3. limit: 见Dictionary; Integer

3.11.3. 返回值

顶顶那记录列表

3.11.4. 前端实现

根据side, amount, deal_stork来判断状态

status:

- 1: 未成交 2: 部分成交 [from pending]
3: 全部成交 4: 部分取消 5: 全部取消 [from finished]

3.11.5. 示例

```
{
  "params": ["BTCETH", 0, 11],
  "result": {
    "id": 100,
    "error": null,
    "result": {
      "offset": 0,
      "limit": 11,
      "records": [
        {
          "id": 5348587,
          "ctime": 1534828511.103844,
          "ftime": 1534828511.103852,
          "user": 6,
          "market": "BTCETH",
          "source": "web",
          "type": 2,
          "side": 2,
          "status": 1, // 对应记录状态
        }
      ]
    }
  }
}
```

```
        "price": "8555",
        "amount": "10",
        "taker_fee": "0.02",
        "maker_fee": "0",
        "deal_stock": "0.00119189",
        "deal_money": "9.9999571",
        "deal_fee": "0.0000238378"
    },
    {
        "id": 5348583,
        "ctime": 1534731563.565035,
        "ftime": 1534731563.56504,
        "user": 6,
        "market": "BTCETH",
        "source": "web",
        "type": 2,
        "side": 2,
        "price": "8551",
        "amount": "10",
        "taker_fee": "0.002",
        "maker_fee": "0",
        "deal_stock": "0.00119189",
        "deal_money": "9.9999571",
        "deal_fee": "0.00000238378"
    }
]
},
}
```

4. Market API

4.1. market.deals

4.1.1. 解释

查看市场上现在的成交情况

4.1.2. 参数

1. market: 见Dictionary; String
2. limit: 见Dictionary; String
3. last_id: 最后一个订单的id, 用来限制数量; String

4.1.3. 返回值

市场上现在的成交情况

4.1.4. 示例

```
{
  "params": ["BTCETH", 10, 1],
  "result": [
    {
      "id": 5,
      "time": 1492697636.238869,
      "type": "sell",
      "amount": "0.1000",
      "price": "7000.00"
    },
    {
      "id": 4,
      "time": 1492697467.1411841,
      "type": "sell",
      "amount": "0.1000",
      "price": "7000.00"
    }
  ]
}
```

4.2. market.kline

4.2.1. 解释

查询k线的信息(高开低收成交量)

4.2.2. 参数

1. market: 见Dictionary; String
2. start: 起始时间(时间戳); Integer
3. end: 截止时间(时间戳); Integer
4. interval: interval, Integer

4.2.3. 返回值

k线的信息

4.2.4. 示例

```
{
  "params": ["BTCUSDT", 1492358400, 1492358400, 2],
  "result": [
    [
      1492358400, // time
      "7000.00", // open
      "8000.0", // close
      "8100.00", // highest
      "6800.00", // lowest
      "1000.00", // volume
      "123456.78", // amount
      "BTCCNY" // market name
    ]
  ]
}
```

4.3. market.status

4.3.1. 解释

查看过去一段时间的市场统计

4.3.2. 参数

1. market: 见Dictionary; String
2. period: 往前看的时间长度; Integer; e.g. 86400 for last 24 hours

4.3.3. 返回值

result

4.3.4. 示例

```
{
  "params": ["BTCETH", 86400],
  "result": {
    "period": 86400,
    "last": "7000.00",
    "open": "0",
    "close": "0",
  }
}
```

```
    "high": "0",  
    "low": "0",  
    "volume": "0"  
  }  
}
```

4.4. market.list

4.4.1. 解释

查询各个市场的相关资讯

4.4.2. 参数

[]

4.4.3. 返回值

各个市场的相关资讯

4.4.4. 示例

```
{  
  "params": [],  
  "result":  
  [  
    {  
      "name": "BTCETH",  
      "stock": "BTC",  
      "money": "ETH",  
      "fee_prec": 4,  
      "stock_prec": 8,  
      "money_prec": 8,  
      "min_amount": "0.001"  
    },  
    {  
      "name": "BHBETH",  
      "stock": "BHB",  
      "money": "ETH",  
      "fee_prec": 4,  
      "stock_prec": 8,  
      "money_prec": 8,  
      "min_amount": "0.001"  
    }  
  ],  
}
```

```
]
}
```

4.5. market.user_deals

4.5.1. 解释

查询用户的操作记录

4.5.2. 参数

1. market: 见Dictionary; String
2. offset: 见Dictionary; Integer
3. limit: 见Dictionary; Integer

4.5.3. 返回值

result

4.5.4. 示例

```
{
  "params": ["BTCETH", 0, 10],
  "result": {
    "offset": 0,
    "limit": 10,
    "records": [
      {
        "time": 1531387835.317415, // timestamp
        "user": 6,
        "id": 61138, // Executed ID
        "side": 2, // side, 1: sell, 2: buy
        "role": 2, // role, 1: Maker, 2: Taker
        "price": "8473",
        "amount": "0.13874996",
        "deal": "1175.62841108",
        "fee": "0.00027749992",
        "deal_order_id": 29216, // Counterpart Transaction ID
        "market": "BTCETH"
      },
      {
        "time": 1531387835.317189,
        "user": 6,
        "id": 61137,
```

```
        "side":2,  
        "role":2,  
        "price":"8473",  
        "amount":"3",  
        "deal":"25419",  
        "fee":"0.006",  
        "deal_order_id":28528,  
        "market":"BTCETH"  
    }  
]  
}
```

```
}
```

4.6. market.last

4.6.1. 解释

查询市场价格

4.6.2. 参数

1. market: 见Dictionary; String

4.6.3. 返回值

result: "price"

4.6.4. 示例

```
{  
  "params": ["BTCUSDT"],  
  "result": "7000"  
}
```