Thsrobot API文档

基础路径： http://localhost:80/api/v1

买入

POST /stock/buy
请求体

```
{
  "code": "002188",
  "price": "1.22",
  "volume": 100,
  "acceptRisk": false
}
```

字段	类型	必填	描述
code	string	是	股票代码，必须是 6位数字的字符串，如： "002188"
price	float	是	价格，最多支持3位小数，如： 1.22
volume	int	是	数量，必须是100倍数，如：1200
acceptRisk	boolean	否	是否接受风险，如果true，表示提示风险也委托交易，默认 false

响应

```
{
    "message": "提交失败：[120022][该功能禁止在目前系统状态下运行]\r。",
    "status": 0
}
```

买入（异步）

POST /stock/sync/buy
请求体

```
{
  "code": "002188",
  "price": "1.22",
  "volume": 100,
  "acceptRisk": false
}
```

字段	类型	必填	描述
code	string	是	股票代码，必须是 6位数字的字符串，如： "002188"
price	float	是	价格，最多支持3位小数，如： 1.22
volume	int	是	数量，必须是100倍数，如：1200
acceptRisk	boolean	否	是否接受风险，如果true，表示提示风险也委托交易，默认 false

响应

```
{
    "message": "提交成功",
    "status": 0
}
```

卖出

POST /stock/sell
请求体

```
{
  "code": "002188",
  "price": "1.22",
  "volume": 100,
  "acceptRisk": false
}
```

字段	类型	必填	描述
code	string	是	股票代码，必须是 6位数字的字符串，如： "002188"
price	float	是	价格，最多支持3位小数，如： 1.22
volume	int	是	数量，必须是100倍数，如：1200
acceptRisk	boolean	否	是否接受风险，如果true，表示提示风险也委托交易，默认 false

响应

```
{
    "message": "提交失败：[120022][该功能禁止在目前系统状态下运行]\r。",
    "status": 0
}
```

卖出（异步）

POST /stock/sync/sell
请求体

```
{
  "code": "002188",
  "price": "1.22",
  "volume": 100,
  "acceptRisk": false
}
```

字段	类型	必填	描述
code	string	是	股票代码，必须是 6位数字的字符串，如： "002188"
price	float	是	价格，最多支持3位小数，如： 1.22
volume	int	是	数量，必须是100倍数，如：1200
acceptRisk	boolean	否	是否接受风险，如果true，表示提示风险也委托交易，默认 false

响应

```
{
    "message": "提交成功",
    "status": 0
}
```

撤单

POST /stock/cancel
请求体

```
{
  "cancelType": 0
}
```

字段	类型	说明
cancelType	int	0=全部取消 1=取消买入 2=取消卖出

响应

```
{
    "status": 0
}
```

撤单（异步）

POST /stock/sync/cancel
请求体

```
{
  "cancelType": 0
}
```

字段	类型	说明
cancelType	int	0=全部取消 1=取消买入 2=取消卖出

响应

```
{
  "message": "提交成功",
  "status": 0
}
```

资金信息

GET /stock/assets
响应

```
{
  "data": {
    "资金余额": "434.90",
    "冻结金额": "",
    "可用金额": "432.90",
    "可取金额": "432.90",
    "股票市值": "86359.00",
    "总资产": "86793.90",
```

```
    "持仓盈亏": "5526.97",
    "当日盈亏": "",
    "当日盈亏比": ""
  },
  "status": 0
}
```


委托

GET /stock/order
响应


```
{
    "data": [
        {
            "交易市场": "深圳A股",
            "合同编号": "13564",
            "备注": ":已成",
            "委托价格": "4.370",
            "委托数量": "1600",
            "委托时间": "13:31:30",
            "成交均价": "4.370",
            "成交数量": "1600",
            "撤消数量": "0",
            "操作": "买入",
            "证券代码": "002494",
            "证券名称": "华斯股份"
        }
    ],
    "status": 0
}
```


持仓

GET /stock/position
响应


```
{
    "data": [
        {
            "交易市场": "上海A股",
            "仓位占比(%)": "0.00",
            "冻结数量": "0",
            "可用余额": "10",
            "市价": "100.000",
            "市值": "1000.000",
            "当日买入": "0",
            "当日卖出": "0",
            "当日盈亏": "0.00",
            "当日盈亏比(%)": "0.00",
```

```
            "成本价": "100.000",
            "持股天数": "--",
            "盈亏": "0.000",
            "盈亏比(%)": "0.00",
            "股票余额": "10",
            "证券代码": "718131",
            "证券名称": "皓元发债"
        }
    ],
    "status": 0
}
```


成交

GET /stock/trade
响应


```
{
    "data": [
        {
            "合同编号": "13564",
            "委托时间": "13:31:30",
            "成交均价": "4.370",
            "成交数量": "1600",
            "成交时间": "13:54:15",
            "成交编号": "0104000064220570",
            "成交金额": "6992.000",
            "操作": "买入",
            "证券代码": "002494",
            "证券名称": "华斯股份"
        }
    ],
    "status": 0
}
```

鉴权

请求头

通过以下方法计算出签名，放在请求头 Authorization 中

```
# coding:utf-8

import hashlib
import hmac
import json
import time
import uuid
from urllib.parse import urlparse, parse_qs, urlencode

import requests
from requests.auth import AuthBase

your_server_addr ="http://localhost:8080/api/v1/stock/"
secretId ="bWYgDxe1ZBiQK4Tt4XCP6vYCWY3QuYxm"
secretKey ="bWYgDxe1ZBiQK4Tt4XCP6vYCWY3QuYxm"


# 认证签名
class SignAuth(AuthBase):
    def __init__(self, secret_id:str = secretKey, secret_key:str = secretKey):
        self.secret_id = secret_id
        self.secret_key = secret_key

    def __call__(self, r):
        # 获取当前时间戳和nonce
        timestamp = str(int(time.time()))
        nonce = str(uuid.uuid4())
        body = r.body or b""

        parsed_url = urlparse(r.url)
        query_params = parse_qs(parsed_url.query)  # 获取查询参数字典
        # 对查询参数进行排序
        sorted_query_params = dict(sorted(query_params.items()))
        # 将排序后的查询参数重新编码为字符串
        sorted_params_str = urlencode(sorted_query_params, doseq=True)

        # 构造待签名字符串
        sign_data = [
            r.method,
            r.path_url.split("?")[0],
            sorted_params_str,
            timestamp,
            nonce,
```

```
            body.decode('utf-8') if isinstance(body, bytes) else body
        ]

        sign_data = '\n'.join(sign_data)

        print("签名数据\n",sign_data)


        # 使用HMAC算法和SHA256哈希函数创建签名
        signature = hmac.new(self.secret_key.encode('utf-8'), sign_data.encode('utf-8'),
hashlib.sha256)

        # 将签名转换为Base64编码的字符串
        signature = signature.digest().hex()

        # 添加必要的认证头
        authorization = f"hmac id=\"{self.secret_id}\", ts=\"{timestamp}\", nonce=\"
{nonce}\", sig=\"{signature}\""

        print("Authorization", authorization)

        r.headers['Authorization'] = authorization


        return r




#股票买入方法
def buy_stock(stock_code,price,vol):
    start_time = time.time()
    print('开始买入:'+stock_code+' 价格: '+str(price)+' 数量: '+str(vol))
    result=requests.post(your_server_addr+"buy", json={
        "code": stock_code,
        "price": price,
        "volume": vol
    }, auth=SignAuth())
    print(result.json())
    end_time = time.time()
    elapsed_time = end_time - start_time
    print(f"买入执行耗时: {elapsed_time} 秒")

#股票卖出方法
def sell_stock(stock_code,price,vol):
    start_time = time.time()
    print('开始卖出:'+stock_code+' 价格: '+str(price)+' 数量: '+str(vol))
    stock_code=stock_code[:6]
    result=requests.post(your_server_addr+"sell", json={
        "code": stock_code,
        "price": price,
        "volume": vol
```

```
    }, auth=SignAuth())
    print(result.json)
    end_time = time.time()
    elapsed_time = end_time - start_time
    print(f"卖出执行耗时：{elapsed_time} 秒")

#委撤撤单方法
def cancel_stock(cancelType):
    start_time = time.time()
    print('开始全部撤单')
    result=requests.post(your_server_addr+"cancel", json={
        "cancelType": cancelType
    }, auth=SignAuth())
    print(result.json)
    end_time = time.time()
    elapsed_time = end_time - start_time
    print(f"卖出执行耗时：{elapsed_time} 秒")

#获取账户资金方法
def get_account():
    result=requests.get(your_server_addr+"funding", auth=SignAuth()).text
    data_dict=json.loads(result)
    return data_dict['data']

#获取委托信息方法
def get_order():
    result=requests.get(your_server_addr+"order", auth=SignAuth()).text
    data_dict=json.loads(result)
    if 'data' in data_dict.keys():
        return data_dict['data']
    else:
        return

#获取持仓方法
def get_position():
    result=requests.get(your_server_addr+"position", auth=SignAuth()).text
    data_dict=json.loads(result)
    return data_dict['data']


############################以下为方法的调用示例##################################
if 1:
    #获取账号账户信息
    print('开始获取账户信息')
    account1=get_account()
    print(account1)
    account_total1=float(account1['total'])
    print("账号总资金："+str(account_total1))

if 1:
    #获取持仓信息
    print('开始获取持仓信息')
```

```
    holdings_dict={}
    holdings = get_position()
    if holdings and len(holdings)>0:
        for i in holdings:
            if int(i['可用余额'])>0:
                stock_code=i['证券代码']
                holdings_dict[stock_code]=int(i['可用余额'])
    print('账号持仓')
    print(holdings_dict)

if 1:
    #股票买入,注：此处买入的价格，必须为现价的+-2%以内（不然会被交易所废单 ），买入数量必须是100的整数倍
    buy_stock('002936',2.01,100)

if 1:
    #获取委托信息
    print('开始获取委托信息')
    print(get_order())


if __name__ == '__main__':
    print(get_order())
    print(get_account() )
    print(get_position() )
    print(get_position() )
```