

# 交易系统交易接口

## 1 应答数据说明

### 1.1 应答数据约定

- 所有接口的响应都是 JSON 格式。
- 响应中如有数组，数组元素以时间升序排列，越早的数据越提前。
- 本篇列出接口的baseurl: <https://api.xxx.com>

### 1.2 HTTP 返回代码

- HTTP 4xx 错误码用于指示错误的请求内容、行为、格式。问题在于请求者。
- HTTP 403 错误码表示违反WAF限制(Web应用程序防火墙)。
- HTTP 429 错误码表示警告访问频次超限，即将被封IP。
- HTTP 418 表示收到429后继续访问，于是被封了。
- HTTP 5xx 错误码用于指示交易系统服务侧的问题。使用接口 /api/v1 时, HTTP 504 表示API服务端已经向业务核心提交了请求但未能获取响应，特别需要注意的是504代码不代表请求失败，而是未知。很可能已经得到了执行，也有可能执行失败，需要做进一步确认。

### 1.3 接口的基本信息

- GET 方法的接口, 参数必须在 query string 中发送, content-type 为 application/x-www-form-urlencoded。
- POST, GET 的接口,参数可以在 content-type 为 application/x-www-form-urlencoded 的 query string 中发送, 也可以在 request body 中发送 (此时传输类型: application/json)。如果你喜欢, 也可以混合这两种方式发送参数,但建议分开。
- 对参数的顺序不做要求。
- 但如果同一个参数名在query string和request body中都有, query string中的会被优先采用。
- 对于业务返回数据。返回成功时, success为true.如下:

```
{
  "retCode": "0000",
  "retMsg": "ok"
  "success": true,
  "body": {},
  "time": "2020-03-11T03:00:13.496Z"
}
```

返回失败时, 如下:

```
```shell
```

```
{
  "retCode": "1000",
  "retMsg": "xx参数不能为空！ ",
  "success": false,
  "body": null,
  "time": "2020-03-11T03:00:14.496Z"
}
```

## 2 交易协议接口

### 2.1 用户登录

用于App，Web，PC端登录，这里通过账户代码登录。

#### HTTP 请求

- POST `/v1/account/login`

```
curl -X POST "https://dev-kg1.jt00000.com/kg1-trade-service/v1/account/login" -H
"accept: */*" -H "Content-Type: application/json" -d "{ \"accountCode\":
\"SOSLMG01\", \"channelType\": \"INTERNET\", \"ipAddress\": \"23.3.3.3\",
\"password\": \"a12345\"}"
```

输入参数：

参数名	类型	是否必需	描述
channelType	String	是	渠道类型
accountCode	String	是	账户代码
password	String	是	登录密码
ipAddress	String	否	登录IP

输出参数：

字段名	类型	描述
accountCode	String	账户代码
accountName	String	账户名称
accountType	String	账户类型
alertChangePwd	Boolean	如果为True，则提示用户密码即将过期，否则，忽略
forceChangePwd	Boolean	如果为True，则强制用户需要修改密码（跳转到密码修改页面），否则，忽略
lastLoginTime	String	最后一次登录时间，格式: yyyy-MM-dd HH:mm:ss
lastLoginIP	String	最后一次登录IP

## 2.2 资产查询

用户查询账户下资产信息。

### HTTP 请求

- GET `/v1/account/accounts/balance`

```
curl -X GET "https://dev-kg1.jt00000.com/kg1-trade-service/v1/account/accounts/balance?accountCode=SOSLMG01" -H "accept: */*"
```

输入参数:

参数名	类型	是否必需	描述
accountCode	String	是	账户代码

输出参数:

字段名	类型	说明
accountCode	String	账户代码
baseCcy	String	基础币种
cash	String	总资金余额
marketValue	String	总市值
frozenCash	String	总冻结金额
withdrawableCash	String	总可提金额
buyPower	String	购买力
- ccy	String	币种代码
- cash	String	资金余额
- marketValue	String	市值
- frozenCash	String	冻结金额
- withdrawableCash	String	可提金额
- buyPower	String	购买力

## 2.3 持仓查询

用户查询账户的持仓信息。

### HTTP 请求

- GET `/v1/account/accounts/position`

```
curl -X GET "https://dev-kg1.jt00000.com/kg1-trade-  
service/v1/account/accounts/position?accountCode=SOSLMG01" -H "accept: */*"
```

输入参数:

参数名	类型	是否必需	描述
accountCode	String	是	账户代码

输出参数:

字段名	类型	说明
exchangeCode	String	交易所代码
productCode	String	产品（股票）代码
productName	String	产品（股票）名称
qty	String	数量
ccy	String	产品计价币种
avgCost	String	均价(成本价)
rate	String	汇率
pl	String	今日盈亏
price	String	当前买入价（实时）
marketValue	String	当前市值（实时）
availSellQty	String	可卖出数量
osSellQty	String	已冻结数量（交易冻结）
availQty	String	可用数量（T+1）
unclearedQty	String	未到账数量
todayBuy	String	今日买入数量
todaySell	String	今日卖出数量

## 2.4 最大买交易数量查询

查询用户对产品最大买入量

### HTTP 请求

GET /v1/order/orders/position/maxBuyQty

```
curl -X GET "https://dev-kg1.jt00000.com/kg1-trade-  
service/v1/order/orders/position/maxBuyQty?  
accountCode=SOSLMG01&channelType=I&exchangeCode=HKEX&price=346&productCode=00700  
" -H "accept: */*"
```

输入参数:

参数名	类型	是否必需	描述
accountCode	String	是	账户代码
exchangeCode	String	是	交易所代码
productCode	String	是	股票代码
price	String	是	股票价格
channelType	String	是	渠道类型(app请填写'I')

输出参数:

字段名	类型	说明
totalCashAvail	String	账户总可用现金(此为账户下各个币种划算的总金额)
cashQty	String	账户总可买产品数量 (产品买卖设置的币种)
ccyCashAvail	String	账户可用现金 (产品买卖设置的币种)
ccyQty	String	账户可买产品数量 (产品买卖设置的币种)

## 2.5 订单委托

用户下单

### HTTP 请求

- POST `/v1/order/orders/place`

```
curl -X POST "https://dev-kg1.jt00000.com/kg1-trade-  
service/v1/order/orders/place" -H "accept: */*" -H "Content-Type:  
application/json" -d "{ \"accountCode\": \"SOSLMG01\", \"bsFlag\": \"B\",  
\"channelType\": \"I\", \"exchangeCode\": \"HKEX\", \"orderType\": \"L\",  
\"password\": \"a12345\", \"price\": \"1000\", \"productCode\": \"00700\",  
\"qty\": \"100\", \"remark\": \"123\"}"
```

输入参数:

参数名	类型	是否必需	描述
channelType	String	是	渠道类型
exchangeCode	String	是	交易所代码
accountCode	String	是	账户代码
productCode	String	是	股票代码
password	String	否	账户密码
price	String	是	价格
qty	String	是	数量
bsFlag	String	是	买卖方向
orderType	String	是	订单类型
remark	String	否	备注

输出参数：

字段名	类型	说明
orderId	String	订单ID
orderStatus	String	订单状态

## 2.6 订单修改

用户订单修改

### HTTP 请求

- POST `/v1/order/orders/update`

```
curl -X POST "https://dev-kg1.jt00000.com/kg1-trade-service/v1/order/orders/update" -H "accept: */*" -H "Content-Type: application/json" -d '{"accountCode": "SOSLMG01", "exchangeCode": "HKEX", "channelType": "I", "orderId": "172153", "password": "a12345", "price": "1001", "productCode": "00700", "qty": "100"}'
```

输入参数：

参数名	类型	是否必需	描述
channelType	String	是	渠道类型
accountCode	String	是	账户代码
password	String	否	账户密码
price	String	是	股票价格
qty	String	是	数量
orderId	String	是	订单ID

输出参数:

无.

## 2.7 订单撤销

用户撤单，这里只是提交撤单请求，并不保证撤单成功

### HTTP 请求

- POST `/v1/order/orders/cancelOrder`

```
curl -X POST "https://dev-kg1.jt00000.com/kg1-trade-service/v1/order/orders/cancel" -H "accept: */*" -H "Content-Type: application/json" -d '{"accountCode": "SOSLMG01", "channelType": "I", "orderId": "172153", "password": "a12345"}'
```

输入参数:

参数名	类型	是否必需	描述
channelType	String	是	渠道类型
accountCode	String	是	账户代码
password	String	否	账户密码
orderId	String	是	订单ID

输出参数:

无.

## 2.8 单个订单查询

用户根据指定订单ID来查询订单详细信息

### HTTP 请求

GET `/v1/order/orders/singleOrder`

```
curl -X GET "https://dev-kg1.jt00000.com/kg1-trade-  
service/v1/order/orders/singleOrder?accountCode=SOSLMG01&orderID=172243" -H  
"accept: */*"
```

输入参数：

参数名	类型	是否必需	描述
accountCode	String	是	账户代码
orderID	String	是	订单ID

输出参数：

字段名	类型	说明
channelType	String	渠道类型
exchangeCode	String	交易所代码
productCode	String	产品代码
productName	String	产品名称
accountCode	String	账户代码
bsFlag	String	买卖方向
orderID	String	订单ID
orderType	String	订单类型
orderStatus	String	订单状态
rejectReason	String	拒绝原因
price	String	价格
qty	String	数量
execQty	String	成交数量
execPrice	String	成交均价
ccy	String	货币代码
remark	String	备注
createTime	String	创建时间，格式: yyyy-MM-dd HH:mm:ss
updateTime	String	最后更新时间，格式: yyyy-MM-dd HH:mm:ss

## 2.9 多订单查询

根据日期范围查询多订单数据

### HTTP 请求



- GET /v1/order/orders

```
curl -X GET "https://dev-kg1.jt00000.com/kg1-trade-service/v1/order/orders?accountCode=SOSLMG01" -H "accept: */*"
```

输入参数:

参数名	类型	是否必需	描述
accountCode	String	是	账户代码
channelType	String	否	渠道类型
orderStatus	String	否	订单状态
beginDate	String	否, 默认为系统日期, yyyy-MM-dd	起始日期
endDate	String	否, 默认为系统日期, yyyy-MM-dd	结束日期

输出参数:

字段名	类型	说明
channelType	String	渠道类型
exchangeCode	String	交易所代码
productCode	String	产品代码
productName	String	产品名称
accountCode	String	账户代码
bsFlag	String	买卖方向
orderId	String	订单ID
orderType	String	订单类型
orderStatus	String	订单状态
rejectReason	String	拒绝原因
price	String	价格
qty	String	数量
execQty	String	成交数量
exePrice	String	成交均价
execAmt	String	成交金额
ccy	String	币种代码
remark	String	备注
createTime	String	创建时间，格式: yyyy-MM-dd HH:mm:ss
updateTime	String	最后更新时间，格式: yyyy-MM-dd HH:mm:ss

## 2.10 未执行订单查询

用户查询可撤单的订单列表

### HTTP 请求

- GET /v1/order/openOrders

```
curl -X GET "https://dev-kg1.jt00000.com/kg1-trade-service/v1/order/openOrders?accountCode=SOSLMG01" -H "accept: */*"
```

输入参数：

参数名	类型	是否必需	描述
accountCode	String	是	账户代码
channelType	String	否	渠道类型

输出参数：

字段名	类型	说明
channelType	String	渠道类型
exchangeCode	String	交易所代码
productCode	String	产品代码
productName	String	产品名称
accountCode	String	账户代码
bsFlag	String	买卖方向
orderId	String	订单ID
orderType	String	订单类型
orderStatus	String	订单状态
rejectReason	String	拒绝原因
price	String	价格
qty	String	数量
execQty	String	成交数量
execPrice	String	成交均价
execAmt	String	成交金额
ccy	String	币种代码
remark	String	备注
createTime	String	创建时间，格式: yyyy-MM-dd HH:mm:ss
updateTime	String	最后更新时间，格式: yyyy-MM-dd HH:mm:ss

## 2.11 成交、资产流水查询

用户查询资产流水（因Ayers系统无分页，故暂时不实现）

### HTTP 请求

GET `/v1/order/myTrades`

输入参数：

参数名	类型	是否必需	描述
accountCode	String	是	账户代码
beginDate	String	否，默认为系统日期，yyyy-MM-dd	起始日期
endDate	String	否，默认为系统日期，yyyy-MM-dd	结束日期

输出参数：

字段名	类型	说明
accountCode	String	账户代码
tranType	String	交易类型
tranDate	String	交易日期
ccy	String	币种代码
exchangeCode	String	交易所代码
productCode	String	产品代码
price	String	交易（成交）价格
qty	String	交易（成交）数量
amt	String	交易（成交）金额
tranDetail	String	交易详细类型
remark	String	备注

## 2.12 用户登出

用于App，Web，PC端登出，这里通过账户代码登出。

### HTTP 请求

- POST `/v1/account/logout`

```
curl -X POST "https://dev-kg1.jt00000.com/kg1-trade-service/v1/account/logout" -H "accept: */*" -H "Content-Type: application/json" -d '{"accountCode": "\SOSLMG01\"}'
```

输入参数：

参数名	类型	是否必需	描述
accountCode	String	是	账户代码
password	String	是	登录密码
ipAddress	String	否	登录IP

## 3 Websocket 订阅接口

## 3.1 协议简介

1. 所有协议均支持http (POST方式) 和websocket请求, 除非有特别指出。
2. websocket连接请发送ping请求, 维持长连接, 因此可以不用发本程序提供的心跳协议。但可能以后会支持业务心跳。
3. 协议本身会生成一个UUID的SessionID, 故暂时不需要显示自己维护。
4. reqtype, reqid, session, data为必填字段, 并区分大小写, 其中reqtype为功能号 (下文有列表说明), reqid为客户端自己维持, 客户端传什么值, 本程序返回该值, 为正整型, 否则强制返回0, session为用户登录后, 程序返回给客户端的一串字符串, 不登录则填空, 上文已做说明, data字段为各个请求特定的数据。
5. 本程序请求包和返回包均为json格式的明文字符串, 组装json格式的请求时, 请注意各个键值对之间不要带空格或者其他字符, 不然程序可能不会解析该请求。
6. 请求包和返回包的字符编码均为utf-8。
7. 同时请求多个商品的接口中, 商品个数不能超过1000, 否则会收到参数错误的回包。
8. 客户端请求包长度不能超过100k, 服务器回包的大小不能超过10M, 否则服务器将会直接断开连接。

## 3.2 客户端实现

具体采用 Socket.IO 实现, 其 JavaScript 实现核心如下:

```
<script>
    var socket = io.connect('http://localhost:9098');

    socket.on('connect', function() {
        output('<span class="connect-msg">Client has connected to the
server!</span>');
    });
    //执行回调
    socket.on('trade', function(data) {
        output('<span class="username-msg">' + data.reqtype + ':</span> ' +
data.msg);
    });

    socket.on('disconnect', function() {
        output('<span class="disconnect-msg">The client has disconnected!
</span>');
    });

    function sendDisconnect() {
        socket.disconnect();
    }

    function sendMessage() {
        var jsonObject = {
            session: '1112-222e-dd33-dd33',
            reqtype: 10,
            reqid: 12039,
            data: {
                productCode: '600446',
                exchangeCode: 'HKEX'
            }
        };
    }
</script>
```

```

    }
    };
    //发送交易类事件请求
    socket.emit('trade', jsonObject);
  }
</script>

```

## 3.3 功能列表

### 3.3.1 客户认证

#### 3.3.1.1 请求

```

{
  "reqtype":10,
  "reqid":1,
  "session": "",
  "data": {
    "accountCode": "SOSLMG01",
    "password": "a12345"
    "ipAddress": 198.22.32.2
  }
}

```

#### 3.3.1.2 应答

```

{
  "status":0,
  "msg": "",
  "servertime": "2020-4-7 07:58:58",
  "reqtype":10,
  "reqid":1,
  "data":{
    "accountCode": "SOSLMG01",
    "accountName": "xx",
    "accountType": "xx",
    "alertChangePwd": "true",
    "forceChangePwd": "true",
    "lastLoginTime": "2020-4-7 15:01:56",
    "lastLoginIP": "198.22.32.2"
  }
}

```

### 3.3.2 订阅订单、成交变动

#### 3.3.2.1 请求

```
{
  "reqtype": 200,
  "reqid": 1,
  "session": "",
  "data": [{
    "exchangeCode": 1000,
    "productCode": "000001"
  }, {
    "exchangeCode": 1,
    "productCode": "600000"
  }]
}
```

### 3.3.2.2 应答

```
{
  "status": 0,
  "msg": "",
  "servertime": "2020-4-7 15:01:56",
  "reqtype": 150,
  "reqid": 1,
  "data": {
    "ret": "1"
  }
}
```

注意：本协议只支持WebSocket请求，收到应答后，会持续收到行情推送。

这里要注意应答的 reqtype 为150，而推送的 reqtype 为250。

### 3.3.2.3 推送

```
{
  "status": 0,
  "msg": "",
  "servertime": "2020-4-7 15:11:26",
  "reqtype": 250,
  "reqid": 1,
  "data": {
    // -- 具体字段如下文
  }
}
```

这里注意，这里推送包含3个事件：

```
CREATION (101),      // 事件编号 101 表示订单创建
TRADE (102),         // 事件编号 102 表示订单成交
CANCELLATION(103);  // 事件编号 103 表示订单取消（撤单）
```

- 订单创建时，data字段如下：

```
{
  "eventType": 101,
  "exchangeCode": "HKEX",
  "productCode": "00700",
  "bsFlag": "B",
```

```
"orderId":"172392",
"clientOrderId":"","
"orderType":"L" ,
"orderStatus":"Q",
"price":"1000",
"qty":"100" ,
"execQty":"0",
"execPrice":"0.00",
"execAmt":"0",
"ccy":"HKD",
"createTime":"2020-4-7 15:11:23"
}
```

- 订单成交时，data字段如下：

```
{
  "eventType":102,
  "exchangeCode":"HKEX",
  "productCode":"00700",
  "tradeID":"20200407:31800389:101",
  "orderId":"172392",
  "clientOrderId":"","
  "orderType":"L" ,
  "orderStatus":"PEX",
  "execQty":"10",
  "execPrice":"101.00",
  "execAmt":"90",
  "tradeTime":"2020-4-7 15:11:23"
}
```

- 订单取消时，data字段如下：

```
{
  "eventType":103,
  "exchangeCode":"HKEX",
  "productCode":"00700",
  "orderId":"172392",
  "clientOrderId":"","
  "orderStatus":"CAN", //全撤单
  "remainQty":"10", //未成交数量
  "lastUpdateTime":"2020-4-7 15:11:23"
}
```

注意，订单状态有如下值：

NEW - 订单创建  
PEX - 部分成交  
FEX - 全成交  
CAN - 已撤单