|  |  |
| --- | --- |
| 产品名称 | 密级 |
| 手机行情服务器 | 保密 |
| 产品版本 |  |
| V3.0 |

# 手机行情服务器与客户端通讯协议

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 | 王雄 | 日期 | 2017-5-02 |
| 评审 |  | 日期 |  |
| 批准 |  | 日期 |  |
| 签发 |  | 日期 |  |
| 修改 |  | 日期 |  |

## 目录

[手机行情服务器与客户端通讯协议 1](#_Toc983)

[目录 2](#_Toc28814)

[一、协议说明 4](#_Toc3732)

[二、websocket参数说明 4](#_Toc23668)

[三、商品行情相关字段说明 5](#_Toc15792)

[四、功能列表 6](#_Toc12627)

[五、接口说明 8](#_Toc26770)

[5.1 心跳(1) 8](#_Toc17360)

[5.2 请求版本号(4) 8](#_Toc18645)

[5.3用户认证（现在认证已经在java那边实现，这个弃用） 9](#_Toc26742)

[5.4退出登录(3) 10](#_Toc25779)

[5.5 获取市场信息(50) 11](#_Toc26894)

[5.6 获取商品详情(51) 13](#_Toc4474)

[5.7 获取商品列表(52) 16](#_Toc21929)

[5.8 请求商品排行(53) 17](#_Toc30449)

[5.9 用户输入代码(54) 21](#_Toc5899)

[5.10 指定商品排序(55) 22](#_Toc663)

[5.11 请求子市场列表(56) 24](#_Toc5463)

[5.12 请求合约信息(57) 25](#_Toc29214)

[5.13 附加条件查询商品(58) 26](#_Toc7464)

[5.14 请求交易时间(59) 27](#_Toc5785)

[5.15 获取期权列表(60) 29](#_Toc16007)

[5.16 获取期权商品列表(61) 30](#_Toc15920)

[5.17请求公共板块配置(63) 31](#_Toc31822)

[5.18 请求公共板块列表(64) 32](#_Toc24663)

[5.19 获得财务数据(64) 33](#_Toc6146)

[5.20 获得上涨下跌家数（市场）(65) 36](#_Toc18652)

[5.21 请求领涨股(66) 36](#_Toc25903)

[5.22 获得多个商品的财务数据(67) 41](#_Toc8263)

[5.23 获得全市场列表(68) 42](#_Toc10424)

[5.24 获得板块中商品的上涨下跌家数(69) 43](#_Toc25702)

[5.25 差价表(70) 44](#_Toc27598)

[5.26 AH与ADR配置(71) 45](#_Toc14834)

[5.27 碎股(72) 46](#_Toc10080)

[5.28查询具体商品的关联商品(74) 47](#_Toc2231)

[5.29 经纪人队列(76) 48](#_Toc5185)

[5.30 按条件筛选涡轮牛熊商品(77) 49](#_Toc7589)

[5.31成交统计(78) 52](#_Toc3423)

[5.32期权查询(79) 53](#_Toc10928)

[5.33券商进出(80) 55](#_Toc7349)

[5.34商品其他信息查询(82) 57](#_Toc31338)

[5.35全部存在关联商品的商品(83) 59](#_Toc20715)

[5.36 获取K线(150) 60](#_Toc7614)

[5.37 获取分笔数据(151) 62](#_Toc10420)

[5.38 获取分时数据(152) 63](#_Toc19510)

[5.39 获取商品行情(153) 65](#_Toc2672)

[5.40 使用合约号获取商品行情(154) 67](#_Toc13230)

[5.41 五日分时(158) 68](#_Toc12707)

[5.42 通过合约号获取行情,需要完全市场和合约号(155) 70](#_Toc6756)

[5.43 请求推送(200) 70](#_Toc2354)

[5.44 删除推送(201) 72](#_Toc3779)

[5.45 同步推送(202) 73](#_Toc24719)

[5.46 上传用户私有数据(400) 74](#_Toc31330)

[5.47 上传机构额外数据(401) 75](#_Toc6965)

[5.48 下载用户私有数据(402) 76](#_Toc23046)

[5.49 下载机构额外数据(403) 77](#_Toc13609)

[5.50 获得上涨下跌家数（板块）(69) 78](#_Toc10923)

[5.51 批量查询商品市场(84) 78](#_Toc3162)

[5.52 商品模糊匹配（市场和板块）(85) 79](#_Toc23518)

[5.53 经济人信息查询(86) 81](#_Toc7694)

[5.54 正股的涡轮牛熊期权查询(87) 82](#_Toc23029)

[5.55 霸才精选查询(88) 84](#_Toc8394)

## 一、协议说明

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。

3.本程序支持不登录获取所有数据，不登录请求数据，则将所有请求中的session字段值填""；

4.reqtype,reqid,session,data为必填字段，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，否则手机服务器将会直接断开连接。

9.http和websocket均支持同时50000个连接。

特别注意：C++区分大小写，json字段传输是，字段名统一小写。

## 二、websocket参数说明

typedef struct \_WebHead

{

u8 iOpcode : 4;

u8 iRsv : 3;

u8 bFin : 1;

u8 iPayloadLen : 7;

u8 bMask : 1;

}T\_WebHead;

enum E\_WebProType

{

EWPTFrames = 0, // 连续的帧

EWPTText = 1, // 文本

EWPTBinanry = 2, // 二进制数据

EWPTClose = 8, // 断开连接

EWPTPing = 9, // PING

EWPTPong = 10, // PONG

};

T\_WebHead stHead;

stHead.bFin = 1;

stHead.bMask = 0;

stHead.iOpcode = EWPTBinanry;

stHead.iRsv = 0;

## 三、商品行情相关字段说明

now：现价

open：开盘价

high：最高价

low：最低价

avg：均价

settle：结算价

lastclose：昨收价

lastsettle：昨结价

curvolume：现手

volume：总手

lasthold：昨持仓

curhold：最近一笔的增仓量

hold：总持仓量

volumerate：量比

buyvolume：内盘

sellvolume：外盘

buyprice0：买一价同理买二价到买五价（buyprice1--buyprice4）

buyvol0：买一量同理买二量到买五量（buyvol1--buyvol4）

sellprice0：卖一价同理卖二价到卖五价（sellprice1--sellprice4）

sellvol0：卖一量同理卖二量到卖五量（sellvol1--sellvol4）

traderate：换手率

moneyin：资金流向

riserate：涨速(最新价和上个5分钟收盘价的比率)

sellamount:外盘成交额

tradecount:成交笔数

riselimit:涨停价

falllimit:跌停价

// 商品类型

enum E\_SymbolType

{

enSTBegin = -1,

enSTFuturesCn, // 国内期货

enSTStockCn, // 国内股票

enSTMony, // 外汇

enSTExp, // 指数栏目

enSTStockHk, // 港股栏目

enSTFuturesForeign, // 国外期货

enSTWarrantCn, // 国内权证

enSTWarrantHk, // 香港权证

enStRelaMonth, // 相对月期货

enStSpot, // 现货

enSTExpForeign, // 国外指数

enSTGoldSpot, // 黄金现货

enSTCnFuOptins = 20, // 大陆期权

enSTEnd,

};

## 四、功能列表

1-心跳

4-请求版本号

50-获取市场信息

51-获取商品详情

52-获取商品列表

53-请求商品排行

54-用户输入代码

55-指定商品排序

56-请求子市场列表

57-请求合约信息

58-附加条件查询商品

59-请求交易时间

60-获取期权列表

61-获取期权商品列表

62-请求公共板块配置

63-请求公共板块列表

64-获得财务数据

65-获得上涨下跌家数（市场）

66-请求领涨股

67-获得多个商品的财务数据

68-获得全市场列表

69-获得板块中商品的上涨下跌家数

70-请求差价表

71-请求AH、ADR对应关系

72-碎股

73-碎股财务

74-关联商品

75-FreeText

76-经纪人队列

77-搜索牛熊证

78-成交统计

79-搜索香港期权

80-券商进出 直接和行情请求保持一致

81-经纪人买卖

82-商品信息其他计算字段（字段太多了重新写个了）

83-全部关联商品

84-批量查询商品的市场

85- 陆港通

86-获取经纪人信息

87-获取交易日期

88-霸才精选查询

150-获取K线

151-获取分笔数据

152-获取分时数据

153-获取商品行情

154-使用合约号获取商品行情, 并添加推送

155-使用合约号获取商品行情

156-获取多分钟或多日K线, 并添加推送

157-通过合约号获取行情,需要完全市场和合约号

158-请求五日分时

200-请求推送(只支持websocket协议)

201-删除推送(只支持websocket协议)

202-同步推送(只支持websocket协议)

400-请求上传数据, 现在只有自选股

401-请求上传机构额外数据

402-请求用户数据

403-请求机构额外数据

## 五、接口说明

### 5.1 心跳(1)

#### 5.1.1请求

{

"reqtype": 1,

"reqid": 1,

"session": "",

"data": {

"connectionid": 1

}

}

#### 5.1.2回包

{

"status": 0,

"msg": "",

"reqid": 1,

"connectionid": 1,

"servertime": "2017-7-5 6:16:43"

}

#### 5.1.3说明

connectionid值由客户端自己维护。本协议只针对websocket连接，如果websocket客户端本身已经发送ping包，则不需要发本心跳包维持连接。

### 5.2 请求版本号(4)

#### 5.2.1请求

{

"reqtype":4,

"reqid":1,

"session":"",

"data": {}

}

#### 5.2.2说明

#### 5.2.3回包

{

"status":0,

"msg":"",

"servertime":"2017-7-5 6:16:43",

"reqtype":4,

"data":{

"funcver":2

}

}

### 5.3用户认证（现在认证已经在java那边实现，这个弃用）

#### 5.3.1 请求

{

"reqtype":5,

"reqid":1,

"session":"",

"data": {

"user":"test",

“pass”:”pass”

"authserid":123,

"userid":123,

"token":"12341asdefasf3241234234235134132412asdf",

"product":"ios",

"org\_code":"dz"

}

}

#### 5.3.2请求包字段说明

#### 5.3.3回包

{

"status":0,

"msg":"",

"reqid":1,

"servertime":"2017-7-5 7:58:58",

"reqtype":5,

"reqid":1,

"data":{

"netsession":"D3087712-3F68-4a17-863A-362941DC7DD3",

"market":[{

"type":0,

"bigmarket":0,

"market":0,

"symboltype":3,

"timezone":8,

"opentime":3600,

"showvolumeunit":100,

"tradetime":["2017-7-4 0:0:0","2017-7-5 0:0:0","2017-7-6 0:0:0","2017-7-7 0:0:0"],

"openclosetime":[{"open":90,"close":210},{"open":300,"close":420}]

}]}

}

#### 5.3.4回包字段说明

netsession即为其他请求所需的session值

### 5.4退出登录(3)

#### 5.4.1请求

{

"reqtype": 6,

"reqid": 1,

"session": "D3087712-3F68-4a17-863A-362941DC7DD3",

"data": {

}

}

#### 5.4.2回包

{

"status": 0,

"msg": "Logout Successfull!",

"reqtype": 6,

"reqid": 1

}

#### 5.4.3 说明

只针对http请求

### 5.5 获取市场信息(50)

#### 5.5.1请求

{

"reqtype": 50,

"reqid": 1,

"session": "",

"data": [{"market": 1001},{"market": 1002}]

}

#### 5.5.2字段说明

market: 市场编码

#### 5.5.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-22 5:55:53",

"reqtype": 50,

"reqid": 1,

"data": {

"market": [{

"type":0,

"bigmarket":1,

"market":1001,

"symboltype":1,

"timezone":8,

"opentime":3600,

"showvolumeunit":100,

"marketname":"SZSE",

"calctype":0,

"tradetime":["2017-7-50:0:0","2017-7-6 0:0:0","2017-7-7 0:0:0","2017-7-10 0:0:0"],

"openclosetime":[{"open":90,"close":210},{"open":300,"close":420}]

}]

}

}

#### 5.5.4字段说明

Type: 类型暂时不用

Bigmarket:大市场

market: 市场编码

Symboltyep:商品类型 （E\_symbolType）

Timezone:市场所在时区（中国是8）

Opentime:开盘时间

Callauction：为国内股票早上集合竞价的开始和结束时间，其余商品类型没有该字段

Showvolumeunit：成交单位

Marketname：市场名

Calctype：计算类型，0用昨昨收计算涨跌，1用昨结计算涨跌

Tradetime：交易日四个交易日

### 5.6 获取商品详情(51)

#### 5.6.1请求

{

"reqtype": 51,

"reqid": 1,

"session": "",

"data": [{

"market": 1000,

"code": "000001"

}]

}

#### 5.6.2字段说明

market 市场代码

code 商品代码

#### 5.6.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-19 0:40:9",

"reqtype": 51,

"reqid": 1,

"data": {

"symbol":

[{"market":3000,

"code":"010001",

"tradecode":"cu1801",

"name":"沪铜1801",

"shortname": "沪铜 01",

"enname": "滬銅 01",

"tradetimeid":2,

"tradetimetype":1,

"lotsize":5,

"minprice":"10",

"commoditymarket":-1

}

]}

}

#### 5.6.4 回包字段说明

market // 市场

code // 代码

tradecode // 合约号

name // 名字

shortname // 简称

twname // 繁体名

twsname // 繁体简称

enname // 英文名

ensname // 英文名

dec // 精确到小数点后多少位

symflag // 其它标志 e\_clientsymflag

tradetimeid // 交易时间id，如果为0表示使用市场的时间

tradetimetype // 交易时间类型(e\_tradetimetype)

lotsize // 每手合约乘数

minprice // 最小变动价

minpricediff // 最小价差

currencyunit // 币种单元

spreadtablecode // 扩展代码

commoditymarket; // 期权标的商品的市场id

commoditycode // 期权标的商品合约号

instrumentcode // 具体代码

inversecode // 比如购期权对应的是沽的期权代码

fuopttype; // 期权类型,沽或购 e\_fuoptiontype

fuoptexectype; // 期权类型,美式，欧式 e\_fuoptionexectype

expiredate; // 期权交割月份，到期日

begindate; // 期权行权开始日

enddate; // 期权行权终止日

strikeprice; // 期权行使价

strikeratio; // 行权比率,1.0(1:1) 0.33333(1:3)

fuoptshowtype // 期权显示类型，比如豆粕或白糖

fuoptshowsubtype // 期权显示二级类型，比如豆粕1407或白糖1407

basicdatatype // 基本数据类型

listingdate // 上市日期

delistingdate // 退市日期

time; // 此数据的更新时间(秒)

millisecond; // 此数据的更新时间(毫秒)

now; // 最新

open; // 开盘

high; // 最高

low; // 最低

avg; // 均价

settle; // 结算价

lastclose; // 昨收

lastsettle; // 昨结算

curvolume; // 现手

volume; // 总手

amount; // 总成交额

lasthold; // 昨持仓

curhold; // 最近一笔的增仓量

hold; // 总持仓量

volumerate // 量比

buyvolume // 内盘, 以买价成交

sellvolume // 外盘, 以卖坐成交

pe // 动态市盈率

traderate; // 换手率

moneyin; // 资金流向

riserate; // 涨速(最新价和上个5分钟收盘价的比率)

sellamount; // 外盘成交额

incomeseason; // 表示市益率是第几季度的(0-3)

easprice // 预计平均结算价

openprice //开盘价iep

openvol// 开盘量iev

netoi//净未平仓合约数

grossoi//未平仓合约数

bestbuyprice 最佳买价

bestsellprice 最佳卖价

buyprice ：买价12挡

buyvol ：买量12挡

buybroker：经纪人买12挡

sellprice：卖价12挡

sellvol：卖量12挡

sellbroker：经纪人卖12挡

orderimbdirection // iep时，买卖盘差额方向

orderimbquantity // iep时，买卖盘差值的绝对值

referenceprice // 收盘竞价(cas)时的参考价

referencelowerprice // cas允许的最低价

referenceupperprice // cas允许的最高价

vcmstarttmie // 熔断的开始时间(mytime)

vcmendtime // 熔断的结束时间(mytime)

vcmreferenceprice // vcm时的参考价

vcmlowerprice // vcm允许最低价

vcmupperprice // vcm允许最高价

tradecount; // 成交笔数

riselimit; // 涨停价

falllimit; // 跌停价

flag： // 板块id

### 5.7 获取商品列表(52)

#### 5.7.1请求

{

"reqtype":52,

"reqid":1,

"session":"",

"data":{

"marketid":3000,

"idtype":1,

"beginpos":0,

"count":10,

"getquote":0

}

}

#### 5.7.2字段说明

Marketid：市场或板块

Beginpos：开始位置

Count：个数

Getquote：是否同时获取行情

Idtype: 1 表示市场 3 表示板块

#### 5.7.3回包

{

"status":0,

"msg":"",

"servertime":"2017-6-19 0:38:16",

"reqtype":52,

"reqid":1,

"data":{

"marketid":3000,

"idtype":1,

"beginpos":0,

"totalcount":249,

"getquote":0,

"pushflag":0,

"symbol":[{

"market":3000,

"code":"010001",

"tradecode":"cu1801",

"name":"沪铜1801",

"shortname": "沪铜1801",

"enname": "沪铜1801",

"dec": 119,

"symflag": 5541013,

"lotsize": 0,

"commoditycode": ""

}]

}

}

#### 5.7.4说明

tradetimeid为交易时间段id，请与59-请求交易时间功能配合使用，该值与59功能返回数据中的id字段一一对应

回包结构体同5.6.4

### 5.8 请求商品排行(53)

#### 5.8.1请求

{

"reqtype":53,

"reqid":1,

"session":"D3087712-3F68-4a17-863A-362941DC7DD3",

"data":

{

"type":1,

"desc":0,

"beginpos":0,

"count":10,

"getquote":1,

"market":[{"marketid":3000, "idtype":1}]

}

}

#### 5.8.2字段说明

type 排序的依据

1-涨跌幅

2-涨跌

3-最新价

4-均价

5-结算价

6-现手

7-买价

8-卖价

9-总手

10-今开

11-昨收价

12-最高价

13-最低价

14-总金额

15-振幅

16-量比

17-委比

18-委差

19-内盘

20-外盘

21-内外盘比

22-昨结算

23-持仓

24-增仓

25-动态市盈率

26-换手率

27-资金流向

28-涨速(最新价和上个5分钟收盘价的比率)

29-名称代码

30-到期日

31-行使价

32-街货比

33-街货量

34-溢价

35-价内价外，暂不支持

36-引伸波幅，暂不支持

37-有效杠杆

38-收回价，暂不支持

39-据收回价，暂不支持

40-杠杆比率

41-换股比率

42-对冲值

43-涨跌额

44-成交量

45-成交额

46-市值

47-涨家数

48-跌家数

49-52周最低

50-52周最高

51-持股数

52-持股比例

53-持仓额

desc 正序还是逆序,1为正序(从大到小)，0为逆序（从小到大）

beginpos count 用来分布

getquote 填1就好

marketid：可能是市场id，也可能是公共板块id，根据idtype的值来确定

idtype:

EMTBlock 为0

EMTBlock=0, // 板块

EMTMarket=1, // 市场

EMTOther=2, // 主力行情(uMarketID:0)或快捷市场(uMarketID>0)

EMTPublicBlock=3, // 公共板块

EMTNightActive=4, // 夜盘主力

注意：请求热门行业和热门概念中指数商品对应的板块ID，保存在该指数商品的合约号中。需要拿到该合约号，才能去获取该板块下面的商品列表。

#### 5.8.3回包

{

"status":0,

"msg":"",

"servertime":"2017-6-23 5:38:53",

"reqtype":53,

"reqid":1,

"data":

{"beginpos":0,

"totalcount":249,

"getquote":1,

"symbol":[

{"market":3000,

"code":"010204",

"tradecode":"fu1704",

"name":"燃油1704",

"lotsize":50,

"minprice":"1",

"commoditymarket":-1,

"time":"2017-6-23 5:39:55",

"now":3773.0,

"open":3773.0,

"high":3773.0,

"low":3773.0,

"avg":3773.0,

"settle":3773.0,

"lastclose":3963.0,

"lastsettle":3963.0,

"curvolume":2.0,

"volume":2.0,

"lasthold":2.0,

"hold":2.0,

"volumerate":0.344760000705719,

"buyvolume":2.0,

"buyprice0":3802.0,

"buyvol0":1.0,

"sellprice0":4049.0,

"sellvol0":1.0

}]

}

}

#### 5.8.4说明

buyprice0,buyvol0表示买一价和买一量，二三四五档数据依次类推

回包同5.6.4

### 5.9 用户输入代码(54)

#### 5.9.1请求

{

"reqtype":54,

"reqid":1,

"session":"",

"data": {

"code":"000001"

}

}

#### 5.9.2回包

{

"status":0,

"msg":"",

"servertime":"2017-6-27 8:59:34",

"reqtype":54,

"reqid":1,

"data":{

"symbol":[

{"market":0,

"code":"000001",

"name":"上证指数",

"shortname": "上证指数",

"dec":2,

"symflag": 126568782,

"commoditymarket":-1}]

}

}

#### 5.9.3说明

回包同5.6.4

### 5.10 指定商品排序(55)

#### 5.10.1请求

{

"reqtype":55,

"reqid":1,

"session":"D3087712-3F68-4a17-863A-362941DC7DD3",

"data": {

"type":1,

"desc":0,

"beginpos":0,

"count":2,

"getquote":1,

"symbol":[{

"market":1001,

"code":"000001"

},

{ "market":1001,"code":"000002"}]

}

}

#### 5.10.2说明

#### 5.10.3回包

{

"status":0,

"msg":"",

"servertime":"",

"reqtype":55,

"reqid":1,

"data":{"beginpos":0,"totalcount":2,"getquote":1,

"symbol":[

{

"market":1001,

"code":"000001",

"name":"平安银行",

"dec":2,

"commoditymarket":-1,

"time":"2017-7-5 7:2:3",

"now":9.369999885559082,

"open":9.289999961853028,

"high":9.380000114440918,

"low":9.270001411437989,

"avg":9.323748588562012,

"settle":9.323748588562012,

"lastclose":9.34000015258789,

"lastsettle":9.34000015258789,

"curvolume":797180.0,

"volume":56772000.0,

"amount":529294112.0,

"volumerate":1.2374509572982789,

"buyvolume":26354784.0,

"sellvolume":30417216.0,

"moneyin":37931744.0,

"riserate":0.10683999955654144,

"sellamount":283689312.0,

"buyprice0":9.369999885559082,

"buyprice1":9.359999656677246,

"buyprice2":9.350000381469727,

"buyprice3":9.34000015258789,

"buyprice4":9.329999923706055,

"buyvol0":61120.0,

"buyvol1":130260.0,

"buyvol2":238559.0,

"buyvol3":308700.0,

"buyvol4":653900.0,

"sellprice0":9.380000114440918,

"sellprice1":9.390000343322754,

"sellprice2":9.399999618530274,

"sellprice3":9.40999984741211,

"sellprice4":9.420000076293946,

"sellvol0":612000.0,

"sellvol1":969638.0,

"sellvol2":1282741.0,

"sellvol3":349000.0,

"sellvol4":196700.0,

"riselimit":10.270000457763672,

"falllimit":8.40999984741211}

]}

}

#### 5.10.4 说明

回包同5.6.4

### 5.11 请求子市场列表(56)

#### 5.11.1 请求

{

"reqtype":56,

"reqid":1,

"session":"",

"data": {}

}

#### 5.11.2字段说明

#### 5.11.3回包

{

"status":0,

"msg":"",

"servertime":"2017-7-5 10:48:18",

"reqtype":56,

"reqid":1,

"data":{

"submarket":[

{"submarketid":1,

"market":3000,

"submarketname":"沪铜",

"tradecode":"CU",

"py":"HT"},

{"submarketid":6,"market":3002,"submarketname":"棉花",

"tradecode":"CF",

"py":"MH"}

]}

}

#### 5.11.4字段说明

"submarketid":子市场id

"submarketname":子市场名称

"tradecode":合约号

"py":拼音

### 5.12 请求合约信息(57)

#### 5.12.1请求

{

"reqtype":57,

"reqid":1,

"session":"",

"data": {}

}

#### 5.12.2请求字段说明

#### 5.12.3回包

{

"status":0,

"msg":"",

"servertime":"2017-7-6 0:30:8",

"reqtype":57,

"reqid":1,

"data":{

"dealinfo":[{

"marketid":3000,

"dealcount":5,

"dealcode":"CU",

"minprice":10.0

},

{"marketid":3000,"dealcount":5,"dealcode":"AL","minprice":5.0},

{"marketid":3002,"dealcount":5,"dealcode":"CF","minprice":5.0}

]}

}

#### 5.12.4回包字段说明

marketid; // 属于哪个市场

dealcode; // 合约名字比如CU

dealcount; // 每手合约数，比如铜为5, 白糖为10

minprice; // 最小变动价

### 5.13 附加条件查询商品(58)

#### 5.13.1请求

{

"reqtype":58,

"reqid":1,

"session":"D3087712-3F68-4a17-863A-362941DC7DD3",

"data": {

"market":[{"marketid":1001，},{"marketid":0}],

"code":"000001",

"tradecode":"000001",

"name":"000001"

}

}

#### 5.13.2字段说明

如果marketid中有一个为负数，则会匹配所有市场，实现模糊匹配，请将code、tradecode、name三个字段填相同的值，支持汉字、全拼、拼音首字母的匹配。

#### 5.13.3回包

{

"status":0,

"msg":"",

"servertime":"2017-7-1 3:26:45",

"reqtype":58,

"reqid":1,

"data":{

"symbol":[{

"market":0,

"code":"000001",

"name":"上证指数",

"dec":2,

"commoditymarket":-1},

{"market":1001,"code":"000001","name":"平安银行","dec":2,"commoditymarket":-1}

]}

}

#### 5.13.4字段说明

回包同5.6.4

### 5.14 请求交易时间(59)

#### 5.14.1请求

{

"reqtype":59,

"reqid":1,

"session":"",

"data":[{"tradetimeid":0}]

}

#### 5.14.2字段说明

tradetimeid // 交易时间ID，0则请求返回全部

#### 5.14.3回包

{

"status":0,

"msg":"",

"servertime":"2017-7-6 0:41:38",

"reqtype":59,

"reqid":1,

"data":{

"tradetime":[

{"id":1,"time":[{"week":10,"open":-670,"close":450,

"time":[{"opentime":-660,"closetime":-330},

{"opentime":60,"closetime":135},

{"opentime":150,"closetime":210},

{"opentime":330,"closetime":420}]}

]},

{"id":2,"time":[{"week":10,"open":-670,"close":450,"time":[{"opentime":-660,"closetime":-420},{"opentime":60,"closetime":135},{"opentime":150,"closetime":210},{"opentime":330,"closetime":420}]}]}

]}

}

#### 5.14.4说明

week // 星期几的时间，0-6表示星期天到星期六，其它时间表示通用时间

C++ 时间结构体

typedef struct \_PhoneTradeTimeRes

{

u32 iID; // ID

u32 uRe[2];

u8 iCount; // 交易时间个数

T\_PhoneTradeTime stTime[1]; // 交易时间,非固定个数由iCount决定

}T\_PhoneTradeTimeRes;

typedef struct \_PhoneTradeTime

{

u8 iWeek; // 星期几的时间，0-6表示星期六到星期天，其它时间表示通用时间

int16 iOpen; // 开盘时间

int16 iClose; // 收盘时间

u8 iCount; // 时间个数

u32 uRe[2]; // 保留

T\_TradeTime stTime[8]; // 交易时间节, 非固定个数由iCount决定

}T\_PhoneTradeTime;

typedef struct \_TradeTime

{

int16 iOpenTime; // 开盘

int16 iCloseTime; // 收盘

} T\_TradeTime;

### 5.15 获取期权列表(60)

#### 5.15.1请求

{

"reqtype": 60,

"reqid": 1,

"session": "",

"pushflag": ,

"data":[{"market":0, "tradecode":"000001"},

{"market":1, "tradecode":"600000"}

]

}

#### 5.15.2字段说明

Pushflag：推送标志

#### 5.15.3回包

{

"status": 0,

"msg": "",

"reqtype": 73,

"session": "",

"data":[{"size": 0,

"market": "",

"code": 76,

"time": 0,

"millisecond": "",

"buycount": "",

"sellcount": 76,

"buy":[{"numoforders": 1, "numofshares": "1","queueprice": 76},

{"numoforders": 1, "numofshares": "1","queueprice": 76}]，

"sell": [{"numoforders": 1, "numofshares": "1","queueprice": 76},

{"numoforders": 1, "numofshares": "1","queueprice": 76}]

}]

}

#### 5.15.4说明

### 5.16 获取期权商品列表(61)

#### 5.16.1请求

{

"reqtype": 61,

"reqid": 1,

"session": "",

"data":{"market": 2002,

"code": 00001,

"pushflag": "",

"codecount": 1,

"language": ""

}

}

#### 5.16.2字段说明

Language: 0简体，1繁体，2英文

#### 5.42.3回包

{

"status": 0,

"msg": "",

"reqtype": 61,

"data": {"size": 0,

"market": "",

"code": 76,

"time": 0,

"millisecond": "",

"buy": 76,

"sell": 0,

"buycount": "",

"sellcount": 76,

“brokerItem”:[{"code": 0, "flag": "","name": 76},

{"code": 0, "flag": "","name": 76}]

}

}

#### 5.16.4说明

Size：数据包大小

Flag:是档位还是券商信息 0-券商 1-档位

### 5.17请求公共板块配置(63)

#### 5.17.1请求

{

"reqtype":62,

"reqid":1,

"session":"",

"data": {"crc32":0},

"language": 0

}

#### 5.17.2 字段说明

Language: 0简体，1繁体，2英文

#### 5.17.3回包

{

"status":0,

"msg":"",

"servertime":"2017-7-6 2:10:5",

"reqtype":62,

"reqid":1,

"data":{

"crc32":2020621486,

"content"："xml文件内容"

}

}

#### 5.17.4说明

### 5.18 请求公共板块列表(64)

#### 5.18.1请求

{

"reqtype":63,

"reqid":1,

"session":"D3087712-3F68-4a17-863A-362941DC7DD3",

"data": {"blockid":7, "beginpos":0,"count":10}

}

#### 5.18.2说明

7为热门行业下的板块，8为热门概念下的板块

#### 5.18.3回包

{

"status":0,

"msg":"",

"servertime":"2017-6-22 3:52:11",

"reqtype":63,

"reqid":1,

"data":{

"blockid":2,

"beginpos":0,

"count":10,

"totalcount":28,

"block":[

{"blockid":2000,

"parentid":2,

"type":1,

"symcount":3265,

"name":"沪深Ａ股"},

]}

}

#### 5.18.3回包

C++ 板块链表结构体

typedef struct \_PhoneRequestSubBlockListRes

{

u16 uSize; // 此结构大小，下一个结构开始加上此长度

u16 iBlockID; // 板块ID,0表示最上层板块

u16 iParentID; // 父ID

u8 iType; // 板块类型

wchar\_t wszName[32]; // unicode编码

u32 iSymCount; // 此板块商品个数

u32 uRev[4]; // 保留

}T\_PhoneRequestSubBlockListRes;

### 5.19 获得财务数据(64)

#### 5.19.1 请求

{

"reqtype":64,

"reqid":1,

"session":"",

"data": {"market":2002, "code":"00700"}

}

#### 5.19.2字段说明

#### 5.19.3回包

{

"status":0,

"msg":"",

"servertime":"",

"reqtype":64,

"reqid":1,

"data":{

"market": 2002, // 市场代码

"code": "00700", // 商品代码

"currencycode": "RMB", // 货币代码

"showunit": 1000000, // 显示单位

"reportdate": "Aug 18(18I)", // 公布日期

"yearend": "06-18I", // 财务年度

"indexname": "恒生指數", // 所属指数

"indexweight": 10.12399959564209, // 指数比重

"indexsp": 1.659000039100647, // 每价位指数变动

"marketcap": 3311187525632, // 市场总值

"allcapital": 9520377856, // 总股本

"circcapital": 9520377856, // 流通股本

"allasset": 646808010752, // 总资产

"flowdebt": 178874007552, // 流动负债

"businessprofit": 48397000704, // 营业利益

"perincome": 9.119999885559082, // 每股收益

"perpureasset": 32.356998443603519, // 每股净资产

"chperpureasset": 32.356998443603519, // 调整每股净资产

"dorrightrate": 0.4560000002384186, // 股东权益比

"incomeseason": 1, // 收益季度

"yearhigh": 353, // 今年高

"yearlow": 300.3999938964844, // 今年低

"week52high": 475.6000061035156, // 52周最高

"week52low": 251.39999389648438, // 52周最低

"yearpricechange": -17.65999984741211, //52 周股价变动

"yearmarketchange": -13.15999984741211, // 52周股价相对恒生指数变动

"avgvolume": 18969012, // 年平均成交量

"avgturnover": 6308316672, // 年平均成交额

"ma10": 346.6600036621094, // ma10

"ma20": 340.0299987792969, // ma20

"ma50": 324.2120056152344, // ma50

"ma90": 307.2200012207031, // ma90

"ma250": 361.01800537109377, // ma250

"rsi10": 59.2599983215332, // rsi10

"osc10": 1.003000020980835, // osc10

"lastpe": 38.130001068115237, //年化市盈率

"yearyield": 0.25, // 年化周息率

"lastprice2pa": 10.75, // 股价/每股净资

"lastprice2cashps": 18.420000076293947, // 股价/每股现金

"lastprice2debtps": 28.149999618530275, // 股价/每股负债

"lastepsgrowth": 29.1299991607666, // 每股盈利增长

"dividendps": 0.8799999952316284, // 每股派息

"cashps": 18.87700080871582, // 每股现金

"debtps": 12.357000350952149, // 每股负债

"cashflowps": 15.163000106811524, // 每股现金流动

"turnover": 147202998272, // 营业额

"pecialitem": 2296999936, // 特殊项目

"profitbeforetax": 51901001728, // 税前利益

"tax": 9347999744, // 税项

"shareholdertoprofit": 41157001216, // 股东应占盈利

"depreciation": 3567000064, // 拆旧

"fixasset": 42155999232, // 固定资产

"invest": 350598987776, // 投资

"currentasset": 181850996736, // 流动资产

"cash": 130312003584, // 现金及银行结存

"inventory": 306000000, // 存货

"shortdebt": 28109000704, // 短期负债

"longdebt": 77876002816, // 长期负债

"shareholderequity": 294786007040, // 股东权益

"currentratio": 1.0169999599456788, // 流动比率

"quickratio": 1.0149999856948853, // 速动比率

"longdebt2equity": 26.417999267578126, // 长期负债/股东权利

"alldebt2equity": 35.952999114990237, // 总负债/股东权益

"alldebt2capex": 22.649999618530275, // 总负债/资本运用

"returnonequity": 27.92300033569336, // 股东权益回报率

"returnontotalasset": 12.72599983215332, // 总资产回报率

"operatingmargin": 35.66400146484375, // 经营利润率

"profitmargin": 27.958999633789064, // 编辑利润率

"inventoryturnoverrate": 962.1110229492188 // 存货周转率

}

}

#### 5.19.4字段说明

allcapital; // 总股本

circcapital; // 流通股本

allasset; // 总资产

flowdebt; // 流动负债

perfund; // 每股公积金

businessprofit; // 营业利益

pernodistribute; // 每股未分配

perincome; // 每股收益

perpureasset; // 每股净资产

chperpureasset; // 调整每股净资产

dorrightrate; // 股东权益比

incomeseason; // 收益季度

### 5.20 获得上涨下跌家数（市场）(65)

#### 5.20.1请求

{

"reqtype": 65,

"reqid": 1,

"session": "",

"data": [{"market":1},{"market":2}]

}

#### 5.20.2请求字段说明

#### 5.20.3回包

{

"status":0,

"msg":"",

"servertime":"2017-6-22 3:52:11",

"reqtype":65,

"reqid":1,

"data":[{"market”:1,”up”:23,”down”:987},{"market”:2,”up”:537,”down”:86}]

}

#### 5.20.4回包字段说明

### 5.21 请求领涨股(66)

#### 5.21.1请求

{

"reqtype": 66,

"reqid": 1,

"session": "",

"data":{

"type":1,

"getquote":1,

"desc":0,

"market":[

{"marketid":3000, "idtype":1},

{"marketid":0, "idtype":1},

{"marketid":1001, "idtype":1}

]

}

}

#### 5.21.2字段说明

desc 填0，返回的领涨股按请求的顺序来

#### 5.21.3回包

{

"status":0,

"msg":"",

"servertime":"2017-6-24 5:56:11",

"reqtype":66,

"reqid":1,

"data":{

"beginpos":0,

"totalcount":3,

"getquote":1,

"symbol":[

{"market":1001,"code":"000001","tradecode":"000001","name":"平安银行","dec":2,"commoditymarket":-1},

{"market":0,"code":"000001","tradecode":"000001","name":"上证指数","dec":2,"commoditymarket":-1},

{"market":3000,"code":"010001","tradecode":"cu1701","name":"沪铜1701","tradetimeid":2,"tradetimetype":1,"lotsize":5,"minprice":"10","commoditymarket":-1}

]

}

}

#### 5.21.4 说明

财务数据C++结构体

typedef struct \_PhoneFinanceRes

{

ushort iSize; // 整个结构大小

ushort iDataBegin; // 指向数据开始的地方

u16 iMarket;

char szCode[MAX\_CODE];

uint uValidFlag[4]; // 有效标志 判断代码(uValidFlag&(1<<(E\_SymFinanceResUpdateFlag%32)))

// 下面数据根据有效标志来定，无效则不存在

char szCurrencyCode[8]; // 报表货币代码

float fShowUnit; // 显示单位

char szReportDate[32]; // 公布日期

char szYearEnd[32]; // 财务年度

char szIndexName[MAX\_NAME]; // 所属指数

float fIndexWeight; // 指数比重

float fIndexSp; // 每价位指数变动

float fMarketCap; // 市价总值

float fAllCapital; // 总股本

float fCircCapital; // 流通股本

float fAllAsset; // 总资产

float fFlowDebt; // 流动负债

float fPerFund; // 每股公积金

float fBusinessProfit; // 营业利益

float fPerNoDistribute; // 每股未分配

float fPerIncome; // 每股收益

float fPerPureAsset; // 每股净资产

float fChPerPureAsset; // 调整每股净资产

float fDorRightRate; // 股东权益比

u8 iIncomeSeason; // 收益季度

float fYearHigh; // 今年高

float fYearLow; // 今年低

float fWeek52High; // 52周高

float fWeek52Low; // 52周低

float fYearPriceChange; // 52周股价变动

float fYearMarketChange; // 52周股价相对恒生指数变动

float fAvgVolume; // 年平均成交量

float fAvgTurnover; // 年平均成交额

float fMa10; // ma10

float fMa20; // ma20

float fMa50; // ma50

float fMa90; // ma90

float fMa250; // ma250

float fRSI10; // rsi10

float fOSC10; // osc10

float fLastPE; // 年化市盈率

float fYearHighPE; // 年最高PE

float fYearLowPE; // 年最低PE

float fYearYield; // 年化周息率

float fYearHighYield; // 年最高周息率

float fYearLowYield; // 年最低周息率

float fLastPrice2PA; // 股价/每股净资

float fLastPrice2CashPS; // 股价/每股現金

float fLastPrice2DebtPS; // 股价/每股负债

float fLastEPSGrowth; // 每股盈利增长

float fDividendPS; // 每股派息

float fCashPS; // 每股现金

float fDebtPS; // 每股负债

float fCashFlowPS; // 每股现金流动

float fTurnover; // 营业额

float fPecialItem; // 特殊项目

float fProfitBeforeTax; // 税前利益

float fTax; // 税项

float fShareHolderToProfit; // 股东应占赢利

float fDepreciation; // 拆旧

float fInterestOnLoans; // 贷款利息

float fInterestCap; // 利息拨作发展资本

float fFixAsset; // 固定资产

float fInvest; // 投资

float fCurrentAsset; // 流动资产

float fCash; // 現金及银行结存

float fInventory; // 存货

float fShortDebt; // 短期负债

float fLongDebt; // 长期负债

float fShareHolderEquity; // 股东权益

float fCurrentRatio; // 流动比率

float fQuickRatio; // 速动比率

float fLongDebt2Equity; // 长期负债/股东权益

float fAllDebt2Equity; // 总负债/股东权益

float fAllDebt2Capex; // 总负债/资本运用

float fNetDebt2Equity; // 净债项/股东权益

float fReturnOnEquity; // 股东权益回报率

float fReturnOnTotalAsset; // 总资产回报率

float fOperatingMargin; // 经营利润率

float fProfitMargin; // 边际利润率

float fInventoryTurnoverRate; // 存货周转率

float fDividendPayoutRatio; // 派息比率

float fSubscriptionRate; // 认购比率

float fLeverageRatio; // 杠杆比率

float fPremium; // 溢价

float fWarrantInMarket; // 市场持货量

float fWarrantInMarketPer; // 市场持货量(%)

float fCapitalAdequacyRatio; // 资本充足率

float fCost2Income; // 成本/收入

float fLoanGrowth; // 贷款增长

float fDepositGrowth; // 存款增长

float fCurAsset2Deposit; // 流动资产/存款

float fLoan2Deposit; // 贷款/存款

float fLoan2Equity; // 贷款/股东权益

float fLoan2TotalAsset; // 贷款/总资产

float fDeposit2Equity; // 存款/股东权益

float fDeposit2TotalAsset; // 存款/总资产

float fROAvgLoans; // 贷款回报率

float fROAvgDeposits; // 存款回报率

float fInterestIncome; // 利息收入

float fOtherIncome; // 其它营业收入

float fOperatExpense; // 营业支出

float fProvisionForBadDebt; // 呆坏账准备

float fDerivativeAsset; // 出售资产

float fLoans; // 贷款

float fCashAndShortTermFund; // 现金及短期资金

float fOtherAsset; // 其它流动资产

float fDeposits; // 存款

float fOtherLiab; // 其他流动负债

}T\_PhoneFinanceRes;

### 5.22 获得多个商品的财务数据(67)

#### 5.22.1请求

{

"reqtype": 67,

"reqid": 1,

"session": "",

"data": [{

"market": 1,

"code": "600000"

}]

}

#### 5.22.2字段说明

#### 5.22.3回包

{

"status":0,

"msg":"",

"servertime":"",

"reqtype":67,

"reqid":1,

"data":[{

"market":1,

"code":"600000",

"allcapital":28103766016.0,

"circcapital":28103766016.0,

"allasset":5889662648320.0,

"perfund":345.443603515625,

"businessprofit":18396999680.0,

"pernodistribute":512.7578735351563,

"perincome":0.5145999789237976,

"perpureasset":16.200000762939454,

"dorrightrate":6.449999809265137}

]

}

#### 5.22.4说明

### 5.23 获得全市场列表(68)

#### 5.23.1请求

{

"reqtype": 68,

"reqid": 1,

"session": "",

"data": { }

}

#### 5.23.2字段说明

#### 5.23.3回包

{

"status":0,

"msg":"",

"servertime":"",

"reqtype":68,

"reqid":1,

"data":{

"market":[

{"bigmarket":0,"market":0,"symboltype":3},

{"bigmarket":0,"market":1,"symboltype":1},

{"bigmarket":0,"market":2,"symboltype":1},

{"bigmarket":0,"market":3,"symboltype":1},

{"bigmarket":0,"market":4,"symboltype":1},

{"bigmarket":0,"market":5,"symboltype":1},

{"bigmarket":0,"market":6,"symboltype":6},

{"bigmarket":0,"market":101,"symboltype":1},

{"bigmarket":1,"market":1000,"symboltype":3},

{"bigmarket":1,"market":1001,"symboltype":1}

]}

}

#### 5.23.4说明

### 5.24 获得板块中商品的上涨下跌家数(69)

#### 5.24.1请求

{

"reqtype": 69,

"reqid": 1,

"session": "",

"data": [{"block":1},{"block":2}]

}

#### 5.24.2字段说明

#### 5.24.3回包

{

"status":0,

"msg":"",

"servertime":"2017-6-22 3:52:11",

"reqtype":69,

"reqid":1,

"data":[{"block:1,”up”:23,”down”:987},{"block:2,”up”:537,”down”:86}]}

}

#### 5.24.4说明

### 5.25 差价表(70)

#### 5.25.1请求

{"reqtype":70,"reqid":1, "session":"", "data": {"market":2002}}

#### 5.25.2字段说明

#### 5.25.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-3-21 1:18:46",

"reqtype": 70,

"reqid": 1,

"data": {

"market": 2002,

"tablecount": 3,

"spreadtabledata": [

{

"code": "01",

"numofitems": 50,

"pricefrom": 10,

"size": 1,

"spreadtable": [

{

"price": 100,

"spreadvalue": 1

}

]

}

]

}

}

#### 5.25.4说明

### 5.26 AH与ADR配置(71)

#### 5.26.1请求

{

"reqtype": 71,

"reqid": 1,

"session": "",

"data": {"flag":0}

}

#### 5.26.2字段说明

flag:0为AH股列表，1为ADR列表

#### 5.26.3回包

{

"status": 0,

"msg": "",

"reqtype": 71,

"data":

{

"AH":[

["am","ac","hm","hc"],

[1,"600000",2002,"00001"]

],

"ADR":[

["hm","hc","mm","mc","r"],

[2002,"00001",40000,"BABA",50]

]

}

}

#### 5.26.4说明

AH：AH列表,am：A股市场，ac：A股商品代码，hm：港股市场，hc：港股代码

ADR：ADR列表，hm：港股市场，hc：港股代码，mm：美股市场，mc美股代码,r 换股比例

### 5.27 碎股(72)

#### 5.27.1请求

{

"reqtype": 72,

"reqid": 1,

"session": "",

"pushflag": ,

"data":[{"market":0, "tradecode":"000001"},

{"market":1, "tradecode":"600000"}

]

}

#### 5.27.2字段说明

Pushflag：推送标志

#### 5.27.3回包

{

"status": 0,

"msg": "",

"reqtype": 73,

"session": "",

"data":[{"size": 0,

"market": "",

"code": 76,

"time": 0,

"millisecond": "",

"buycount": "",

"sellcount": 76,

"buy":[{"numoforders": 1, "numofshares": "1","queueprice": 76},

{"numoforders": 1, "numofshares": "1","queueprice": 76}]，

"sell": [{"numoforders": 1, "numofshares": "1","queueprice": 76},

{"numoforders": 1, "numofshares": "1","queueprice": 76}]

}]

}

#### 5.27.4说明

### 5.28查询具体商品的关联商品(74)

#### 5.28.1请求

{

"reqtype":74,

"reqid":1,

"session":"",

"data": [{"market":2002,"code":"00700"}]

}

#### 5.28.2请求字段说明

market 市场

code 商品代码

#### 5.28.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-3-11 11:9:21",

"reqtype": 74,

"reqid": 1,

"data": {

"market": 2002,

"code": "00700",

"linksymcount": 2,

"linksymitem": [

{

"market": 2003,

"code": "11042"

},

{

"market": 2003,

"code": "11045"

}

]

}

}

#### 5.28.4回包字段说明

### 5.29 经纪人队列(76)

#### 5.29.1请求

{

"reqtype": 76,

"reqid": 1,

"session": "",

"data":{"market": 2002,

"code": "00001",

"pushflag":0,

"codecount": 1,

"language": 0

}

}

#### 5.29.2字段说明

Language: 0简体，1繁体，2英文

#### 5.29.3回包

{

"status": 0,

"msg": "",

"reqtype": 76,

"data": {"size": 0,

"market": "",

"code": 76,

"time": 0,

"millisecond": "",

"buyvalid": 76,

"sellvalid": 0,

"buycount": 40,

"sellcount": 40,

“buy”:[{"level": 1, "list": [{"code": 0, "flag": "","name": 76},{"code": 0, "flag": "","name": 76}]}],

“sell”:[{"level": 1, "list": [{"code": 0, "flag": "","name": 76},{"code": 0, "flag": "","name": 76}]}],

}

}

#### 5.29.4说明

Size：数据包大小

Flag:是档位还是券商信息 0-券商 1-档位

### 5.30 按条件筛选涡轮牛熊商品(77)

#### 5.30.1请求

{

"reqtype": 77,

"reqid": 1,

"session": "",

"data":{

"type":1,

"publisher": "AAA",

"enddate": 1,

"leverageratio": 1,

"strikeprice": {"low":10.1,"high":100.0},

"warrantinmarketper": {"low":10.1,"high":100.0},

"subscriptionrate": {"low":10.1,"high":100.0},

"premium": {"low":10.1,"high":100.0},

"recycling": {"low":10.1,"high":100.0},

"amplitudeexplicate": {"low":10.1,"high":100.0},

"beginpos":0,

"count":10,

"sorttype":1,

"desc":0

}

}

#### 5.30.2字段说明

搜索的条件是下列10个，值为空或者为0的时候，表示该字段不作为搜索条件

type-类型： 1:认购 2：认沽 3：牛证 4：熊证 5：全部 6：涡轮（认购+认沽） 7：牛熊（牛证+熊证）

publisher-发行人

enddate-到期日：1：所有2：小于3个月，3：3-6个月，4:6-12个月，5：大于12个月

leverageratio-杠杆比例: 1：小于1倍，2:1-5倍，3:5-10倍，4：大于10倍，5：所有默认

strikeprice-行使价：5

warrantinmarketper-街货比率：6

subscriptionrate-换购比例：7

premium-溢价：8

recycling-收回价：9，暂不支持

amplitudeexplicate-引伸波动：10，暂不支持

排序规则如下：

desc 正序还是逆序,1为正序(从大到小)，0为逆序（从小到大）

beginpos count 用来分页

sorttype 排序的依据

1-涨跌幅

2-涨跌

3-最新价

4-均价

5-结算价

6-现手

7-买价

8-卖价

9-总手

10-今开

11-昨收价

12-最高价

13-最低价

14-总金额

15-振幅

16-量比

17-委比

18-委差

19-内盘

20-外盘

21-内外盘比

22-昨结算

23-持仓

24-增仓

25-动态市盈率

26-换手率

27-资金流向

28-涨幅

29-名称代码

30-到期日

31-行使价

32-街货比

33-街货量

34-溢价

35-价内价外，暂不支持

36-引伸波幅，暂不支持

37-有效杠杆

38-收回价，暂不支持

39-据收回价，暂不支持

40-杠杆比率

41-换股比率

42-对冲值

#### 5.30.3回包

{

"status": 0,

"msg": "",

"reqtype": 76,

"data": {

"beginpos":0,

"totalcount":249,

"symbol":[{"market":2005, -市场id

"code":"07435", -商品code,

"enname":"英文",

"name":"简体",

"twname":"繁体",

"now":1.0, -现价

"raiserate":0.1, -涨跌幅

"raise":1.0, -涨跌

"volume":100, -成交量

"amount":105.0, -成交额

"enddate":20190114, -到期日

"strikeprice":1.0, -行使价

"warrantinmarketper":0.1, -街货比

"warrantinmarket":100.0, -街货量

"premium":1.0, -溢价

"inout":1.0, -价内价外

"amplitudeexplicate":0.1, -引伸波幅

"levertrue":10, -实际杠杆

"recycling":1.0, -收回价

"recyclingdiff":1.0, -距收回价

"leverageratio":0.1, -杠杆比率

"subscriptionrate":0.1,-换股比率

"delta":1.0 -对冲值

}]

}

}

#### 5.30.4说明

### 5.31成交统计(78)

#### 5.31.1请求

{

"reqtype": 78,

"reqid": 1,

"session": "",

"market": 1,

"code": "600000"

}

#### 5.31.2字段说明

#### 5.31.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-7-5 3:20:56",

"reqtype": 78,

"reqid": 1,

"data": {

"market": 1,

"code": "600000",

"dealcount": [{"price": 12.55, "bvolumn": 1,"svolumn": 1555.10,"avolumn": 1555.10},

{"price": 12.55, "bvolumn": 1,"svolumn": 1555.10,"avolumn": 1555.10},

] }

}

#### 5.31.4说明

Flag：内外盘标志（1 内盘 2 外盘）

以市场和商品代码请求当天所有的分笔，服务器根据请求到的数据，以价格和内外屏标志为key，计算成交额，返回给前端

### 5.32期权查询(79)

#### 5.32.1请求

{"reqtype":79,"reqid":1, "session":"", "data": {"market":2002, "code":"00700", "month":2}}

#### 5.32.2请求字段说明

market 市场

code 商品代码

#### 5.32.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-2-26 1:42:1",

"reqtype": 79,

"reqid": 1,

"data": {

"market": 2007,

"count": 10,

"symbol": [

{

"market": 2007,

"code": ".A5010.00B9",

"tradecode": "A5010.00B9",

"name": "A5010.00B9",

"shortname": "A5010.00B9",

"twname": "A5010.00B9",

"twsname": "A5010.00B9",

"enname": "A5010.00B9",

"ensname": "A5010.00B9",

"dec": 2,

"currencycode": "HKD",

"commoditycode": "A50",

"instrumentcode": "6",

"expiredate": 20190227,

"strikeprice": 10,

"time": "2019-2-26 1:30:0",

"millisecond": 411,

"lastclose": 4.28000020980835,

"lastsettle": 4.28000020980835,

"lasthold": 6,

"hold": 6,

"grossoi": 6

}

]

}

}

#### 5.32.4回包字段说明

行情回包字段同153

### 5.33券商进出(80)

#### 5.33.1请求

{"reqtype":80,"reqid":1, "session":"", "data": {"market":2002, "code":"00700"}}

#### 5.33.2请求字段说明

market 市场

code 商品代码

#### 5.33.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-2-23 6:43:37",

"reqtype": 80,

"reqid": 1,

"data": {

"market": 2002,

"code": "00700",

"count": 3,

"brokerdata": [

"broker",

{

"type": 1,

"count": 0,

"brokeriodata": []

},

"broker",

{

"type": 2,

"count": 20,

"brokeriodata": [

"brokerio",

{

"code": 1196,

"buy": 1385,

"sell": 10

},

"brokerio",

{

"code": 5465,

"buy": 1339,

"sell": 43

}

]

},

"broker",

{

"type": 3,

"count": 20,

"brokeriodata": [

"brokerio",

{

"code": 1196,

"buy": 1385,

"sell": 10

}

]

}

]

}

}

#### 5.33.4回包字段说明

type：1 一日 2 五日 3 二十日，

code：经济人代码

buy：买入次数，

sell：卖出次数

### 5.34商品其他信息查询(82)

#### 5.34.1请求

{

"reqtype":82,

"reqid":1,

"session":"",

"data": [{"market":2002,"code":"00001"}]

}

#### 5.34.2请求字段说明

market 市场

code 商品代码

#### 5.34.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-3-5 9:31:6",

"reqtype": 82,

"reqid": 1,

"data": {

"symbol": [

{

"market": 2002,

"code": "00001",

"tradecode": "0001",

"name": "",

"shortname": "",

"twname": "",

"twsname": "",

"enname": "CKH HOLDINGS",

"ensname": "CKH HOLDINGS",

"premium": 0,

"streetratio": 0,

"streetvolumn": 0,

"holdamount": 0,

"splitoffratio": 0,

"inandoutamount": 0,

"delta": 0,

"strikeprice": 0,

"andsome": 0,

"unit": 500,

"callprice": 0,

"expiredate": 0,

"stretchband": 0,

"effectivelever": 0,

"recyclingprice": 0,

"lasttradeday": 0,

"now": 82,

"open": 83,

"high": 83.1500015258789,

"low": 82,

"avg": 82.28791809082031,

"settle": 0,

"lastclose": 83.69999694824219,

"lastsettle": 83.79289245605469,

"curvolume": 1105500,

"volume": 6162493,

"amount": 507105760

}

]

}

}

#### 5.34.4回包字段说明

// 扩展商品基本信息字段

premium; // 溢价

streetratio; // 街货比

streetvolumn; // 街货量

holdamount; // 持股价

splitoffratio; // 换股率

inandoutamount; // 价内/外

delta; // 对充值

strikeprice; // 行使价

andsome; // 大和点

unit; // 每手

callprice; // 收回价

expiredate; // 到期日

stretchband; // 引伸波幅

effectivelever; // 有效杠杆

recyclingprice; // 距回收价

lasttradeday; // 最后交易日

// 行情基本信息字段

now; // 最新

open; // 开盘

high; // 最高

low; // 最低

avg; // 均价

settle; // 结算价

lastclose; // 昨收

lastsettle; // 昨结算

curvolume; // 现手

volume; // 总手

amount; // 总成交额

### 5.35全部存在关联商品的商品(83)

#### 5.35.1请求

{

"reqtype": 83,

"reqid": 1,

"session": "",

"data":{"market": 2002}

}

#### 5.35.2请求字段说明

market 市场

code 商品代码

#### 5.35.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-3-11 11:36:40",

"reqtype": 83,

"reqid": 1,

"data": [

{

"market": 2002,

"code": "00001",

"count": 56

},

{

"market": 2002,

"code": "00002",

"count": 9

}

]

}

#### 5.35.4回包字段说明

market 市场

code 商品代码

count: 个数

### 5.36 获取K线(150)

#### 5.36.1请求

{

"reqtype": 150,

"reqid": 1,

"session": "",

"data": {

"market": 0,

"code": "000001",

"klinetype": 11,

"weight": 0,

"timetype": 2,

"time0": "2017-6-21 12:00:00",

"time1": "2017-6-21 12:00:00",

"count": 10

}

}

#### 5.36.2说明

market 市场代码

code 商品代码

klinetype

enPKLTMinute=1, // 分钟K线

enPKLT5Minute, // 5分钟K线

enPKLT60Minute, // 60分钟K线

enPKLT3Minute, // 3分钟

enPKLT15Minute, // 15分钟

enPKLT30Minute, // 30分钟

enPKLT120Minute, // 120分钟

enPKLTDay = 10, // 日K线

enPKLTWeek, // 周K线

enPKLTMonth = 20, // 月K线

enPKLTSeason = 21, // 季K线

enPKLTYear = 30, // 年K线

weight 0,1,2,

0表示不复权，1表示前复权，2表示后复权

timetype 0,1,2

当TimeType为0时，表示请求time0-time1此时间段的数据

timetime 为1时 time0 往后--> count条

timetime 为2时 count <-- 往前time0 count条

#### 5.36.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-22 6:39:18",

"reqtype": 150,

"reqid": 1,

"data": {

"market": 0,

"code": "000001",

"klinetype": 11,

"weight": 0,

"lastoldtime": "2014-1-2 0:0:0",

"kline": [

["time", "open", "high", "low", "close", "volume", "amount"],

["2017-4-17 0:0:0", 3229.94, 3229.94, 3147.06, 3173.15, 2288.0, 96.0],

["2017-4-24 0:0:0", 3164.24, 3164.24, 3097.33, 3154.65, 3792.0, 28.0]

]

}

}

所有字段的有效值均为两位有效数字，请注意。

### 5.37 获取分笔数据(151)

#### 5.37.1 请求

{

"reqtype": 151,

"reqid": 1,

"session": "",

"data": {

"market": 1,

"code": "600000",

"timetype": 0,

"timevalue0": 100,

"timevalue1": 100,

"day": "2017-7-3 0:00:00"

}

}

#### 5.37.2字段说明

Market： 市场

Code： 商品代码

timetype 0,1,1

timetype 0时timevalue0和timevalue1表示范围

timetype 1时timevalue0-->timevalue1条（即timevalue0这个时间点之后的多少（timevalue1）条记录）

timetype 2时 timevalue1条<--timevalue0（即timevalue0这个时间点之前的多少（timevalue1）条记录）

day 请求哪一天的分笔

注意这里的时间和K线的不一样，timevalue0\*60+day才表示一个时间点

timevalue0相对于day的分钟数

#### 5.37.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-7-5 3:20:56",

"reqtype": 151,

"reqid": 1,

"data": {

"market": 1,

"code": "600000",

"day": "1990-12-16 2:9:51",

"lasthold": 0.0,

"prevclose": 12.55,

"prevsettle": 12.55,

"tick": [

["time", "millisecond", "flag", "price", "volume", "amount"],

["2017-7-3 1:40:0", 55680, 1, 12.60, 1236.0, 15576.0],

["2017-7-3 1:40:2", 55040, 1, 12.60, 2300.0, 28988.0]

]

}

}

#### 5.37.4字段说明

lasthold:昨日持仓，

prevclose：昨收，

prevsettle昨结，

price：价格，

volume：成交量，

amount：成交额或总持仓

### 5.38 获取分时数据(152)

#### 5.38.1请求

{

"reqtype": 152,

"reqid": 1,

"session": "",

"data": {

"market": 3000,

"code": "010002",

"timetype": 0,

"timevalue0": 90,

"timevalue1": 450,

"day": "2017-6-7 0:00:00"

}

}

#### 5.38.2请求字段说明

market 市场

code 商品代码

timetype 0,1,2

timetype 0时timevalue0和timevalue1表示范围

timetype 1时timevalue0-->timevalue1条（即timevalue0这个时间点之后的多少（timevalue1）条记录）

timetype 2时 timevalue1条<--timevalue0（即timevalue0这个时间点之前的多少（timevalue1）条记录）

day 请求哪一天的分时,时间必须填0:00:00

注意这里的时间和K线的不一样，timevalue0\*60+day才表示一个时间点

timevalue0相对于day的分钟数

#### 5.38.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-19 0:54:50",

"reqtype": 152,

"reqid": 1,

"data": {

"market": 3000,

"code": "010002",

"day": "1990-12-16 2:9:51",

"prevclose": 46320.0,

"prevsettle": 46370.0,

"trend": [

["time", "price", "avg", "volume", "amount"],

[0, 46370.0, 46370.0, 0.0, 9672.0]

]

}

}

#### 5.38.4回包字段说明

prevclose：昨收，

Prevsettle：昨结，

price：价格，

volume：成交量，a

mount：成交额或总持仓，

avg：均价

### 5.39 获取商品行情(153)

#### 5.39.1请求

{

"reqtype": 153,

"reqid": 1,

"session": "",

"data": {

"getsyminfo": 1,

"symbol": [{

"market": 3000,

"code": "010023"

}]

}

}

#### 5.39.2字段说明

getsyminfo 1表示获取行情的同时是否获取商品信息

market: 市场编码

code 商品代码

#### 5.39.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-22 5:55:53",

"reqtype": 153,

"reqid": 1,

"data": {

"symbol": [{

"market": 3000,

"code": "010023",

"tradecode": "010023",

"name": "沪铜连三",

"tradetimeid": 2,

"tradetimetype": 1,

"lotsize": 5,

"minprice": "10",

"commoditymarket": -1,

"time": "2017-6-22 5:56:30",

"millisecond": 500,

"now": 46240.0,

"open": 46050.0,

"high": 46430.0,

"low": 46050.0,

"avg": 46271.98,

"settle": 46271.98,

"lastclose": 45750.0,

"lastsettle": 45710.0,

"curvolume": 2.0,

"volume": 11112.0,

"lasthold": 39230.0,

"curhold": -2.0,

"hold": 40370.0,

"volumerate": 30.95,

"buyvolume": 5119.0,

"sellvolume": 5993.0,

"buyprice0": 46230.0,

"buyvol0": 14.0,

"sellprice0": 46250.0,

"sellvol0": 41.0

"listingdate": 11112.0,

"delistingdate": 39230.0,

"orderimbdirection": -2.0,

"orderimbquantity": 40370.0,

"referenceprice": 30.95, // 收盘竞价（CAS）时的参考价

"referencelowerprice": 5119.0, // CAS允许的最低价

"referenceupperprice": 5993.0, // CAS允许的最高价

"vcmstarttime": 46230.0, // 熔断的开始时间

"vcmendtime": 14.0,

"vcmreferenceprice": 46250.0,

"vcmlowerprice": 41.0

"vcmupperprice": 11112.0,

"riselimit": 39230.0,

"falllimit": -2.0,

"minprice": 40370.0,

"commoditymarket": 30.95,

"inversecode": 5119.0,

"fuopttype": 5993.0,

"fuoptexectype": 46230.0,

"begindate": 14.0,

"enddate": 46250.0,

"strikeration": 41.0

"fuoptshowtype": 14.0,

"fuoptshowsubtype": 46250.0,

"tradecount": 41.0

}]

}

}

### 5.40 使用合约号获取商品行情(154)

#### 5.40.1请求

{

"reqtype": 155,

"reqid": 1,

"session": "",

"data": {

"code": "000001"

}

}

#### 5.40.2字段说明

code 商品代码

#### 5.40.3回包

参见获取商品行情

### 5.41 五日分时(158)

#### 5.41.1请求

{

"reqtype": 158,

"reqid": 1,

"session": "",

"data": {

"market": 3000,

"code": "010002",

"timetype": 0,

"timevalue0": 90,

"timevalue1": 450,

"day": "2017-6-7 0:00:00"

}

}

#### 5.41.2请求字段说明

market 市场

code 商品代码

timetype 0,1,2

timetype 0时timevalue0和timevalue1表示范围

timetype 1时timevalue0-->timevalue1条（即timevalue0这个时间点之后的多少（timevalue1）条记录）

timetype 2时 timevalue1条<--timevalue0（即timevalue0这个时间点之前的多少（timevalue1）条记录）

day 请求哪一天的分时,时间必须填0:00:00 (根据这个时间找到五天数据)

注意这里的时间和K线的不一样，timevalue0\*60+day才表示一个时间点

timevalue0相对于day的分钟数

#### 5.41.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-19 0:54:50",

"reqtype": 158,

"reqid": 1,

"data":

[ {"market": 3000,

"code": "010002",

"day": "1990-12-16 2:9:51",

"prevclose": 46320.0,

"prevsettle": 46370.0,

"trend": [

["time", "price", "avg", "volume", "amount"],

[0, 46370.0, 46370.0, 0.0, 9672.0]

]

},

{"market": 3000,

"code": "010002",

"day": "1990-12-17 2:9:51",

"prevclose": 46320.0,

"prevsettle": 46370.0,

"trend": [

["time", "price", "avg", "volume", "amount"],

[0, 46370.0, 46370.0, 0.0, 9672.0]

]

}

]

}

#### 5.41.4回包字段说明

prevclose：昨收，

Prevsettle：昨结，

price：价格，

volume：成交量，a

mount：成交额或总持仓，

avg：均价

### 5.42 通过合约号获取行情,需要完全市场和合约号(155)

#### 5.42.1请求

{

"reqtype": 157,

"reqid": 1,

"session": "",

"data":{

"getsyminfo":1,

"symbol":[{"market":0, "tradecode":"000001"},

{"market":1, "tradecode":"600000"}]}

}

}

#### 5.42.2字段说明

Getsyminfo:1表示获取行情的同时是否获取商品信息

tradecode商品代码

#### 5.42.3回包

同获取商品行情

### 5.43 请求推送(200)

#### 5.43.1请求

{

"reqtype": 200,

"reqid": 1,

"session": "",

"data": [{

"market": 1000,

"code": "000001",

"type": 3,

"language": 0

}, {

"market": 1,

"code": "600000",

"type": 3,

"language": 0

}]

}

#### 5.43.2字段说明

market 市场代码

code 商品代码

type 推送类型 1行情 2分笔 4 经纪人 任意两个可以组合推送

Language 语言类型 0 简体 1 繁体 2 英文

#### 5.43.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-19 0:40:9",

"reqtype": 150,

"reqid": 1,

"data": {

"ret": "1"

}

}

#### 5.43.4说明

本协议只支持websocket请求，收到回包后，会持续收到行情推送。

推送包格式（行情推送包字段与153回包的意义一致，reqtype值为250，分笔推送包字段与151回包的意义一致，reqtype值为251），小数点后有效位不超过两位

#### 5.43.5 推送包

{

"status": 0,

"msg": "",

"servertime": "2017-12-25 6:9:53",

"reqtype": 250,

"reqid": 1,

"data": {

"symbol": [{

"market": 1001,

"code": "000001",

"time": "2017-12-25 6:9:57",

"millisecond": 0,

"avg": 13.60,

"settle": 13.60,

"curvolume": 49496.0,

"volume": 125763584.0,

"amount": 1711012224.0,

"volumerate": 1.18,

"sellvolume": 59640432.0,

"traderate": 0.74,

"moneyin": -86120232.0,

"sellamount": 812736768.0,

"buyvol0": 231104.0,

"buyvol2": 290800.0,

"sellvol0": 3700.0

}]

}

### 5.44 删除推送(201)

#### 5.44.1请求

{

"reqtype": 201,

"reqid": 1,

"session": "",

"data": [{

"market": 1000,

"code": "000001",

"type": 3

}, {

"market": 1,

"code": "600000",

"type": 3

}]

}

}

#### 5.44.2字段说明

market 市场代码

code 商品代码

type 推送类型 1行情 2分笔 3两个都要

#### 5.44.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-19 0:40:9",

"reqtype": 150,

"reqid": 1,

"data": {

"ret": "1"

}

}

#### 5.44.4说明

本协议只支持websocket请求；

### 5.45 同步推送(202)

#### 5.45.1请求

{

"reqtype": 202,

"reqid": 1,

"session": "",

"data": [{

"market": 1000,

"code": "000001",

"type": 3

},

{

"market": 1,

"code": "600000",

"type": 3

}]

}

#### 5.45.2字段说明

market 市场代码

code 商品代码

type 推送类型 1行情 2分笔 3两个都要

#### 5.45.3回包

{

"status": 0,

"msg": "",

"servertime": "2017-6-19 0:40:9",

"reqtype": 150,

"reqid": 1,

"data": {

"ret": "1"

}

}

#### 5.45.4说明

本协议只支持websocket请求；

### 5.46 上传用户私有数据(400)

#### 5.46.1请求

{

"reqtype": 400,

"reqid": 1,

"session": "",

"datatype": ,

"datalen": 1,

"data": ""

}

#### 5.46.2字段说明

Datatype 数据类型

Datalen数据长度

Data数据内容

#### 5.46.3回包

{

"status": 0,

"msg": "",

"reqtype": 400,

"Ret": 0

}

#### 5.46.4说明

### 5.47 上传机构额外数据(401)

#### 5.47.1请求

{

"reqtype": 401,

"reqid": 1,

"session": "",

"datatype": ,

"datalen": 1,

"orgid": 1,

"data": ""

}

#### 5.47.2字段说明

Datatype 数据类型

Datalen数据长度

Data数据内容

Orgid 机构id

#### 5.47.3回包

{

"status": 0,

"msg": "",

"reqtype": 401,

"Ret": 0

}

#### 5.47.4说明

### 5.48 下载用户私有数据(402)

#### 5.48.1请求

{

"reqtype": 402,

"reqid": 1,

"session": "",

"datatype": ,

}

#### 5.48.2字段说明

#### 5.48.3回包

{

"status": 0,

"msg": "",

"reqtype": 402,

"datatype": ,

"data": ""

}

#### 5.48.4说明

### 5.49 下载机构额外数据(403)

#### 5.49.1请求

{

"reqtype": 403,

"reqid": 1,

"session": "",

"datatype": ,

"orgid": 1

}

#### 5.49.2字段说明

#### 5.49.3回包

{

"status": 0,

"msg": "",

"reqtype": 403,

"datatype": ,

"orgid": 1,

"data": ""

}

#### 5.49.4说明

### 5.50 获得上涨下跌家数（板块）(69)

#### 5.50.1请求

{

"reqtype": 69,

"reqid": 1,

"session": "",

"data": [{"block":1},{"block":7700}]

}

#### 5.50.2请求字段说明

#### 5.50.3回包

{

"status":0,

"msg":"",

"servertime":"2017-6-22 3:52:11",

"reqtype":65,

"reqid":1,

"data":[{" block”:1,”up”:23,”down”:987},{" block”:2,”up”:537,”down”:86}]

}

### 5.51 批量查询商品市场(84)

#### 5.51.1请求

{

"reqtype":84,

"reqid":1,

"session":"",

"data": [{"code":"00001"},{"code":"00002"},{"code":"00003"},{"code":"00004"}]

}

#### 5.51.2说明

#### 5.51.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-8-8 10:22:32",

"reqtype": 84,

"reqid": 1,

"data": {

"symbol": [{"market": 2002,"code": "00001"},

{"market": 2002,"code": "00002"},

{"market": 2002,"code": "00003"},{"market": 2002,"code": "00004"}]

}

}

### 5.52 商品模糊匹配（市场和板块）(85)

#### 5.52.1请求

{

"reqtype":85,

"reqid":1,

"session":"",

"data": {

"market":[{"marketid":2002},{"marketid":8400}],

"code":"00001",

"tradecode":"00001",

"name":"00001"

}

}

#### 5.52.2字段说明

Getquote:是否需要行情数据（值为0,1），默认为0不需要。

如果id中有一个为负数，则会匹配所有市场或者板块；

iType: 1表示码表里面的市场id，3表示板块里面的板块id

实现模糊匹配，请将code、tradecode、name三个字段填相同的值，支持汉字、全拼、拼音首字母的匹配。

#### 5.52.3回包

{

"status":0,

"msg":"",

"servertime":"2017-7-1 3:26:45",

"reqtype":58,

"reqid":1,

"data":{

"symbol":[{

"market":2002,

"code":"00001",

"name":"长和",

"flag":"8600"

}

]}

}

#### 5.52.4 回包字段说明

Flag：8300:港股通(滬) 8400:沪股通 8500: 深股通 8600:港股通(深)

Flag是一个8300,8600这种以逗号分割的json字符串，如果flag为空，这说明该商品只是属于market市场，如果flag不为空，这就是该商品即属于market市场也属于下面板块文件。

### 5.53 经济人信息查询(86)

#### 5.53.1请求

{

"reqtype": 86,

"reqid": 1,

"session": "",

"data":{"bigmarket": 2,"market": 2002}

}

#### 5.53.2字段说明

Bigmarket:2 大市场，目前经纪人都是属于港股的，默认传2就行

Market：2002 默认2002市场就行

#### 5.53.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-9-4 9:30:38",

"reqtype": 86,

"reqid": 1,

"data": {

"brokerinfo": [

{

"brokerid": 1077,

"brokerseat": "2840,2842,2846,2847,2848,2850,2851,2857,4160,4164,4165,4167,4168,4169,9713,9723",

"brokername": "麦格理",

"brokerfname": "麦格理资本股份有限公司",

"brokerwname": "麥格理",

"brokerwfname": "麥格理資本股份有限公司",

"brokerename": "Macquarie Capital",

"brokerefname": "Macquarie Capital Limited"

},

{

"brokerid": 1080,

"brokerseat": "2220,2221,2222,2227,2228,2229",

"brokername": "鼎珮证券",

"brokerfname": "鼎珮证券有限公司",

"brokerwname": "鼎珮證券",

"brokerwfname": "鼎珮證券有限公司",

"brokerename": "VMS Securities",

"brokerefname": "VMS Securities Limited"

},

}

#### 5.53.4 回包字段说明

Brokerid：经纪人代码

Brokerseat：经纪人席位

Brokername：经纪人简称

Brokerfname：经纪人全称

brokerwname: 名称繁体

brokerwfname: 名称繁体全称

brokerename:英文名称

brokerefname:英文名称全称

### 5.54 正股的涡轮牛熊期权查询(87)

#### 5.54.1请求

{

"reqtype": 87,

"reqid": 1,

"session": "",

"data":{"contract": 300,"volume": 50000,"code": "00700","flag": 1}

}

#### 5.54.2字段说明

contract:期权合约数

Volume：成交量（对应涡轮牛熊的成交量）

Code:正股代码

备注：填contract和volume可以查询出正股下涡轮牛熊成交量大于xxx的数量有多少。默认是300和1000000.

Flag: 需不需要显示商品数组，1是需要显示 0 不需要 默认不需要

#### 5.54.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-11-25 7:20:40",

"reqtype": 87,

"reqid": 1,

"data": {

"Turbine": {

"TurbineVaildCount": 5575,

"TurbineActionCount": 14,

"symbol": [

{

"market": 2003,

"code": "11281",

"zgcode": "00001",

"sztradecode": "11281",

"name": "长和法兴零二购Ａ",

"fuopttype": 0,

"strikeprice": 74.93000030517578,

"volume": 140000.0

}

},

"Ushikuma": {

"UshikumaVaildCount": 15559,

"UshikumaActionCount": 0,

"symbol": []

},

"Option": {

"OptionVaildCount": 58968,

"OptionActionCount": 1,

"symbol": [

{

"market": 2007,

"code": ".CKH70.00W9",

"zgcode": "00001",

"sztradecode": "CKH70.00W9",

"name": "CKH70.00W9",

"fuopttype": 0,

"strikeprice": 70.0,

"volume": 401.0

}

]

}

}

}

#### 5.54.4 回包字段说明

Turbine：涡轮

TurbineVaildCount：2003市场下所有有效涡轮

TurbineActionCount：对应正股活跃的涡轮

symbol：涡轮商品数组

Ushikuma: 牛熊

UshikumaVaildCount: 2004市场下所有有效牛熊

UshikumaActionCount:对应正股活跃的牛熊

Option：期权

OptionVaildCount:2007市场下所有有效期权

OptionActionCount：对应正股下活跃期权

### 5.55 霸才精选查询(88)

#### 5.55.1请求

{

"reqtype": 88,

"reqid": 1,

"session": "",

"data":{"market": "2002,2031","type": 0}

}

#### 5.55.2字段说明

Market:默认给2002,2031就行，目前没有用到该字段

Type: E\_IndexSortType

enum E\_IndexSortType

{

EISTZCEMA = 0, // 主力吸筹穿过100线+EMA金叉（周期10、30） EISTZXEMA, // 主力出货穿过100线+EMA死叉（周期10、30）

EISTJDRSI, // 狙击手（多）+RSI周期为9，值<30

EISTJKRSI, // 狙击手（空）+RSI周期为9，值>70

};

#### 5.55.3回包

{

"status": 0,

"msg": "",

"servertime": "2019-11-27 2:53:28",

"reqtype": 88,

"reqid": 1,

"data": {

"totalcount": 12,

"type": 0,

"symbol": [

{

"market": 2002,

"code": "00059"

}

]

}

}

#### 5.55.4 回包字段说明

totalcount：一共多少只股票

type：请求的类型 E\_IndexSortType