行情接口接入说明

(v1.0.2.3)

目录

1. 前言

阅读文档前确认您已经在阿里云 API 市场购买接口获得授权。

上海锃然网络有限责任公司致力于互联网应用、全球行情数据、交易系统、数据分析智能化工具和服务的企业。 我们提供全球主流指数行情、沪深 A 股、国内期货、外盘期货、CFD 合约，货币利率，加密货币等数据，推送及查询服务。同时我们提供对应所有这些市场、品种合约的交易系统。

2. API 行情接口使用

接口数据包括沪深 A 股行情，国内期货主力合约行情，外盘期货主力合约行情，全球主流货币对交易行情，全球主流 CFD 合约行情，加密货币。

建议行情数据主要用来数据分析和模型模拟推算，数据仅供参考，不构成任何投资建议，用户据此操作造成的风险及后果均由用户自行负责。数据资料均来源于互联网，版权归原作者所有。

同时我们提供对应所有这些市场、品种合约的交易系统，如需请联系接洽。

a) 品种

部分 Symbol 对照表			
完整 Symbol	市场	品种代码	名称
沪深 A 股			
SH000001			上证指数
SH000002			A 股指数
SH000003			B 股指数
SH000004			工业指数
SH000005			商业指数
SH000006			地产指数
SH000007			公用指数
SH000008			综合指数
SZ399305			基金指数

.....上海 A 股近 2 千代码			
SH600000	上海 A 股	600000	浦发银行
..... 深圳 A 股近 2 千代码			
SZ000001	深圳 A 股	000001	平安银行
国内期货			
包含主力合约和各品种加权指数(代码后加大写字母 I),s 如沪铜指数 SHFECUI			
CFFEXIC	中金交易所	IC	中证 500 股指
CFFEXIF		IF	沪深 300 股指
CFFEXIH		IH	上证 50 股指
CFFEXT		T	10 年国债
CFFEXTF		TF	5 年国债
CZCEAP	郑州交易所	AP	苹果
CZCECF		CF	棉花
CZCECY		CY	棉纱
CZCEFG		FG	玻璃
CZCEJR0		JR	粳稻
CZCELR0		LR	晚籼稻
CZCEMA0		MA	甲醇
CZCEOI0		OI	菜籽油
CZCERI0		RI	早籼稻
CZCERM0		RM	菜籽粕
CZCERS		RS	油菜籽
CZCESF		SF	硅铁
CZCESM		SM	锰硅
CZCESR		SR	白糖
CZCETA		TA	PTA
CZCEWH		WH	强麦
CZCEZC		ZC	郑煤
DCEA	大连交易所	A	豆
DCEB		B	豆
DCEBB		BB	胶合板
DCEC		C	玉米
DCECS		CS	玉米淀粉
DCEI		I	铁矿石
DCEJ		J	焦炭
DCEJD		JD	鸡蛋
DCEJM		JM	焦煤
DCEL		L	聚乙烯
DCEM		M	豆粕
DCEP		P	棕榈油

DCEPP		PP	聚丙烯
DCEV		V	聚氯乙烯
DCEY		Y	豆油
INESC	上海能源交易所	SC	原油
SHFEAG	上海期货交易所	AG	银
SHFEAL		AL	铝
SHFEAU		AU	黄金
SHFEBU		BU	沥青
SHFECU		CU	铜
SHFEHC		HC	热轧卷板
SHFENI		NI	镍
SHFEPB		PB	铅
SHFERB		RB	螺纹
SHFERU		RU	橡胶
SHFESN		SN	锡
SHFEZN		ZN	锌
国际期货			
包括股指期货,能源期货,外汇期货,商品期货			
APEXPF	APEX	PF	棕榈油
CBOTC	CBOT	C	玉米
COMEXGC	COMEX	GC	黄金
COMEXHG	COMEX	HG	铜
COMEXSI	COMEX	SI	白银
ICEB	ICE	B	伦敦布伦特原油
LMECA	LME	CA	LME 铜
NYMEXCL	NYMEX	CL	原油
NYMEXNG	NYMEX	NG	天然气
NYMEXPA	NYMEX	PA	钯
NYMEXPL	NYMEX	PL	铂
CMEBTC	CME	BTC	比特币 CME
CMEDX	CME	DX	美元指数
CMEEC	CME	EC	欧元/美元
CMEGBP	CME	GBP	英镑/美元
CMEJY	CME	JY	日元/美元
ASXAP	ASX	AP	澳大利亚 200
CBOTYM	CBOT	YM	道琼斯 30
CFFEXIF	CFFEX	IF	沪深 300
CMEES	CME	ES	美国标普 500
CMENQ	CME	NQ	纳斯达克 100
EUREXDAX	EUREX	DAX	德国 DAX 指数
HKEXHSI	HKEX	HSI	恒生指数

HKEXMCH	HKEX	MCH	小型 H 股指数
HKEXMHI	HKEX	MHI	小型恒生指数
LIFFEZ	LIFFE	Z	英国 100
SGXSFC	SGX	SFC	富时中国 A50 指数
SIMEXSG	SIMEX	SG	新加坡 MSCI
外汇差价合约			
包括其他交叉汇率			
AUDUSD	——	AUDUSD	AUDUSD
EURUSD	——	EURUSD	EURUSD
GBPUSD	——	GBPUSD	GBPUSD
USDCAD	——	USDCAD	USDCAD
USDCHF	——	USDCHF	USDCHF
USDCNH	——	USDCNH	USDCNH
USDJPY	——	USDJPY	USDJPY
XAGUSD	——	XAGUSD	XAGUSD
XAUUSD	——	XAUUSD	XAUUSD
EURJPY	——	EURJPY	EURJPY
DINIW		DINIW	美元指数
外汇差价合约			
包括其他 CFD 合约			
AUS200	——		澳大利亚标准普尔
COPPER	——		铜
CORN	——		玉米
COTTON	——		棉花
ESTX50	——		道琼斯欧洲指数
FRA40	——		法国指数
GER30	——		德国GER30指数
HK50	——		恒指
HTG_OIL	——		燃油期货
JPN225	——		日经指数
NAS100	——		美国纳斯达克
PALLAD	——		钯期货
PLAT	——		铂期货
SOYBEAN	——		大豆
SPX500	——		标普
SUGAR	——		食糖
UK_OIL	——		布伦特原油
UK100	——		英国富时指数
US_NATG	——		美国天然气

US_OIL	——		美国原油
US30	——		道琼斯工业平均指数
WHEAT	——		小麦
.....			
加密数字货币			
包括其他数字货币			
ADA	——		艾达币
BCH	——		比特币现金
BTC	——		比特币
EOS	——		EOS
ETH	——		以太坊
LTC	——		莱特币
MIOTA	——		埃欧塔
USDT	——		泰达币
XLM	——		恒星币
XRP	——		瑞波币

3. 行情查询接口接入

a) 实时行情协议

1、购买 api 接口获得授权。

2、在阿里云商品详情中找到接口协议（即 Api 接口）,仔细阅读 api 接入步骤和内容。

3、实时接口输入输出协议

输入参数			
名称	类型	长度	备注
Symbol	String	32	字母开头，只包含字母和数字

范例：http://xxx/xxx?symbol=shfeau

输出 Json 结构			
字段	类型	长度	备注
Code	int		字母开头，只包含字母和数字 Code>=0 为查询并返回成功
Msg	String		返回描述，一般失败才有描述
Obj	JsonObj		返回 Json 结构，结构如下

各市场品种返回数据有差别，部分字段没有数据是正常的

输出数据 Json 结构			
字段	类型	长度	备注
M	string		市场代码
S	string		品种
C	string		代码
N	String		中文名
O	decimal		开盘价
H	decimal		高
L	decimal		低
P	decimal		当前价
YC	decimal		昨收价
A	decimal		总额
V	Long		总量
Tick	Long		时间戳
B1	decimal		买一
B2	decimal		买二
B3	decimal		买三
B4	decimal		买四
B5	decimal		买五
S1	decimal		卖一
S2	decimal		卖二
S3	decimal		卖三
S4	decimal		卖四
S5	decimal		卖五
B1V	Long		买一量
B2V	Long		买二量
B3V	Long		买三量
B4V	Long		买四量
B5V	Long		买五量
S1V	Long		卖一量
S2V	Long		卖二量
S3V	Long		卖三量
S4V	Long		卖四量
S5V	Long		卖五量
HS	Decimal		换手率（如果有）
VF	Decimal		振幅（如果有）
LS	Decimal		流通市值（如果有）
ZS	Decimal		总市值（如果有）
SY	Decimal		市盈率（如果有）
SJ	Decimal		市净率（如果有）
ZF	Decimal		涨幅（如果有）

HD	Long		持仓（如果有）
JS	Decimal		结算价（如果有）

成功范例:

{"Code":0,"Msg":"",
"Obj":{
"B1":271.100, --买一
"B1V":129, --买一量
"B2":0, --买二
"B2V":0,
"B3":0, --买三
"B3V":0,
"B4":0, --买四
"B4V":0,
"B5":0, --买五
"B5V":0,
"S1":271.150, --卖一
"S1V":20, --卖一量
"S2":0, --卖二
"S2V":0,
"S3":0, --卖三
"S3V":0,
"S4":0, --卖四
"S4V":0,
"S5":0, --卖五
"S5V":0,
"ZT":280.85, --涨停价
"DT":259.19, --跌停价
"O":270.39, --今开
"H":271.69, --最高
"L":270.14, --最低
"YC":270.55, --昨收
"A":35513202100.0,--交易额
"V":130972, --交易量
"P":271.14, --当前价
"Tick":1529911046, --标准时间戳
"N":"黄金主连", --品种名
"M":"SHFE", --市场
"S":"AU", --品种代码
"C":"", --编号
"SY":0, --市盈率(如果有)
"SJ":0, --市净率(如果有)
"VF":0, --振幅(如果有)

```
        "ZF":0,          --涨幅(如果有)
        "HS":0,          --换手率(如果有)
        "LS":0,          --流通市值(如果有)
        "ZS":0           --总市值(如果有)
        "JS":0           --结算价(如果有)
        "HD":0           --持仓(如果有)
    }
}
```

失败范例

```
{
   "Code":-100,
     "Msg":"symbol not exists"
}
```

b) 批量行情协议

输入输出协议

输入参数			
名称	类型	长度	备注
Symbols	String		品种代码组，逗号分隔，最多一次 20

范例：http://xxx/xxx?symbols=shfeau

输出 Json 结构			
字段	类型	长度	备注
Code	int		字母开头，只包含字母和数字 Code>=0 为查询并返回成功
Msg	String		返回描述，一般失败才有描述
Obj	JsonArray		返回 Json 结构数组

成功范例：

```
{"Code":0,"Msg":"",
    "Obj":[{
    "B1":271.100,        --买一
    "B1V":129,           --买一量
    "B2":0,              --买二
    "B2V":0,
    "B3":0,              --买三
    "B3V":0,
```

```
        "B4":0,                  --买四
        "B4V":0,
        "B5":0,                  --买五
        "B5V":0,
        "S1":271.150,            --卖一
        "S1V":20,                --卖一量
        "S2":0,                  --卖二
        "S2V":0,
        "S3":0,                  --卖三
        "S3V":0,
        "S4":0,                  --卖四
        "S4V":0,
        "S5":0,                  --卖五
        "S5V":0,
        "ZT":280.85,                --涨停价
        "DT":259.19,                --跌停价
        "O":270.39,                 --今开
        "H":271.69,                 --最高
        "L":270.14,                 --最低
        "YC":270.55,                --昨收
        "A":35513202100.0,          --交易额
        "V":130972,                 --交易量
        "P":271.14,                 --当前价
        "Tick":1529911046,          --标准时间戳
        "N":"黄金主连",              --品种名
        "M":"SHFE",                 --市场
        "S":"AU",                   --品种代码
        "C":"",                     --编号
          "SY":0,                   --市盈率(如果有)
          "SJ":0,           --市净率(如果有)
          "VF":0,           --振幅(如果有)
          "ZF":0,           --涨幅(如果有)
          "HS":0,           --换手率(如果有)
          "LS":0,           --流通市值(如果有)
          "ZS":0            --总市值(如果有)
          "JS":0            --结算价(如果有)
          "HD":0            --持仓(如果有)
       },{}
         ,{}...
      ]
}
```

失败范例

```
{
   "Code":-100,
     "Msg":"symbol not exists"
}
```

c) 行情列表查询协议/分页

输入输出协议

输入参数			
名称	类型	长度	备注
Rout	String	32	类型，GBFSB/外汇，GBCFD/CFD，GBFT/国际期货，CNFT/国内期货，CNST/沪深 A 股，GBDC/数字货币
Where	String	32	以关键字开头的查询
p	Int		页码
ps	Int		每页行数
Sort	String	32	排序字段，可取 S,P,ZS,SY,ZF
SortType	Int		0 升序，1 降序

范例：http://xxx/xxx?p=1&ps=10&where=&rout=&sort=&sorttype=

输出 Json 结构			
字段	类型	长度	备注
Code	int		字母开头，只包含字母和数字 Code>=0 为查询并返回成功
Msg	String		返回描述，一般失败才有描述
Obj	JsonArray		返回 Json 结构数组

d) K 线历史行情协议

K 线历史数据接口输入输出协议

输入参数			
名称	类型	长度	备注
Symbol	String	32	字母开头，只包含字母和数字
Period	String	32	1M,5M,15M,30Ms1H,4H,D,W,M
Pnum	Int		数量

范例：http://xxx/xxx?period=1M&pnum=10&symbol=shfeau

输出 Json 结构			
字段	类型	长度	备注
Code	int		字母开头，只包含字母和数字 Code>=0 为查询并返回成功
Msg	String		返回描述，一般失败才有描述
Obj	JsonObj		返回 Json 结构，结构如下

各市场品种返回数据有差别，部分字段没有数据是正常的

输出数据 Json 结构			
字段	类型	长度	备注
O	decimal		开盘价
H	decimal		高
L	decimal		低
C	decimal		收
A	decimal		总额
V	Long		总量
Tick	Long		时间戳

成功范例：


```
{
  "Code": 0,
  "Msg": "",
  "Obj": [
    {
      "C": 16.47,
      "Tick": 1529856000,
      "O": 16.59,
      "H": 16.62,
      "L": 16.4,
      "YC": 0,
      "A": 352461552,
      "V": 21345191
    },
    {
      "C": 16.23,
      "Tick": 1529942400,
      "O": 16.4,
      "H": 16.54,
      "L": 16.2,
      "YC": 0,
      "A": 310260023,
      "V": 18978862
    }
```

```
    ]
}
```

失败范例

```
{
   "Code":-100,
     "Msg":"symbol not exists"
}
```

e) 国内期货交易信息查询（手续费/合约数等）

接口输入输出协议

输入参数			
名称	类型	长度	备注
M	String	32	市场代码 SHFE,DCE,CZCE,CFFEX,INE
S	String	32	品种标识

范例：http://xxx/xxx?m=SHFE&s=au

成功范例：

```
{
  "Code": 0,
  "Msg": "",
  "Obj": [
     {"S":"sc"          //标识
     ,"SN":"原油",     //名
     "C":"1907",       //代码
     "OF":20,          //开仓手续费，>1 时单位为元，按手数算；小于 1 时按交易金额算
     "CF":20,          //平仓手续费
     "CF0":0,          //平今手续费
     "OF0":20,         //
     "JGF":0,          //交割费用
     "P":462.8,        //收盘价
     "BR":0.07,        //买多保证金率
     "SR":0.07,        //卖空保证金率
     "BR0":0.07,       //投机买多保证金率
     "SR0":0.07,             //投机卖空保证金率
```

```
        "STDV":1000,  //标准手合约数
        "PT":0.1,         //价格变动单位
        "ISM":0,          //是否主力合约
        "STP":0,          //结算价
        "UTimeL":1532479134    //数据获取时间
        }
        ,{...}
        ]
}
```

失败范例

```
    {
       "Code":-100,
         "Msg":"fail"
    }
```

f) 阿里云接入代码范例

参考阿里云应用接入的对应开发语言的接口接入范例代码，修改授权信息，整合进您的应用逻辑即可完成。

Curl Demo

```
curl -i --get --include 'http://xxx?symbol=USDCHN'   -H 'Authorization:APPCODE 你自己的 AppCode'
```

Java Demo

```
public static void main(String[] args) {

String host = "http://xxxx";
String path = "/xxx";
String method = "GET";
String appcode = "你自己的 AppCode";
Map<String, String> headers = new HashMap<String, String>();
//最后在 header 中的格式(中间是英文空格)为 Authorization:APPCODE
73fe94948385f570e3c139105
headers.put("Authorization", "APPCODE " + appcode);
Map<String, String> querys = new HashMap<String, String>();
querys.put("symbol", "USDCHN");
```

```
try {
  /**
  * 重要提示如下:
  * HttpUtils 请从
  *
github.com/aliyun/api-gateway-demo-sign-java/blob/master/src/main/java/com/aliyun/api/gateway/d
l/HttpUtils.java
  * 下载
  *
  * 相应的依赖请参照
  * https://github.com/aliyun/api-gateway-demo-sign-java/blob/master/pom.xml
  */
  HttpResponse response = HttpUtils.doGet(host, path, method, headers, querys);
  System.out.println(response.toString());
  //获取 response 的 body
  //System.out.println(EntityUtils.toString(response.getEntity()));
} catch (Exception e) {
  e.printStackTrace();
}
```

Python Demo

```
import urllib, urllib2, sys



host = 'http://xxx'
path = '/xxx'
method = 'GET'
appcode = '你自己的 AppCode'
querys = 'symbol=USDCHN'
bodys = {}
url = host + path + '?' + querys

request = urllib2.Request(url)
request.add_header('Authorization', 'APPCODE ' + appcode)
response = urllib2.urlopen(request)
content = response.read()
if (content):
    print(content)
```

C# Demo

```
//using System.IO;
//using System.Text;
//using System.Net;
//using System.Net.Security;
//using System.Security.Cryptography.X509Certificates;

        private const String host = "http://xxxx";
        private const String path = "/xxxx";
        private const String method = "GET";
        private const String appcode = "你自己的 AppCode";

        static void Main(string[] args)
        {
            String querys = "symbol=USDCHN";
            String bodys = "";
            String url = host + path;
            HttpWebRequest httpRequest = null;
            HttpWebResponse httpResponse = null;

            if (0 < querys.Length)
            {
                url = url + "?" + querys;
            }

            if (host.Contains("https://"))
            {
                ServicePointManager.ServerCertificateValidationCallback = new
RemoteCertificateValidationCallback(CheckValidationResult);
                httpRequest = (HttpWebRequest)WebRequest.CreateDefault(new Uri(url));
            }
            else
            {
                httpRequest = (HttpWebRequest)WebRequest.Create(url);
            }
            httpRequest.Method = method;
            httpRequest.Headers.Add("Authorization", "APPCODE " + appcode);
            if (0 < bodys.Length)
            {
                byte[] data = Encoding.UTF8.GetBytes(bodys);
                using (Stream stream = httpRequest.GetRequestStream())
                {
                    stream.Write(data, 0, data.Length);
                }
            }
```

```
            try
            {
                httpResponse = (HttpWebResponse)httpRequest.GetResponse();
            }
            catch (WebException ex)
            {
                httpResponse = (HttpWebResponse)ex.Response;
            }

            Console.WriteLine(httpResponse.StatusCode);
            Console.WriteLine(httpResponse.Method);
            Console.WriteLine(httpResponse.Headers);
            Stream st = httpResponse.GetResponseStream();
            StreamReader reader = new StreamReader(st, Encoding.GetEncoding("utf-8"));
            Console.WriteLine(reader.ReadToEnd());
            Console.WriteLine("\n");

        }

        public static bool CheckValidationResult(object sender, X509Certificate certificate, X509Chain
chain, SslPolicyErrors errors)
        {
            return true;
        }
```

ObjectC Demo

```
NSString *appcode = @"你自己的 AppCode";
NSString *host = @"http://xxxx";
NSString *path = @"/xxxx";
NSString *method = @"GET";
NSString *querys = @"?symbol=USDCHN";
NSString *url = [NSString stringWithFormat:@"%@%@%@",  host,  path , querys];
NSString *bodys = @"";

NSMutableURLRequest *request = [NSMutableURLRequest requestWithURL:[NSURL URLWithString:
url]  cachePolicy:1  timeoutInterval:  5];
request.HTTPMethod  =  method;
[request addValue:  [NSString  stringWithFormat:@"APPCODE %@" ,  appcode]
forHTTPHeaderField:  @"Authorization"];
NSURLSession *requestSession = [NSURLSession
sessionWithConfiguration:[NSURLSessionConfiguration defaultSessionConfiguration]];
NSURLSessionDataTask *task = [requestSession dataTaskWithRequest:request
    completionHandler:^(NSData * _Nullable body , NSURLResponse * _Nullable response, NSError *
```

```
_Nullable error) {
    NSLog(@"Response object: %@" , response);
    NSString *bodyString = [[NSString alloc] initWithData:body encoding:NSUTF8StringEncoding];

    //打印应答中的 body
    NSLog(@"Response body: %@" , bodyString);
    }];

[task resume];
```

Php Demo

```
<?php
    $host = "http://xxxx";
    $path = "/xxxx";
    $method = "GET";
    $appcode = "你自己的 AppCode";
    $headers = array();
    array_push($headers, "Authorization:APPCODE " . $appcode);
    $querys = "symbol=USDCHN";
    $bodys = "";
    $url = $host . $path . "?" . $querys;

    $curl = curl_init();
    curl_setopt($curl, CURLOPT_CUSTOMREQUEST, $method);
    curl_setopt($curl, CURLOPT_URL, $url);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($curl, CURLOPT_FAILONERROR, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_HEADER, true);
    if (1 == strpos("$".$host, "https://"))
    {
        curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
        curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, false);
    }
    var_dump(curl_exec($curl));
?>
```

4. 实时行情推送接入

接入前请先购买 api 接口获得授权，并联系我们获取 Token。

第一步：根据授权 token 获取连接 ip 地址

get http://map.konpn.com:10002/query/mqttlocation/0/token
注： token：授权 token
返回

{"Code":0,"Msg":"","Obj":
{"IP":"115.28.26.130","MQPORT":10099,"WSPORT":10199,"HTTPPORT":10094}}

Code=0 表示返回成功
Obj 结构字段分别为：IP: 该授权的连接 ip,
MQPORT :mq 连接端口,
WSPORT :websocket 端口,
HTTPPORT :httpget 端口

a) WebSocket 协议

Demo：http://www.zengr.net/demo/wsdemo.html

1、连接

WebSocket(ws://授权 ip:端口/connect/formurlencoded/token);

注：建立连接，请求返回数据结构可以为 formurlencoded，json，text
授权 token

2、按品种订阅

Ws.send(/sub/EURUSD,SH000001,JPN225,USDCNH,品种代码)
注：订阅行情，根据授权如果订阅数为 N，N 个以后的订阅会将第 1 个订阅取消，依次类推

3、按市场订阅

Ws.send(/subrout/GBDC,CNST，市场代码)
注：订阅整个市场行情

4、取消订阅

Ws.send(/unsub/EURUSD,品种代码,CNST,市场代码)
注：取消订阅行情

5、设置订阅字段

Ws.send(/fields/B1,S1,B2,S2,A,V)
注：设置订阅的字段，选用功能。不设置时默认数据为基本价格数据，可订阅数据字段请参考行情结构

6、发送心跳

Ws.send(/heartbeat/ok)
注：30 秒要发送心跳包，保持推送连接

b) Http 推送协议

Demo：http://www.zengr.net/demo/httpdemo.html

1、请求下发

get http://授权 ip:端口
/mqhttp/connect?serializetype=formurlencoded&token=testpwd&hookurl=http%3a%2f%2fwww.test.com%2fsdf-123.html
注：serializetype：请求下发数据结构，formurlencoded/json/text 中之一
token：授权 token 为 testpwd
hookurl：post 下发调用接口，urlencode 编码

2、按品种订阅行情

get http://授权 ip:端口
/mqhttp/sub?token=testpwd&symbols=EURUSD,SH000001,JPN225,USDCNH,品种代码
注：token：授权 token
symbols：订阅行情，根据授权如果订阅数为 N，N 个以后的订阅会将第 1 个订阅取消，依次类推

3、按市场订阅

get http://授权 ip:端口/mqhttp/subrout?token=testpwd&routs=GBDC,市场代码
注：订阅整个市场行情
token：授权 token
routs: GBDC,CNST，市场代码

4、取消订阅

get http://授权 ip:端口
/mqhttp/unsub?token=testpwd&symbols=EURUSD,SH000001,JPN225,USDCNH
注：取消订阅品种
token：授权 token
symbols：取消订阅的代码

get http://授权 ip:端口/mqhttp/unsubrout?token=testpwd&routs=GBDC,GBFT
注：取消订阅市场
token：授权 token
routs：取消订阅的市场

5、设置返回字段（可选）

get http://授权 ip:端口/mqhttp/fields?token=testpwd&fields=B1,S1,B2,S2,A,V
注：token：授权 token

fields：数据字段，不设置时默认数据为基本价格数据，可订阅数据字段请参考行情结构

6、发送心跳

get http://授权 ip:端口/mqhttp/heartbeat?token=testpwd

注：token：授权 token

30 秒需要发送心跳包，保持连接

c) MQtt 协议

Demo：http://www.zengr.net/demo/mqttdemo.html

1、连接协议

UserName:客户端 ip，UserPassword: 授权 token

ServerIp: 授权 ip 和 端口

ClientId：授权 token

2、按品种订阅

发送 Topic: cmd/sub

发送 Payload: 品种代码，逗号分隔

等级：AtMostOnce

3、按市场订阅

发送 Topic: cmd/subrout

发送 Payload: 市场代码，逗号分隔

等级：AtMostOnce

4、取消订阅

发送 Topic: cmd/unsub

发送 Payload: 市场代码，品种代码逗号分隔

等级：AtMostOnce

5、发送心跳

Topic：cmd/heartbeat
Payload：ok
30 秒要发送心跳包，保持连接

5. 使用场景

a) 模拟交易比赛：机构或团体的互动活动

b) 教育培训机构：实现学生的学习提高和选拔

c) 数据分析和交易模型推算：数据分析和模型测试等等

d) 其他

6. 行情分类

1、沪深 A 股

包含沪深 A 股市场，主板，创业板，中小板股票的实时行情

2、国内期货

包含上海能源交易所，上海期货交易所，郑州商品交易所，大连商品交易所，中金交易所的在线上百个品种行情，注意：只提供主力合约行情。如您有需要可以另外给我们提需求

3、外盘期货

包含全球股指期货，外汇期货，金属，能源，农产品期货等。等数十个在线品种行情，注意：只提供主力合约行情。如您有需要可以另外给我们提需求

4、点差交易合约（外汇交易）

包括主流直盘，交叉盘，黄金白银等货币对合约

5、CFD 合约

包括全球主流指数合约，石油，棉花等合约

6、加密货币

包括全球主流热门加密数字货币

7、香港美国股票

包括香港/美国热门、主板股票