

# 蓝目数字货币共享数据网站 API 规范



**蓝目数据工作室**

<http://www.blueye.info>

杭州展赛信息技术有限公司

## 目录

一、基本需求 .....	3
二、WebSocket 方式行情数据目录 .....	4
1.消息格式定义 .....	4
2.消息体定义 .....	5
3.登录消息 .....	6
4.行情消息 .....	7
4.1 Ticker 快照行情 .....	7
4.2 Depth 深度行情 .....	7
三、Restful 模式下的行情数据目录 .....	8
1. restful 模式下的 Kline K 线行情 .....	8
2. restful 模式下的 ticker 行情 .....	9
3. restful 模式下的 depth 行情 .....	9
四. Restful 模式下的交易消息 .....	10
1.下单 .....	10
2.撤单 .....	11
3.查单 .....	12
五、历史数据文件下载 .....	14

## 一、基本需求

本文档是提供给专业的量化交易员，以及相关的 APP 和网站开发人员使用。

提供基于目前数字货币的各类数据的客户端访问接口及 API 调用规范。目前已经支持的数据和指令包含：

- 1，实时行情数据 Ticker;
- 2，深度行情数据 Depth;
- 3，历史 K 线数据 Kline ；
- 4，商品委托买卖 Order ；
- 5，已发委托撤单 Cancel ；
- 6，已发委托查询 Order\_info。

接口规范包含：

- 1，基于 Websocket 方式的 Ticker 和 Depth 行情订阅规范；
- 2，基于 https 方式的 Kline 数据下载规范；

API 支持的开发语言包括：

- 1, 基于 Boost 的 C++语言；
- 2，Python 语言；
- 3，Java 语言。

每种接口规范及语言调用规范，均需要撰写标准的文档格式，并且发布到网站。支持中、英文两版。

## 二、WebSocket 方式行情数据目录

服务端返回的数据，都是按照标准格式来给客户端的。

### 1.消息格式定义

消息格式包括消息长度和消息体两部分，定义如下：

`<message> = <message_length> <message_body>`

`<message_length>`：消息长度，为固定的 4 位字符，内容为用字符表示的整数值，仅由空格和 0~9 数字字符构成，用于记录其后全部消息内容`<message_body>`的字符个数（不包括自己`<message_length>`这 4 个字符）。样例：`" 8"," 33"," 123","2699"`

`<message_body>`：消息体，为变长的数据，数据各字段之间用逗号分隔。

`<message>`：为完整消息，定义为`<message_length> <message_body>`，仅由可见字符构成，可以直接打印查看，并可通过 HTTP、文本文件等方式进行传输。

`<message>`完整样例:

`"9a,b,c,d,e"`

`" 261,zb,btc_usdt,this_week,34"`

### 2.消息体定义

下面的说明均是对`<message_body>`进一步定义，不包括`<message_length>`。

消息体定义格式为消息头和消息体，如下：

`<message_body> = <HEADER> (, <data>)`

其中消息头字段为：

`<HEADER> = type, token, exchange_name, symbol_type, symbol_name, symbol_info, account_id, req_id`

**type:** 为整数，数据类型，管理为 0~9，行情为 10~39，交易为 40~69，监控为 70~99

**token(新增):** 为一个加密字符串，通过登录获得，最大长度为 16 字节。除登录消息以外，所有网关发起的请求消息，均需要填充此字段；所有从网关传输给客户端的消息（如行情消息，交易请求的回复消息等），均无需填充此字段，如果填充，也将被客户端忽略。

**exchange\_name:** 为交易所域名，不含 "www./com/.net"，全小写，最大长度为 20 字节

**symbol\_type:** 为整数，

0 为 Spot 数字货币现货，

1 为 Future 数字货币期货，

2 为 Option 数字货币期权

**symbol\_name:** 为品种名称，最大长度为 20 字节

**symbol\_info:** 为品种信息，最大长度为 20 字节。当 symbol\_type=0（为现货）时，表明是否为保证金品种，0 为非保证金品种，1 为保证金品种。当

symbol\_type=1（为期货）时，为合约类别（如 okex 对应的是 this\_week、

next\_week、quarter 等）

**account\_id**:在交易网关配置文件上，通过 exchange&api\_key&secret\_key 映射出来的一个虚拟用户 id.

**req\_id**: 请求 ID，是 UTC 的时间戳，精确到毫秒级，每次请求都不能重复，用来限制重发。每个请求对应的回复消息，也会在 header 里带上此 req\_id，以便两者相匹配。

### 3.登录消息

行情和交易端在连接网关之前，均需要登录进行身份认证，返回 token 作为后续传输的消息包头身份标识。

请求：<message\_body>=<HEADER>,username,password

<HEADER>.type=70

username 网关配置的用户名

password:网关配置用户名的密码

应答：<message\_body>=<HEADER>,result,error\_code,error\_message,token

result: 0=失败，1=成功

error\_code: 出错代码，仅当 result=0 时，此段才有值

error\_message: 出错信息，仅当 result=0 时，此段才有值，最大长度为 50 字节

token: 新生成的用于请求验证的 token

备注： token 的生成机制，采用用户 ID,创建机器时间，生成一个 256 位的哈希值，

如果用户 token 已存在并且未失效，则直接返回，否则返回空。

## 4.行情消息

行情约定如下：

1. 行情数据采用服务端全推方式，发送给客户端。
2. 客户端不发送订阅请求，而是由配置文件配置后自动产生订阅。
3. timestamp 为 Unix 时间戳

### 4.1 Ticker 快照行情

<HEADER>.type=11:

<message\_body> = <HEADER>,timestamp,last,buy,sell,limit\_high,limit\_low,day  
\_high,day\_low,vol,change,unit\_amount,hold\_amount;

### 4.2 Depth 深度行情

<HEADER>.type=12:

<message\_body> = <HEADER>,timestamp,flag,bids\_count,asks\_count,bids[pric  
e,vol,.....],asks[price,vol,.....]

flag=0：标识当前包不是最后一个包，后续还有包，策略不应使用目前累积的不完整的深度行情数据。

flag=1：标识当前包是最后一个包，策略可以使用当前累积的完整的深度行情用于计

算。

### 三、Restful 模式下的行情数据目录

#### 1. restful 模式下的 Kline K 线行情

这部分数据采用历史下载的方式，从另外一台 redis 服务器上下载，规范如下：

请求 <http://kx.18hezi.com/bitquote/v1/kline>(通过采集交易所行情生成的K线)

请求参数: exchange, symbol, since,size,type , since可空 , size可空、默认取最大值

2000 , 当since为空,取最新size条数据

type:1min 3min 5min 10min 30min 1day...

例如 : exchange =bitfinex.com, symbol=btc\_usd, since=0,size=100 .

应答 : kline历史数据流采用json的格式返回 , 具体如下 :

```
[  
  
  {  
  
    "open": 1234.5678,  
  
    "high": 2345.5678  
  
    "low": 1233.5678,  
  
    "close": "1236.5678",  
  
    "vol": "0.23"  
  
    "timestamp": 1472711925,  
  
    }...  
]
```



举例：

[http://kx.18hezi.com/bitquote/v1/kline?exchange=okex&symbol=btc\\_usdt&since=0&size=100&type=1min](http://kx.18hezi.com/bitquote/v1/kline?exchange=okex&symbol=btc_usdt&since=0&size=100&type=1min)

## 2. restful 模式下的 ticker 行情

请求：<http://kx.18hezi.com/bitquote/v1/ticker>

请求参数: exchange, symbol, 例如: exchange =bitfinex.com,symbol=btc\_usd

应答: ticker 行情数据流如下:

```
{
  "ticker": {
    "vol": "40.463",
    "last": "0.899999",
    "sell": "0.5",
    "buy": "0.225",
    "high": "0.899999",
    "low": "0.081",
    "open": "0.231", //可能为空
    "close": "0.231", //可能为空
    "cny": "40.2222" //对应人民币的价格，可能为空或 0
  },
  "timestamp": "1507875747359"
}
```

举例：[http://kx.18hezi.com/bitquote/v1/ticker?exchange=bitfinex&symbol=btc\\_usd](http://kx.18hezi.com/bitquote/v1/ticker?exchange=bitfinex&symbol=btc_usd)

## 3. restful 模式下的 depth 行情

请求：<http://kx.18hezi.com/bitquote/v1/depth>

请求参数: exchange, symbol, 例如: exchange =bitfinex,symbol=btc\_usd

应答: depth 行情数据流，详细如下:

```
{
  "asks": [
    [
      83.28,
      11.8
    ]...
  ],
  "bids": [
```

```
[
    81.91,
    3.65
]...
],
"timestamp": 时间戳
}
```

## 四. Restful 模式下的交易消息

目前主要参考 OKEX 的交易 API 进行设计，实际开发过程中，可能会根据实际交易需求，以及不同交易所和交易 API 进行简化或修改。

每个应答消息的<HEADER>，与其请求消息的<HEADER>一致，因此可以通过<HEADER>中的 req\_id，将请求消息与应答消息对应起来。

### 1.下单

请求：

<message\_body>=<HEADER>,price,amount,buy\_sell,order\_type,open\_close,  
leverage

<HEADER>.type=40

price 委托价格，数字型字符串，比如 0.12345678.

amount 委托数量，数字型字符串，比如 10.12.

buy\_sell 委托方向, 0-买入，1-卖出.

order\_type 委托类型（0-限价委托，1-市价委托）.

open\_close 开平标识 ( 0-开仓 , 1-平仓 )。

leverage 期货合约杠杆 , 数字型字符串 , 比如 10 代表 10 倍杠杆。此字段对于现货交易不适用 , 直接留空。

应答 : <message\_body>=<HEADER>,result,error\_code,error\_message,order\_id

result: 0=失败 , 1=成功

error\_code: 出错代码 , 仅当 result=0 时 , 此段才有值

error\_message: 出错信息 , 仅当 result=0 时 , 此段才有值 , 最大长度为 50 字

节

order\_id: 委托序号 , 最大长度为 20 的字符串

## 2.撤单

请求 : <message\_body>=<HEADER>,order\_id,buy\_sell

<HEADER>.type=41

order\_id. 委托序号 , 最大长度为 20 的字符串

buy\_sell 委托方向, 0-买入 , 1-卖出 , 目前只有 bithumb 交易所要求提供此字段 , 其它交易所留空。

应答 : <message\_body>=<HEADER>,result,error\_code,error\_message

result: 0=失败 , 1=成功

error\_code: 出错代码 , 仅当 result=0 时 , 此段才有值

error\_message: 出错信息，仅当 result=0 时，此段才有值，最大长度为 50 字

节

### 3.查单

请求：

<message\_body>=<HEADER>,order\_id,status,current\_page,page\_length

<HEADER>.type=42

order\_id: 委托序号，最大长度为 20 的字符串，“-1” 查询所有未完成订单，否则查询相应订单号的订单

status: 委托状态(0 等待成交 1 部分成交 2 全部成交 -1 撤单 4 撤单处理中 5 撤单中)

current\_page:当前页数（目前版本的交易网关，似乎没有处理这个字段，请确认）

page\_length:每页记录数

应答：

<message\_body>=<HEADER>,result,error\_code,error\_message,count,[记录]

result: 0=失败，1=成功

error\_code: 出错代码，仅当 result=0 时，此段才有值

error\_message: 出错信息，仅当 result=0 时，此段才有值，最大长度为 50 字

节

count:记录条数，以下为每一条记录的信息

amount(double): 委托数量

contract\_name(string): 合约名称

create\_date(long): 委托时间

deal\_amount(double): 成交数量

fee(double): 手续费

order\_id(string ): 订单 ID，最大长度为 20 的字符串

price(double): 订单价格

price\_avg(double): 平均价格

status(int): 订单状态(0:等待成交,1:部分成交, 2:全部成交, -1:撤单,4:撤单处

理中)

symbol(string)

type(int): 订单类型 1:开多, 2:开空, 3:平多,4:平空.

unit\_amount(double):合约面值.此字段只针对期货，现货为空

lever\_rate(double):杠杆倍数。okex 默认为 10。此字段只针对期货，现货

为空

## 五、历史数据文件下载

不同于前面“三.1”节介绍的每个品种的 K 线下载，对于一些大批量数据（例如 1 个月至 1 年以上的数据，一个交易所的所有数据，等等），我们采用文件压缩的方式，供用户一次性下载。

举例如下：

下载 binance 交易所的品种 ada\_bnb 的某日 1 分钟历史 K 线数据，我们采用如下地址：

[http://kx.blueye.info/download2/2017-12-29/Kline\\$bitfinex&bcc\\_btc&1min.dat](http://kx.blueye.info/download2/2017-12-29/Kline$bitfinex&bcc_btc&1min.dat)。

其中文件的命名规范：交易所、品种、周期（比如 1day, 1min, ）。例如 Kline\$binance&ada\_bnb&1day.dat，kline 为 k 线的意思，binance 为交易所，ada\_bnb 为品种名称，1day 为 k 线周期。

文件内容如下(day 和 1min 文件内容规范一致)：

时间（LINUX 时间乘以 1000，单位毫秒）

成交量、开盘价、最低价、最高价、收盘价。

数据样本举例：

*[1515043080000,9897.82980822,0.00983965,0.00980104,0.00983965,0.00*

*980103]* , 其中 :

*1515043080000* 为时间 ( LINUX 时间乘以 1000 , 单位毫秒 ) , *9897.82980822*

为成交量 ; *0.00983965* 开盘价 , *0.00980104* 为最低价 , **0.00983965** 为最高价、

*0.00980103* 为收盘价。