**数字货币网关消息定义**

|  |  |  |
| --- | --- | --- |
| **版本** | **编写** | **修改日期** |
| 1.5 |  | 2018-03-19 |
| 1.6 |  | 2018-04-01 |
| 1.7 |  | 2018-04-07 |
| **截至2018-4-7，共实现以下消息：**  **4.1统一ticker数据**  **4.2 统一depth数据**  **5.1.委托买卖**  **5.2.委托撤单**  **5.3.查询委托** | | |
| 1.8 | 修改req\_id的描述 | 2018-06-15 |
| 5.5 | 增加校验token的请求 | 2018-06-19 |
| 3 | 删除登录消息 | 2018-06-19 |
|  | | |
|  | | |

目录

[1.消息格式定义 2](#_Toc510377616)

[2.消息体定义 2](#_Toc510377617)

[3.登录消息 3](#_Toc510377618)

[4.行情消息 3](#_Toc510377619)

[4.1 Ticker快照行情 3](#_Toc510377620)

[4.2 Depth深度行情 3](#_Toc510377621)

[4.2 Kline K线行情 4](#_Toc510377622)

[5.交易消息 4](#_Toc510377623)

[5.1.下单 4](#_Toc510377624)

[5.2.撤单 4](#_Toc510377625)

[5.3.查单 5](#_Toc510377626)

[5.4.查账户 5](#_Toc510377627)

[6.监控消息 6](#_Toc510377628)

[7.管理消息 7](#_Toc510377629)

[7.1.输出日志 7](#_Toc510377630)

[7.2.启动策略 7](#_Toc510377631)

[7.3.暂停策略 8](#_Toc510377632)

[7.4.继续策略 8](#_Toc510377633)

[7.5.停止策略 8](#_Toc510377634)

[7.6.重启策略 8](#_Toc510377635)

## 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[price,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。

### 4.2 Kline K线行情

<HEADER>.type=13: <message\_body>=<HEADER>,timestamp,cycle\_type(T,S,M,H,D,W,O,Q,Y,V,R,C),cycle\_value(1,2,3,...),open,high,low,close,vol1,vol2

返回行情周期类型的定义：T=Tick, S=Second, M=Minute, H=Hour, D=Day, W=Week, O=Month, Q=Quarter, Y=Year, V=Volume, R=Range, C=Custom

行情返回中为空的字段，表示没有值，无需处理。

## 5.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。

### 5.1.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

### 5.2.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

### 5.3.查单

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

buy\_sell 委托方向, 0-买入，1-卖出.（bithumb需要此参数，其他交易所留空）

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

### 5.4.查账户

请求：<message\_body>=<HEADER>,symbol\_type,symbol\_info

<HEADER>.type=43

symbol\_type: 0表示查现货，1表示查期货。当symbol\_type=0时，需要根据symbol\_info是否为“1”，来确定是否查询Margin帐户：symbol\_info=0，表示查询现货账户，symbol\_info=1，表示查询symbol\_name对应的margin账户。

应答：  
<message\_body>=<HEADER>,result,error\_code,error\_message,count,[symbol,free,freezed]

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

symbol:品种组合代码

free: 现货为可用余额，期货为账户权益

freezed: 冻结余额，只针对现货

### 5.5.校验token

请求：<message\_body>=<HEADER>

<HEADER>.type=45

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

result: 0=失败，1=成功

## 6.监控消息

为保证不影响行情和交易网关的运行性能，改监控后端程序采用独立运行的方式，可以和交易程序运行在同一主机（不同的网络侦听端口），也可以采取分布式运行方式。

交易记录由自动交易发起端在每执行一次交易请求成功后，通过TCP协议向监控端口发送一条交易记录日志。并作特别说明如下：

A, 如果交易记录为开仓并且未成交，则监控端增加一条交易记录；

B, 如果交易记录为开仓并且已成交，则监控端根据交易发起端传过来的内容直接修改trans\_no对应的记录所有字段即可；

C, 如果交易记录为平仓并且未成交，则监控端根据交易发起端传过来的内容直接修改trans\_no对应的记录所有字段即可；

D, 如果交易记录为平仓并且已成交，则监控端根据交易发起端传过来的内容直接修改trans\_no对应的记录所有字段即可。

交易记录的请求和应答消息格式如下

请求：<message\_body>=<HEADER>,m\_trans\_no,....

<HEADER>.type=75

m\_trans\_no: string,

m\_open\_order.m\_order\_no,

m\_open\_order.m\_order\_price,

m\_open\_order.m\_order\_volume,

m\_open\_order.m\_method

m\_open\_order.m\_order\_time: int

m\_open\_order.m\_executed\_no

m\_open\_order.m\_executed\_price

m\_open\_order.m\_executed\_volume

m\_open\_order.m\_executed\_time: int

m\_close\_order.m\_order\_no,

m\_close\_order.m\_order\_price,

m\_close\_order.m\_order\_volume,

m\_close\_order.m\_method,

m\_close\_order.m\_order\_time: int

m\_close\_order.m\_executed\_no,

m\_close\_order.m\_executed\_price,

m\_close\_order.m\_executed\_volume,

m\_close\_order.m\_executed\_time: int

m\_status: int

m\_profit\_percent

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

result: 0=失败，1=成功

备注：请求消息字段之间用逗号分隔，只有字段值没有字段名称。

举例：<HEADER>,1,2,BTC\_USDT,1,6365000,1,5,1504668364,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0

## 7.管理消息

这类消息主要是由用户端向交易策略端发起的执行消息。交易策略端的启动、停止等等，可以使用网关端的shell命令来进行操作。这种模式下，网关端与交易策略就需要运行在同一台主机上。如果网关端和策略端分布式部署，则需要在策略端增加命令交易的侦听端口，这会增加程序的工作量和复杂度。因此该部分工作先采用同一主机运行的模式。

远程遥控指令消息格式定义如下：

### 7.1.输出日志

请求：<message\_body>=<HEADER>,log\_level,log\_text

<HEADER>.type=0

log\_level:int，0=ALL,1=TRACE,2=DEBUG,3=INFO,4=AUDIT,5=WARN,6=ERROR,7=FATAL

log\_text: string,日志内容

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

<HEADER>.type=0

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

### 7.2.启动策略

请求：<message\_body>=<HEADER>

<HEADER>.type=1

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

### 7.3.暂停策略

请求：<message\_body>=<HEADER>

<HEADER>.type=2

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

### 7.4.继续策略

请求：<message\_body>=<HEADER>

<HEADER>.type=3

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

### 7.5.停止策略

请求：<message\_body>=<HEADER>

<HEADER>.type=4

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

### 7.6.重启策略

请求：<message\_body>=<HEADER>

<HEADER>.type=5

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