

- [简介](#)
- [快速入门](#)
 - [接入准备](#)
 - [SDK与代码示例](#)
 - [接口类型](#)
 - [接入URLs](#)
 - [签名认证](#)
 - [子账号](#)
 - [业务字典](#)
- [接入说明](#)
 - [接口概览](#)
 - [限频规则](#)
 - [请求格式](#)
 - [返回格式](#)
 - [错误信息](#)
 - [使用建议](#)
- [常见问题](#)
 - [接入、验签相关](#)
 - [行情相关](#)
 - [交易相关](#)
 - [账户充值相关](#)
 - [技术支持](#)
- [基础信息](#)
 - [获取所有交易对](#)
 - [获取所有币种](#)
 - [APIv2 币种参考信息](#)
 - [获取当前系统时间](#)
- [行情数据](#)
 - [K 线数据 \(蜡烛图 \)](#)
 - [聚合行情 \(Ticker \)](#)
 - [所有交易对的最新 Tickers](#)
 - [市场深度数据](#)
 - [最近市场成交记录](#)
 - [获得近期交易记录](#)
 - [最近24小时行情数据](#)
- [账户相关](#)
 - [账户信息](#)
 - [账户余额](#)
 - [账户流水](#)
 - [资产划转 \(母子账号之间 \)](#)
 - [子账号余额 \(汇总 \)](#)
 - [子账号余额](#)
- [钱包 \(充值与提现 \)](#)
 - [APIv2 充币地址查询](#)
 - [APIv2 提币额度查询](#)
 - [虚拟币提现](#)
 - [取消提现](#)
 - [充提记录](#)
- [现货](#)
 - [下单](#)
 - [撤销订单](#)
 - [撤销订单 \(基于client order ID \)](#)
 - [查询当前未成交订单](#)
 - [批量撤销订单 \(open orders \)](#)
 - [批量撤销订单](#)
 - [查询订单详情](#)
 - [查询订单详情 \(基于client order ID \)](#)
 - [成交明细](#)

- [搜索历史订单](#)
 - [搜索最近48小时内历史订单](#)
 - [当前和历史成交](#)
 - [获取用户当前手续费率](#)
- [Websocket行情数据](#)
 - [简介](#)
 - [K线数据](#)
 - [市场深度行情数据](#)
 - [买一卖一逐笔行情](#)
 - [成交明细](#)
 - [市场概要](#)
- [Websocket资产及订单](#)
 - [简介](#)
 - [订阅账户更新](#)
 - [订阅订单更新](#)
 - [订阅订单更新 \(NEW\)](#)
 - [请求用户资产数据](#)
 - [请求当前及历史订单](#)
 - [以订单编号请求订单](#)
- [Websocket资产及订单 \(v2\)](#)
 - [简介](#)
 - [订阅清算后成交明细](#)
 - [订阅账户变更](#)

简介

API 简介

欢迎使用火币韩国 API !

此文档是火币韩国API的唯一官方文档，火币韩国API提供的能力会在此持续更新，请大家及时关注。

你可以通过点击上方菜单来切换获取不同业务的API，还可通过点击右上角的语言按钮来切换文档语言。

文档右侧是针对请求参数以及响应结果的示例。

火币韩国 REST API URL

<https://api-cloud.huobi.co.kr>

火币韩国Websocket Feed (行情)

`wss://api-cloud.huobi.co.kr/ws`

专业交易人员使用的 REST API URL

<https://krapi.huobi.pro> (AWS Tokyo 地区专用)

- 我们为预申请者 and 选定的成员提供专用的URL服务。
- 详细内容，请参阅[面向专业交易者的API服务指南](https://support.huobi.co.kr/hc/en/articles/360035235091) (<https://support.huobi.co.kr/hc/en/articles/360035235091>) 。

做市商项目

欢迎有优秀 maker 策略且交易量大的用户参与长期做市商项目。如果您的火币现货账户中有折合大于10BTC资产，请提供以下信息发送邮件至：

- MM-kr@huobi.com Huobi Korea（现货）做市商申请；
- 1. 提供 UID（需不存在返佣关系的 UID）；
- 2. 提供其他交易平台 maker 交易量截图证明（比如30天内成交量，或者 VIP 等级等）；
- 3. 请简要阐述做市方法，不需要细节。

i 做市商项目不支持点卡抵扣、VIP、交易量相关活动以及任何形式的返佣活动。

联系我们

API 使用中如有疑问或咨询事项，请参考[咨询事项 Q&A](#)或者通过加入[火币韩国API交流Telegram群（http://bit.ly/21Y4o7v）](http://bit.ly/21Y4o7v)咨询。

快速入门

接入准备

如需使用API，请先登录网页端，完成API key的申请和权限配置，再据此文档详情进行开发和交易。

您可以点击 [这里](#) 创建 API Key。

每个母账号可创建5组Api Key，每个Api Key可对应设置读取、交易、提币三种权限。

权限说明如下：

- 读取权限：读取权限用于对数据的查询接口，例如：订单查询、成交查询等。
- 交易权限：交易权限用于下单、撤单、划转类接口。
- 提币权限：提币权限用于创建提币订单、取消提币订单操作。

创建成功后请务必记住以下信息：

- [Access Key](#) API 访问密钥
- [Secret Key](#) 签名认证加密所使用的密钥（仅申请时可见）

i 创建 API Key 时可以绑定 IP 地址，未绑定 IP 地址的 API Key 有效期为90天。出于安全考虑，强烈建议您绑定 IP 地址。

i 风险提示：这两个密钥与账号安全紧密相关，无论何时都请勿将二者同时向其它人透露。API Key的泄露可能会造成您的资产损失（即使未开通提币权限），若发现API Key泄露请尽快删除该API Key。

SDK与代码示例

SDK (推荐)

Java

Python3

C++

其它代码示例

<https://github.com/huobiapi?tab=repositories>

接口类型

火币为用户提供两种接口，您可根据自己的使用场景和偏好来选择适合的方式进行查询行情、交易或提现。

REST API

REST，即Representational State Transfer的缩写，是目前较为流行的基于HTTP的一种通信机制，每一个URL代表一种资源。

交易或资产提现等一次性操作，建议开发者使用REST API进行操作。

WEBSOCKET API

WebSocket是HTML5一种新的协议（Protocol）。它实现了客户端与服务器全双工通信，通过一次简单的握手就可以建立客户端和服务器的连接，服务器可以根据业务规则主动推送信息给客户端。

市场行情和买卖深度等信息，建议开发者使用WebSocket API进行获取。

接口鉴权

以上两种接口均包含公共接口和私有接口两种类型。

公共接口可用于获取基础信息和行情数据。公共接口无需认证即可调用。

私有接口可用于交易管理和账户管理。每个私有请求必须使用您的API Key进行签名验证。

接入URLs

您可以自行比较使用api-cloud.huobi.co.kr和krapi.huobi.pro(日本aws专用)两个域名的延迟情况，选择延迟低的进行使用。

REST API

`https://api-cloud.huobi.co.kr`

Websocket Feed (行情)

`wss://api-cloud.huobi.co.kr/ws`

Websocket Feed (资产和订单)

wss://api-cloud.huobi.co.kr/ws/v1

❗ 鉴于延迟高和稳定性差等原因，不建议通过代理的方式访问火币韩国API。

❗ 为保证API服务的稳定性，建议使用日本AWS云服务器进行访问。如使用中国大陆境内的客户端服务器，连接的稳定性将难以保证。

签名认证

签名说明

API 请求在通过 internet 传输的过程中极有可能被篡改，为了确保请求未被更改，除公共接口（基础信息，行情数据）外的私有接口均必须使用您的 API Key 做签名认证，以校验参数或参数值在传输途中是否发生了更改。

每一个API Key需要有适当的权限才能访问相应的接口，每个新创建的API Key都需要分配权限。在使用接口前，请查看每个接口的权限类型，并确认你的API Key有相应的权限。

一个合法的请求由以下几部分组成：

- 方法请求地址：即访问服务器地址 api-cloud.huobi.co.kr，比如 api-cloud.huobi.co.kr/v1/order/orders。
- API 访问Id (AccessKeyId)：您申请的 API Key 中的 Access Key。
- 签名方法 (SignatureMethod)：用户计算签名的基于哈希的协议，此处使用 HmacSHA256。
- 签名版本 (SignatureVersion)：签名协议的版本，此处使用2。
- 时间戳 (Timestamp)：您发出请求的时间 (UTC 时间)。如：2017-05-11T16:22:06。在查询请求中包含此值有助于防止第三方截取您的请求。
- 必选和可选参数：每个方法都有一组用于定义 API 调用的必需参数和可选参数。可以在每个方法的说明中查看这些参数及其含义。
 - 对于 GET 请求，每个方法自带的参数都需要进行签名运算。
 - 对于 POST 请求，每个方法自带的参数不进行签名认证，并且需要放在 body 中。
- 签名：签名计算得出的值，用于确保签名有效和未被篡改。

签名步骤

规范要计算签名的请求 因为使用 HMAC 进行签名计算时，使用不同内容计算得到的结果会完全不同。所以在进行签名计算前，请先对请求进行规范化处理。下面以查询某订单详情请求为例进行说明：

查询某订单详情时完整的请求URL

https://api-cloud.huobi.co.kr/v1/order/orders?

AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx

&SignatureMethod=HmacSHA256

&SignatureVersion=2

&Timestamp=2017-05-11T15:19:30

&order-id=1234567890

1. 请求方法 (GET 或 POST)，后面添加换行符 “\N”

GET\n

2. 添加小写的访问地址，后面添加换行符 “\N”

```
api-cloud.huobi.co.kr\n
```

3. 访问方法的路径，后面添加换行符 “\N”

```
/v1/order/orders\n
```

4. 对参数进行URI编码，并且按照ASCII码顺序进行排序

例如，下面是请求参数的原始顺序，且进行URI编码后

```
AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx
```

```
order-id=1234567890
```

```
SignatureMethod=HmacSHA256
```

```
SignatureVersion=2
```

```
Timestamp=2017-05-11T15%3A19%3A30
```

❶ 使用 UTF-8 编码，且进行了 URI 编码，十六进制字符必须大写，如 “:” 会被编码为 “%3A”，空格被编码为 “%20”。

❷ 时间戳（Timestamp）需要以YYYY-MM-DDThh:mm:ss格式添加并且进行 URI 编码。

经过排序之后

```
AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx
```

```
SignatureMethod=HmacSHA256
```

```
SignatureVersion=2
```

```
Timestamp=2017-05-11T15%3A19%3A30
```

```
order-id=1234567890
```

5. 按照以上顺序，将各参数使用字符 “&” 连接

```
AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2017-05-11T15%3A19%3A30&order-id=1234567890
```

6. 组成最终的要进行签名计算的字符串如下

```
GET\n
```

```
api-cloud.huobi.co.kr\n
```

```
/v1/order/orders\n
```

```
AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2017-05-11T15%3A19%3A30&order-id=1234567890
```

7. 用上一步里生成的“请求字符串”和你的密钥 (SECRET KEY) 生成一个数字签名

1. 将上一步得到的请求字符串和 API 私钥作为两个参数，调用HmacSHA256哈希函数来获得哈希值。

2. 将此哈希值用base-64编码，得到的值作为此次接口调用的数字签名。

4F65x5A2bLyMwVQj3Aqp+B4w+ivaA7n50i2SuYtCJ9o=

8. 将生成的数字签名加入到请求的路径参数里

- 1. 把所有必须的认证参数添加到接口调用的路径参数里
- 2. 把数字签名在URL编码后加入到路径参数里，参数名为“Signature”。

最终，发送到服务器的 API 请求应该为

```
https://api-cloud.huobi.co.kr/v1/order/orders?AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx&order-id=1234567890&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2017-05-11T15%3A19%3A30&Signature=4F65x5A2bLyMwVQj3Aqp%2BB4w%2BivaA7n50i2SuYtCJ9o%3D
```

子账号

子账号可以用来隔离资产与交易，资产可以在母子账号之间划转；子账号用户只能在子账号内进行交易，并且子账号之间资产不能直接划转，只有母账号有划转权限。

子账号拥有独立的登录账号密码和 API Key，均由母账号在网页端进行管理。

每个母账号可创建200个子账号，每个子账号可创建5组Api Key，每个Api Key可对应设置读取、交易两种权限。

子账号的 API Key 也可绑定 IP 地址, 有效期的限制与母账号的API Key一致。

您可以点击 [这里](#) 创建子账号并管理。

子账号可以访问所有公共接口，包括基本信息和市场行情，子账号可以访问的私有接口如下：

接口	说明
POST /v1/order/orders/place	创建并执行订单
POST /v1/order/orders/{order-id}/submitcancel	撤销一个订单
POST /v1/order/orders/batchcancel	批量撤销订单
POST /v1/order/orders/batchCancelOpenOrders	撤销当前委托订单
GET /v1/order/orders/{order-id}	查询一个订单详情
GET /v1/order/orders	查询当前委托、历史委托
GET /v1/order/openOrders	查询当前委托订单
GET /v1/order/matchresults	查询成交
GET /v1/order/orders/{order-id}/matchresults	查询某个订单的成交明细
GET /v1/account/accounts	查询当前用户的所有账户
GET /v1/account/accounts/{account-id}/balance	查询指定账户的余额

 其他接口子账号不可访问，如果尝试访问，系统会返回 “error-code 403”。

业务字典

交易对

交易对由基础币种和报价币种组成。以交易对 BTC/USDT 为例，BTC 为基础币种，USDT 为报价币种。

基础币种对应字段为 base-currency。

报价币种对应字段为 quote-currency。

账户

不同业务对应需要不同的账户，account-id为不同业务账户的唯一标识ID。

account-id可通过/v1/account/accounts接口获取，并根据account-type区分具体账户。

账户类型包括：

- spot：现货账户
- otc：OTC账户
- point：点卡账户

订单、成交相关ID说明

- order-id：订单的唯一编号
- client-order-id：客户自定义ID，该ID在下单时传入，与下单成功后返回的order-id对应，该ID 24小时内有效。
- match-id：订单在撮合中的顺序编号
- trade-id：成交的唯一编号

订单类型

当前火币的订单类型是由买卖方向以及订单操作类型组成，例如：buy-market，buy为买卖方向，market为操作类型。

买卖方向：

- buy：买
- sell：卖

订单种类：

- limit：限价单，该类型订单需指定下单价格，下单数量。
- market：市价单，该类型订单仅需指定下单金额或下单数量，不需要指定价格，订单在进入撮合时，会直接与对手方进行成交，直至金额或数量低于最小成交金额或成交数量为止。
- limit-maker：限价挂单，该订单在进入撮合时，只能作为maker进入市场深度，若订单会被成交，则撮合会直接拒绝该订单。
- ioc：立即成交或取消（immediately or cancel），该订单在进入撮合后，若不能直接成交，则会被直接取消（部分成交后，剩余部分也会被取消）。
- stop-limit：止盈止损单，设置高于或低于市场价格的订单，当订单到达触发价格后，才会正式的进入撮合队列。

订单状态

- submitted：等待成交，该状态订单已进入撮合队列当中。
- partial-filled：部分成交，该状态订单在撮合队列当中，订单的部分数量已经被市场成交，等待剩余部分成交。
- filled：已成交。该状态订单不在撮合队列中，订单的全部数量已经被市场成交。
- partial-canceled：部分成交撤销。该状态订单不在撮合队列中，此状态由partial-filled转化而来，订单数量有部分被成交，但是被撤销。
- canceled：已撤销。该状态订单不在撮合订单中，此状态订单没有任何成交数量，且被成功撤销。
- canceling：撤销中。该状态订单正在被撤销的过程中，因订单最终需在撮合队列中剔除才会被真正撤销，所以此状态为中间过渡态。

接入说明

接口概览

接口分类	分类链接	概述
基础类	/v1/common/*	基础类接口，包括币种、币种对、时间戳等接口
行情类	/market/*	公共行情类接口，包括成交、深度、行情等
账户类	/v1/account/* /v1/subuser/*	账户类接口，包括账户信息，子用户等
订单类	/v1/order/*	订单类接口，包括下单、撤单、订单查询、成交查询等

该分类为大类整理，部分接口未遵循此规则，请根据需求查看有关接口文档。

限频规则

- 每个API Key 在1秒之内限制10次
- 若接口不需要API Key，则每个IP在1秒内限制10次

比如：

- 资产订单类接口调用根据API Key进行限频：1秒10次
- 行情类接口调用根据IP进行限频：1秒10次

请求格式

所有的API请求都是restful，目前只有两种方法：GET和POST

- GET请求：所有的参数都在路径参数里
- POST请求，所有参数以JSON格式发送在请求主体（body）里

返回格式

所有的接口都是JSON格式。在JSON最上层有四个字段：`status`，`ch`，`ts` 和 `data`。前三个字段表示请求状态和属性，实际的业务数据在`data`字段里。

以下是一个返回格式的样例：

```
{
  "status": "ok",
```

```
"ch": "market.btcusdt.kline.1day",
"ts": 1499223904680,
"data": // per API response data in nested JSON object
}
```

参数名称	数据类型	描述
status	string	API接口返回状态
ch	string	接口数据对应的数据流。部分接口没有对应数据流因此不返回此字段
ts	int	接口返回的UTC时间的时间戳，单位毫秒
data	object	接口返回数据主体

错误信息

行情 API 错误信息

错误码	描述
bad-request	错误请求
invalid-parameter	参数错误
invalid-command	指令错误

交易 API 错误信息

错误码	描述
base-symbol-error	交易对不存在
base-currency-error	币种不存在
base-date-error	错误的日期格式
account for id 12,345 and user id 6,543,210 does not exist	account-id 错误，请使用GET /v1/account/accounts 接口查询
account-frozen-balance-insufficient-error	余额不足无法冻结
account-transfer-balance-insufficient-error	余额不足无法划转
bad-argument	无效参数
api-signature-not-valid	API签名错误
gateway-internal-error	系统繁忙，请稍后再试
ad-ethereum-addresss	请输入有效的以太坊地址
order-accountbalance-error	账户余额不足

错误码	描述
order-limitorder-price-error	限价单下单价格超出限制
order-limitorder-amount-error	限价单下单数量超出限制
order-orderprice-precision-error	下单价格超出精度限制
order-orderamount-precision-error	下单数量超过精度限制
order-marketorder-amount-error	下单数量超出限制
order-queryorder-invalid	查询不到此条订单
order-orderstate-error	订单状态错误
order-datelimit-error	查询超出时间限制
order-update-error	订单更新出错

使用建议

- 1、行情类数据信息，建议使用WebSocket方式实时接收数据，并对接收数据进行缓存操作。
- 2、获取最新成交价，推荐使用/market/trade接口或使用WebSocket订阅market.\$symbol.trade.detail，获取实时成交价格。
- 3、订单成交推送建议使用orders.\$symbol.update主题，该主题具有严格的时序性以及更低的时延。
- 4、下单资产信息使用WebSocket定订阅accounts主题，并结合Rest接口定时请求资产数据进行更新。

常见问题

接入、验签相关

Q1：一个用户可以申请多少个API KEY？

A: 每个母账号可创建5组Api Key，每个Api Key可对应设置读取、交易、提币三种权限。每个母账号还可创建200个子账号，每个子账号可创建5组Api Key，每个Api Key可对应设置读取、交易两种权限。

以下是三种权限的说明：

- 读取权限：读取权限用于对数据的查询接口，例如：订单查询、成交查询等。
- 交易权限：交易权限用于下单、撤单、划转类接口。
- 提币权限：提币权限用于创建提币订单、取消提币订单操作。

Q2：为什么经常出现断线、超时的情况？

A：请检查是否属于以下情况：

1. 请确认是否使用 api-cloud.huobi.co.kr 域名访问火币韩国 API
2. 请使用日本云服务器

Q3：为什么WEBSOCKET总是断开连接？

A：常见原因有：

1. 未回复Pong，WebSocket连接需在接收到服务端发送的Ping信息后回复Pong，保证连接的稳定。
2. 网络原因造成客户端发送了Pong消息，但服务端并未接收到。
3. 网络原因造成连接断开。
4. 建议用户做好WebSocket连接断连重连机制，在确保心跳（Ping/Pong）消息正确回复后若连接意外断开，程序能够自动进行重新连接。

Q4：为什么签名认证总返回失败？

A：请对比使用Secret Key签名前的字符串与以下字符串的区别

```
GET\n
api-cloud.huobi.co.kr\n
/v1/account/accounts\n
AccessKeyId=rfhxxxxx-950000847-boooooo3-432c0&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2019-10-28T07%3A28%3A38
```

对比时请注意一下几点：

1、签名参数应该按照ASCII码排序。比如下面是原始的参数：

```
AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx
order-id=1234567890
SignatureMethod=HmacSHA256
SignatureVersion=2
Timestamp=2017-05-11T15%3A19%3A30
```

排序之后应该为：

```
AccessKeyId=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx
SignatureMethod=HmacSHA256
SignatureVersion=2
Timestamp=2017-05-11T15%3A19%3A30
order-id=1234567890
```

2、签名串需进行URI编码。比如：

- 冒号：会被编码为%3A，空格会被编码为%20
- 时间戳需要格式化为YYYY-MM-DDThh:mm:ss，经过URI编码之后为2017-05-11T15%3A19%3A30

3、签名需进行 base64 编码

4、Get请求参数需在签名串中

5、时间为UTC时间转换为YYYY-MM-DDTHH:mm:ss

6、检查本机时间与标准时间是否存在偏差（偏差应小于1分钟）

7、WebSocket发送验签认证消息时，消息体不需要URI编码

8、签名时所带Host应与请求接口时Host相同

9、Api Key 与 Secret Key中是否存在隐藏特殊字符，影响签名

当前官方已支持Java、Python3、C++三种语言的SDK，可根据语言选择使用或进行参考。

SDK下载地址

Python签名样例代码

JAVA签名样例代码

C++签名样例代码

Q5：调用接口返回GATEWAY-INTERNAL-ERROR错误是什么原因？

A：请检查是否属于以下情况：

1. account-id 是否正确，需要由 GET /v1/account/accounts 接口返回的数据。
2. 可能为网络原因，请稍后进行重试。
3. 发送数据格式是否正确(需要标准JSON格式)。
4. POST请求头header需要声明为 `Content-Type:application/json`。

Q6：调用接口返回 LOGIN-REQUIRED错误是什么原因？

A：请检查是否属于以下情况：

1. 应该将AccessKey参数带入URL中。
2. account-id 是否正确，是由 `GET /v1/account/accounts` 接口返回的数据。
3. POST 请求不需要把body里的内容计算进签名。
4. GET 请求应该将参数按照 ASCII 码表顺序排序。

行情相关

Q1：当前盘口数据多久更新一次？

A：当前盘口数据一秒更新一次，无论是RESTful查询或是Websocket推送都是一秒一次。若需要实时的买一卖一价格数据，可使用WebSocket订阅market.\$symbol.bbo主题，该主题会实时推送最新的买一卖一价格数据。

Q2：24小时行情接口(/MARKET/DETAIL)数据接口的成交量会出现负增长吗？

A：/market/detail接口中的成交量、成交金额为24小时滚动数据（平移窗口大小24小时），有可能会后一个窗口内的累计成交量、累计成交额小于前一窗口的情况。

Q3：如何获取最新成交价格？

A：推荐使用REST API `GET /market/trade` 接口请求最新成交，或使用WebSocket订阅market.\$symbol.trade.detail主题，根据返回数据中的price获取。

Q4：K线是按照什么时间开始计算的？

A：K线周期以新加坡时间为基准开始计算，例如日K线的起始周期为新加坡时间0时-新加坡时间次日0时。

交易相关

Q1：ACCOUNT-ID是什么？

A：account-id对应用户不同业务账户的ID，可通过/v1/account/accounts接口获取，并根据account-type区分具体账户。

账户类型包括：

- spot 现货账户
- otc OTC账户
- point 点卡账户

Q2：CLIENT-ORDER-ID是什么？

A：client-order-id作为下单请求标识的一个参数，类型为字符串，长度为64。此id为用户自己生成，24小时内有效。

Q3：如何获取下单数量、金额、小数限制、精度的信息？

A：可使用 Rest API `GET /v1/common/symbols` 获取相关币对信息，下单时注意最小下单数量和最小下单金额的区别。

常见返回错误如下：

- order-value-min-error: 下单金额小于最小交易额
- order-orderprice-precision-error：限价单价格精度错误
- order-orderamount-precision-error：下单数量精度错误
- order-limitorder-price-max-error：限价单价格高于限价阈值
- order-limitorder-price-min-error：限价单价格低于限价阈值
- order-limitorder-amount-max-error：限价单数量高于限价阈值
- order-limitorder-amount-min-error：限价单数量低于限价阈值

Q4：WEBSOCKET 订单更新推送主题ORDERS.\$SYMBOL 和 ORDERS.\$SYMBOL.UPDATE的区别？

A：区别如下：

1. order.\$symbol 主题作为老的推送主题，会在一段时间后停止主题的维护和使用，推荐使用order.\$symbol.update主题。
2. 新主题orders.\$symbol.update具有严格的时序性，保证数据严格按照撮合成交顺序进行推送，且具有更快的时效性以及更低的时延。
3. 为减少重复数据推送量以及更快的速度，在orders.\$symbol.update推送中并未携带原始订单数量，价格信息，若需要此信息，建议可在下单时在本地维护订单信息，或在接收到推送消息后，使用Rest接口进行查询。

Q5：为什么收到订单成功成交的消息后再次进行下单，返回余额不足？

A：为保证订单的及时送达以及低延时，订单推送的结果是在撮合后直接推送，此时订单可能并未完成资产的清算。

建议使用以下方式保证资金可以正确下单：

1. 结合资产推送主题accounts同步接收资产变更的消息，确保资金已经完成清算。
2. 收到订单推送消息时，使用Rest接口调用账户余额，验证账户资金是否足够。
3. 账户中保留相对充足的资金余额。

Q6: 撮合结果里的FILLED-FEES和FILLED-POINTS有什么区别？

A: 撮合成交中的成交手续费分为普通手续费以及抵扣手续费两种类型，两种类型不会同时存在。

1. 普通手续费表示，在成交时，未开启HT抵扣、点卡抵扣，使用原币进行手续费扣除。例如：在BTCUSDT币种对下购买BTC，filled-fees字段不为空，表示扣除了普通手续费，单位是BTC。
2. 抵扣手续费表示，在成交时，开启了HT抵扣或点卡抵扣，使用HT或点卡进行手续费的抵扣。例如BTCUSDT币种对下购买BTC，HT\点卡充足时，filled-fees为空，filled-points不为空，表示扣除了HT或点卡作为手续费，扣除单位需参考fee-deduct-currency字段

Q7: 撮合结果中MATCH-ID和TRADE-ID有什么区别？

A: match-id表示订单在撮合中的顺序号，trade-id表示成交时的序号，一个match-id可能有多个trade-id（成交时），也可能没有trade-id(创建订单、撤销订单)

Q8: 为什么基于当前盘口买一或者卖一价格进行下单触发了下单限价错误？

A: 当前火币有基于最新成交价上下一定幅度的限价保护，对流动性不好的币，基于盘口数据下单可能会触发限价保护。建议基于ws推送的成交价+盘口数据信息进行下单

账户充提相关

Q1：为什么创建提币时返回API-NOT-SUPPORT-TEMP-ADDR错误？

A：因安全考虑，API创建提币时仅支持已在提现地址列表中的地址，暂不支持使用API添加地址至提币地址列表中，需在网页端或APP端添加地址后才可在API中进行提币操作。

Q2：为什么USDT提币时返回INVALID-ADDRESS错误？

A：USDT币种为典型的一币多链币种，创建提币订单时应填写chain参数对应地址类型。以下表格展示了链和chain参数的对应关系：

链	chain 参数
ERC20（默认）	usdt ERC20
OMNI	usdt
TRX	trc20usdt

如果chain参数为空，则默认的链为ERC20，或者也可以显示将参数赋值为usdt ERC20。

如果要提币到OMNI或者TRX，则chain参数应该填写usdt或者trc20usdt。chain参数可使用值请参考 [GET /v2/reference/currencies](#) 接口。

Q3：创建提币时FEE字段应该怎么填？

A：请参考 [GET /v2/reference/currencies](#)接口返回值，返回信息中withdrawFeeType为提币手续费类型，根据类型选择对应字段设置提币手续费。

提币手续费类型包含：

- transactFeeWithdraw：单次提币手续费（仅对固定类型有效，withdrawFeeType=fixed）
- minTransactFeeWithdraw：最小单次提币手续费（仅对区间类型有效，withdrawFeeType=circulated）
- maxTransactFeeWithdraw：最大单次提币手续费（仅对区间类型和有上限的比例类型有效，withdrawFeeType=circulated or ratio）
- transactFeeRateWithdraw：单次提币手续费率（仅对比例类型有效，withdrawFeeType=ratio）

Q4：如何查看我的提币额度？

A：请参考/v2/account/withdraw/quota接口返回值，返回信息中包含您查询币种的单次、当日、当前、总提币额度以及剩余额度的信息。

备注：若您有大额提币需求，且提币数额超出相关限额，可联系官方客服（发送信息至 API-kr@huobi.com 邮箱）进行沟通。

技术支持

若以上内容未能帮助到您，可选择以下任一方式联系我们：

1、火币韩国API交流Telegram群（<http://bit.ly/2lY4o7v>） 2、发送邮件至 API-kr@huobi.com 为了能够更快的了解和调查您反馈的问题，请按照如下模板向我们反馈问题。

1. UID
2. AccessKey
3. 完整请求URL
4. 请求参数
5. 请求时间点
6. 接口返回原始数据
7. 问题说明：（例如操作步骤，字段疑问，问题发生频率）
8. 签名前字符串（签名认证错误时必须填）

下方是一个应用了模板的例子：

1. UID: 123456
2. AccessKey: rfhxxxxx-950000847-boooooo3-432c0
3. 完整请求URL: <https://api-cloud.huobi.co.kr/v1/account/accounts?&SignatureVersion=2&SignatureMethod=HmacSHA256&Timestamp=2019-11-06T03%3A25%3A39&AccessKeyId=rfhxxxxx-950000847-boooooo3-432c0&Signature=HhJwApXKpaLPewiYLczwflkoTPnFPgYF61iq0iTFF8%3D>
4. 请求参数: 无
5. 请求时间点: 2019-11-06 11:26:14
6. 接口返回原始数据: {"status":"error","err-code":"api-signature-not-valid","err-msg":"Signature not valid: Incorrect Access key [Access key错误]","data":null}
7. 问题说明: 调用接口时发生了错误
8. 签名前字符串（签名认证错误时必须填）
GET\n
api-cloud.huobi.co.kr\n
/v1/account/accounts\n
AccessKeyId=rfhxxxxx-950000847-boooooo3-432c0&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2019-11-06T03%3A26%3A13

注意：Access Key仅能证明您的身份，不会影响您账户的安全。切记不要将Secret Key信息分享给任何人，若您不小心将Secret Key暴露，请尽快删除其对应的API Key，以免造成您的账户损失。

基础信息

获取所有交易对

此接口返回所有火币韩国站支持的交易对。

```
curl "https://api-cloud.huobi.co.kr/v1/common/symbols"
```

HTTP 请求

- GET /v1/common/symbols

请求参数

此接口不接受任何参数。

Responds:

```
"data": [
  {
    "base-currency": "etc",
    "quote-currency": "usdt",
    "price-precision": 6,
    "amount-precision": 4,
    "symbol-partition": "default",
    "symbol": "etcusdt",
    "state": "online",
    "value-precision": 8,
    "min-order-amt": 0.001,
    "max-order-amt": 10000,
    "min-order-value": 0.0001
  },
  {
    "base-currency": "ltc",
    "quote-currency": "usdt",
    "price-precision": 6,
    "amount-precision": 4,
    "symbol-partition": "main",
    "symbol": "ltcusdt",
    "state": "online",
    "value-precision": 8,
    "min-order-amt": 0.001,
    "max-order-amt": 10000,
    "min-order-value": 100,
    "leverage-ratio": 4
  }
]
```

返回字段

字段名称	数据类型	描述
base-currency	string	交易对中的基础币种
quote-currency	string	交易对中的报价币种
price-precision	integer	交易对报价的精度（小数点后位数）

字段名称	数据类型	描述
amount-precision	integer	交易对基础币种计数精度（小数点后位数）
symbol-partition	string	交易区，可能值: [main，innovation]
symbol	string	交易对
state	string	交易对状态；可能值: [online，offline,suspend] online - 已上线；offline - 交易对已下线，不可交易；suspend -- 交易暂停
value-precision	integer	交易对交易金额的精度（小数点后位数）
min-order-amt	long	交易对最小下单量（下单量指当订单类型为限价单或sell-market时，下单接口传的'amount'）
max-order-amt	long	交易对最大下单量
min-order-value	long	最小下单金额（下单金额指当订单类型为限价单时，下单接口传入的(amount * price)。当订单类型为buy-market时，下单接口传的'amount'）

获取所有币种

此接口返回所有火币韩国站支持的币种。

```
curl "https://api-cloud.huobi.co.kr/v1/common/currencys"
```

HTTP 请求

- GET `/v1/common/currencys`

请求参数

此接口不接受任何参数。

Response:

```
{
  "data": [
    "usdt",
    "eth",
    "etc"
  ]
}
```

返回字段

❗ 返回的“data”对象是一个字符串数组，每一个字符串代表一个支持的币种。

APIv2 币链参考信息

此节点用于查询各币种及其所在区块链的静态参考信息（公共数据）

HTTP 请求

- GET /v2/reference/currencies

```
curl "https://api-cloud.huobi.co.kr/v2/reference/currencies?currency=usdt"
```

请求参数

字段名称	是否必需	类型	字段描述	取值范围
currency	false	string	币种	btc, ltc, bch, eth, etc ... (取值参考 GET /v1/common/currencys)
authorizedUser	false	boolean	已认证用户	true or false (如不填，缺省为true)

Response:

```
{
  "code": 200,
  "data": [
    {
      "chains": [
        {
          "chain": "trc20usdt",
          "depositStatus": "allowed",
          "maxTransactFeeWithdraw": "1.00000000",
          "maxWithdrawAmt": "280000.00000000",
          "minDepositAmt": "100",
          "minTransactFeeWithdraw": "0.10000000",
          "minWithdrawAmt": "0.01",
          "numOfConfirmations": 999,
          "numOfFastConfirmations": 999,
          "withdrawFeeType": "circulated",
          "withdrawPrecision": 5,
          "withdrawQuotaPerDay": "280000.00000000",
          "withdrawQuotaPerYear": "280000.00000000",
          "withdrawQuotaTotal": "280000.00000000",
          "withdrawStatus": "allowed"
        },
        {
          "chain": "usdt",
          "depositStatus": "allowed",
          "maxWithdrawAmt": "19000.00000000",
          "minDepositAmt": "0.0001",
          "minWithdrawAmt": "2",
          "numOfConfirmations": 30,
```

```
        "numOfFastConfirmations":15,
        "transactFeeRateWithdraw":"0.00100000",
        "withdrawFeeType":"ratio",
        "withdrawPrecision":7,
        "withdrawQuotaPerDay":"90000.00000000",
        "withdrawQuotaPerYear":"111000.00000000",
        "withdrawQuotaTotal":"1110000.00000000",
        "withdrawStatus":"allowed"
    },
    {
        "chain":"usdterc20",
        "depositStatus":"allowed",
        "maxWithdrawAmt":"18000.00000000",
        "minDepositAmt":"100",
        "minWithdrawAmt":"1",
        "numOfConfirmations":999,
        "numOfFastConfirmations":999,
        "transactFeeWithdraw":"0.10000000",
        "withdrawFeeType":"fixed",
        "withdrawPrecision":6,
        "withdrawQuotaPerDay":"180000.00000000",
        "withdrawQuotaPerYear":"200000.00000000",
        "withdrawQuotaTotal":"300000.00000000",
        "withdrawStatus":"allowed"
    }
],
"currency":"usdt",
"instStatus":"normal"
}
]
```

响应数据

字段名称	是否必需	数据类型	字段描述	取值范围
code	true	int	状态码	
message	false	string	错误描述（如有）	
data	true	object		
{ currency	true	string	币种	
{ chains	true	object		
chain	true	string	链名称	
numOfConfirmations	true	int	安全上账所需确认次数（达到确认次数后允许提币）	
numOfFastConfirmations	true	int	快速上账所需确认次数（达到确认次数后允许交易但不允许提币）	

字段名称	是否必需	数据类型	字段描述	取值范围
minDepositAmt	true	string	单次最小充币金额	
depositStatus	true	string	充币状态	allowed,prohibited
minWithdrawAmt	true	string	单次最小提币金额	
maxWithdrawAmt	true	string	单次最大提币金额	
withdrawQuotaPerDay	true	string	当日提币额度	
withdrawQuotaPerYear	true	string	当年提币额度	
withdrawQuotaTotal	true	string	总提币额度	
withdrawPrecision	true	int	提币精度	
withdrawFeeType	true	string	提币手续费类型（特定币种在特定链上的提币手续费类型唯一）	fixed,circulated,ratio
transactFeeWithdraw	false	string	单次提币手续费（仅对固定类型有效，withdrawFeeType=fixed）	
minTransactFeeWithdraw	false	string	最小单次提币手续费（仅对区间类型有效，withdrawFeeType=circulated）	
maxTransactFeeWithdraw	false	string	最大单次提币手续费（仅对区间类型和有上限的比例类型有效，withdrawFeeType=circulated or ratio）	
transactFeeRateWithdraw	false	string	单次提币手续费率（仅对比例类型有效，withdrawFeeType=ratio）	
withdrawStatus}	true	string	提币状态	allowed,prohibited
instStatus }	true	string	币种状态	normal,delisted

状态码

状态码	错误信息	错误场景描述
200	success	请求成功
500	error	系统错误
2002	invalid field value in "field name"	非法字段取值

获取当前系统时间

此接口返回当前的系统时间，新加坡时区（UTC +8），单位毫秒。

```
curl "https://api-cloud.huobi.co.kr/v1/common/timestamp"
```

HTTP 请求

- GET /v1/common/timestamp

请求参数

此接口不接受任何参数。

Response:

"data": 1494900087029

行情数据

K 线数据（蜡烛图）

此接口返回历史K线数据。

HTTP 请求

- GET /market/history/kline

curl "https://api-cloud.huobi.co.kr/market/history/kline?period=1day&size=200&symbol=btcusdt"

请求参数

参数	数据类型	是否必须	默认值	描述	取值范围
symbol	string	true	NA	交易对	btcusdt, ethbtc... (取值参考 GET /v1/common/symbols) >
period	string	true	NA	返回数据时间粒度，也就是每根蜡烛的时间区间	1min, 5min, 15min, 30min, 60min, 4hour, 1day, 1mon, 1week, 1year
size	integer	false	150	返回 K 线数据条数	[1, 2000]

❗ 当前 REST API 不支持自定义时间区间，如需要历史固定时间范围的数据，请参考 WebSocket API 中的 K 线接口。

❗ 获取 hb10 净值时，symbol 请填写 “hb10” 。

❗ K线周期以新加坡时间为基准开始计算，例如日K线的起始周期为新加坡时间0时-新加坡时间次日0时。

Response:

```
[
  {
    "id": 1499184000,
    "amount": 37593.0266,
    "count": 0,
    "open": 1935.2000,
    "close": 1879.0000,
    "low": 1856.0000,
    "high": 1940.0000,
    "vol": 71031537.97866500
  }
]
```

响应数据

字段名称	数据类型	描述
id	integer	调整为新加坡时间的时间戳，单位秒，并以此作为此K线柱的id
amount	float	以基础币种计量的交易量
count	integer	交易次数
open	float	本阶段开盘价
close	float	本阶段收盘价
low	float	本阶段最低价
high	float	本阶段最高价
vol	float	以报价币种计量的交易量

聚合行情 (Ticker)

此接口获取ticker信息同时提供最近24小时的交易聚合信息。

HTTP 请求

- GET /market/detail/merged

```
curl "https://api-cloud.huobi.co.kr/market/detail/merged?symbol=ethusdt"
```

请求参数

参数	数据类型	是否必须	默认值	描述	取值范围
symbol	string	true	NA	交易对	btcusdt, ethbtc... (取值参考GET /v1/common/symbols)

Response:

```
{
  "id":1499225271,
  "ts":1499225271000,
  "close":1885.0000,
  "open":1960.0000,
  "high":1985.0000,
  "low":1856.0000,
  "amount":81486.2926,
  "count":42122,
  "vol":157052744.85708200,
  "ask":[1885.0000,21.8804],
  "bid":[1884.0000,1.6702]
}
```

响应数据

字段名称	数据类型	描述
id	integer	NA
amount	float	以基础币种计量的交易量
count	integer	交易次数
open	float	本阶段开盘价
close	float	本阶段最新价
low	float	本阶段最低价
high	float	本阶段最高价
vol	float	以报价币种计量的交易量
bid	object	当前的最高买价 [price, quote volume]
ask	object	当前的最低卖价 [price, quote volume]

所有交易对的最新 Tickers

获得所有交易对的 tickers，数据取值时间区间为24小时滚动。

 此接口返回所有交易对的 ticker，因此数据量较大。

HTTP 请求

- GET `/market/tickers`

```
curl "https://api-cloud.huobi.co.kr/market/tickers"
```

请求参数

此接口不接受任何参数。

Response:

```
[
  {
    "open":0.044297,      // 开盘价
    "close":0.042178,    // 收盘价
    "low":0.040110,      // 最高价
    "high":0.045255,     // 最低价
    "amount":12880.8510,
    "count":12838,
    "vol":563.0388715740,
    "symbol":"ethbtc"
  },
  {
    "open":0.008545,
    "close":0.008656,
    "low":0.008088,
    "high":0.009388,
    "amount":88056.1860,
    "count":16077,
    "vol":771.7975953754,
    "symbol":"ltcbtc"
  }
]
```

响应数据

核心响应数据为一个对象列，每个对象包含下面的字段

字段名称	数据类型	描述
amount	float	以基础币种计量的交易量
count	integer	交易笔数
open	float	开盘价
close	float	最新价
low	float	最低价
high	float	最高价
vol	float	以报价币种计量的交易量
symbol	string	交易对，例如btcusdt, ethbtc

市场深度数据

此接口返回指定交易对的当前市场深度数据。

HTTP 请求

- GET /market/depth

```
curl "https://api-cloud.huobi.co.kr/market/depth?symbol=btcusdt&type=step2"
```

请求参数

参数	数据类型	必须	默认值	描述	取值范围
symbol	string	true	NA	交易对	btcusdt, ethbtc... (取值参考 GET /v1/common/symbols)
depth	integer	false	20	返回深度的数量	5 , 10 , 20
type	string	true	step0	深度的价格聚合度，具体说明见下方	step0 , step1 , step2 , step3 , step4 , step5

❗ 当type值为 ‘step0’ 时， ‘depth’ 的默认值为150而非20。

参数type的各值说明

取值	说明
step0	无聚合
step1	聚合度为报价精度*10
step2	聚合度为报价精度*100
step3	聚合度为报价精度*1000
step4	聚合度为报价精度*10000
step5	聚合度为报价精度*100000

Response:

```
{
  "version": 31615842081,
  "ts": 1489464585407,
  "bids": [
    [7964, 0.0678], // [price, amount]
    [7963, 0.9162],
    [7961, 0.1],
    [7960, 12.8898],
    [7958, 1.2],
    ...
  ],
  "asks": [
```

```
[7979, 0.0736],
[7980, 1.0292],
[7981, 5.5652],
[7986, 0.2416],
[7990, 1.9970],
...
]
```

响应数据

i 返回的JSON顶级数据对象名为'tick'而不是通常的'data'。

字段名称	数据类型	描述
ts	integer	调整为新加坡时间的时间戳，单位毫秒
version	integer	内部字段
bids	object	当前的所有买单 [price, quote volume]
asks	object	当前的所有卖单 [price, quote volume]

最近市场成交记录

此接口返回指定交易对最新的一个交易记录。

HTTP 请求

- GET /market/trade

```
curl "https://api-cloud.huobi.co.kr/market/trade?symbol=ethusdt"
```

请求参数

参数	数据类型	是否必须	默认值	描述
symbol	string	true	NA	btcusdt, ethbtc... (取值参考GET /v1/common/symbols)

Response:

```
{
  "id": 600848670,
  "ts": 1489464451000,
  "data": [
    {
      "id": 600848670,
      "trade-id": 102043494568,
```

```
    "price": 7962.62,
    "amount": 0.0122,
    "direction": "buy",
    "ts": 1489464451000
  }
]
```

响应数据

i 返回的JSON顶级数据对象名为'tick'而不是通常的'data'。

字段名称	数据类型	描述
id	integer	唯一交易id (将被废弃)
trade-id	integer	唯一成交ID (NEW)
amount	float	以基础币种为单位的交易量
price	float	以报价币种为单位的成交价格
ts	integer	调整为新加坡时间的时间戳，单位毫秒
direction	string	交易方向：“buy” 或 “sell”，“buy” 即买，“sell” 即卖

获得近期交易记录

此接口返回指定交易对近期的所有交易记录。

HTTP 请求

- GET `/market/history/trade`

```
curl "https://api-cloud.huobi.co.kr/market/history/trade?symbol=ethusdt&size=2"
```

请求参数

参数	数据类型	是否必须	默认值	描述
symbol	string	true	NA	btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>)
size	integer	false	1	返回的交易记录数量，最大值2000

Response:

```
[
  {
    "id": 31618787514,
```

```
    "ts":1544390317905,
    "data":[
      {
        "amount":9.000000000000000000,
        "ts":1544390317905,
        "trade-id": 102043483472,
        "id":3161878751418918529341,
        "price":94.690000000000000000,
        "direction":"sell"
      },
      {
        "amount":73.771000000000000000,
        "ts":1544390317905,
        "trade-id": 102043483473
        "id":3161878751418918532514,
        "price":94.660000000000000000,
        "direction":"sell"
      }
    ]
  },
  {
    "id":31618776989,
    "ts":1544390311353,
    "data":[
      {
        "amount":1.000000000000000000,
        "ts":1544390311353,
        "trade-id": 102043494568,
        "id":3161877698918918522622,
        "price":94.710000000000000000,
        "direction":"buy"
      }
    ]
  }
]
```

响应数据

❗ 返回的数据对象是一个对象数组，每个数组元素为一个调整为新加坡时间的时间戳（单位毫秒）下的所有交易记录，这些交易记录以数组形式呈现。

参数	数据类型	描述
id	integer	唯一交易id（将被废弃）
trade-id	integer	唯一成交ID（NEW）
amount	float	以基础币种为单位的交易量
price	float	以报价币种为单位的成交价格
ts	integer	调整为新加坡时间的时间戳，单位毫秒

参数	数据类型	描述
direction	string	交易方向：“buy” 或 “sell”，“buy” 即买，“sell” 即卖

最近24小时行情数据

此接口返回最近24小时的行情数据汇总。

HTTP 请求

- GET `/market/detail`

```
curl "https://api-cloud.huobi.co.kr/market/detail?symbol=ethusdt"
```

请求参数

参数	数据类型	是否必须	默认值	描述
symbol	string	true	NA	btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>)

Response:

```
{
  "amount":613071.438479561,
  "open":86.21,
  "close":94.35,
  "high":98.7,
  "id":31619471534,
  "count":138909,
  "low":84.63,
  "version":31619471534,
  "vol":5.6617373443873316E7
}
```

响应数据

i 返回的JSON顶级数据对象名为'tick'而不是通常的'data'。

字段名称	数据类型	描述
id	integer	响应id
amount	float	以基础币种计量的交易量
count	integer	交易次数
open	float	本阶段开盘价

字段名称	数据类型	描述
close	float	本阶段收盘价
low	float	本阶段最低价
high	float	本阶段最高价
vol	float	以报价币种计量的交易量
version	integer	内部数据

账户相关

i 访问账户相关的接口需要进行签名认证。

账户信息

API Key 权限：读取

查询当前用户的所有账户 ID `account-id` 及其相关信息

HTTP 请求

- GET `/v1/account/accounts`

请求参数

无

Response:

```
{
  "data": [
    {
      "id": 100001,
      "type": "spot",
      "subtype": "",
      "state": "working"
    }
    {
      "id": 100002,
      "type": "margin",
      "subtype": "btcusdt",
      "state": "working"
    }
  ],
  {
```

```
    "id": 100003,
    "type": "otc",
    "subtype": "",
    "state": "working"
  }
]
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
id	true	long	account-id	
state	true	string	账户状态	working：正常, lock：账户被锁定
type	true	string	账户类型	spot：现货账户，otc：OTC 账户，point：点卡账户

账户余额

API Key 权限：读取

查询指定账户的余额，支持以下账户：

spot：现货账户，otc：OTC 账户，point：点卡账户

HTTP 请求

- GET `/v1/account/accounts/{account-id}/balance`

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
account-id	true	string	account-id，填在 path 中，取值参考		GET <code>/v1/account/accounts</code>

Response:

```
{
  "data": {
    "id": 100009,
    "type": "spot",
    "state": "working",
    "list": [
      {
        "currency": "usdt",
        "type": "trade",
        "balance": "5007.4362872650"
      },
      {
        "currency": "usdt",
        "type": "frozen",
```

```
    "balance": "348.1199920000"
  }
]
}
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
id	true	long	账户 ID	
state	true	string	账户状态	working：正常 lock：账户被锁定
type	true	string	账户类型	spot：现货账户，otc：OTC 账户，point：点卡账户
list	false	Array		

list字段说明

参数名称	是否必须	数据类型	描述	取值范围
balance	true	string	余额	
currency	true	string	币种	
type	true	string	类型	trade: 交易余额，frozen: 冻结余额

账户流水

API Key 权限：读取

该节点基于用户账户ID返回账户流水。

HTTP REQUEST

- GET /v1/account/history

请求参数

参数名称	是否必需	数据类型	描述	缺省值	取值范围
account-id	true	string	账户编号,取值参考 GET /v1/account/accounts		
currency	false	string	币种,即btc, ltc, bch, eth, etc ...(取值参考 GET /v1/common/currencys)		
transact-types	false	string	变动类型，可多选	all	trade (交易), transact-fee (交易手续费), deduction (手续费抵扣), transfer (划转), deposit-withdraw (充提), withdraw-fee (提币手续费), other-types (其他)

参数名称	是否必需	数据类型	描述	缺省值	取值范围
start-time	false	long	远点时间 unix time in millisecond. 以transact-time为key进行检索. 查询窗口最大为1小时. 窗口平移范围为最近30天.	((end-time) – 1hour)	(((end-time) – 1hour), (end-time)]
end-time	false	long	近点时间unix time in millisecond. 以transact-time为key进行检索. 查询窗口最大为1小时. 窗口平移范围为最近30天.	current-time	[(current-time) – 29days,(current-time)]
sort	false	string	检索方向	asc	asc or desc
size	false	int	最大条目数量	100	[1,500]

Response:

```
{
  "status": "ok",
  "data": [
    {
      "account-id": 5260185,
      "currency": "btc",
      "transact-amt": "0.002393000000000000",
      "transact-type": "transfer",
      "record-id": 89373333576,
      "avail-balance": "0.002393000000000000",
      "acct-balance": "0.002393000000000000",
      "transact-time": 1571393524526
    },
    {
      "account-id": 5260185,
      "currency": "btc",
      "transact-amt": "-0.002393000000000000",
      "transact-type": "transfer",
      "record-id": 89373382631,
      "avail-balance": "0E-18",
      "acct-balance": "0E-18",
      "transact-time": 1571393578496
    }
  ]
}
```

响应数据

字段名称	数据类型	描述	取值范围
status	string	状态码	
data	object		
{ account-id	long	账户编号	

字段名称	数据类型	描述	取值范围
currency	string	币种	
transact-amt	string	变动金额（入账为正 or 出账为负）	
transact-type	string	变动类型	
avail-balance	string	可用余额	
acct-balance	string	账户余额	
transact-time	long	交易时间（数据库记录时间）	
record-id }	string	数据库记录编号（全局唯一）	

资产划转（母子账号之间）

API Key 权限：交易

母账户执行母子账号之间的划转

HTTP 请求

- POST `/v1/subuser/transfer`

请求参数

参数	是否必填	数据类型	说明	取值范围
sub-uid	true	long	子账号uid	-
currency	true	string	币种，即btc, ltc, bch, eth, etc ...(取值参考GET /v1/common/currency)	-
amount	true	decimal	划转金额	-
type	true	string	划转类型	master-transfer-in（子账号划转给母账户虚拟币）/ master-transfer-out（母账户划转给子账号虚拟币）/master-point-transfer-in（子账号划转给母账户点卡）/master-point-transfer-out（母账户划转给子账号点卡）

Response:

```
{
  "data":123456,
  "status":"ok"
}
```

响应数据

参数	是否必填	数据类型	长度	说明	取值范围
data	true	int	-	划转订单id	-
status	true		-	状态	"OK" or "Error"

错误码

error_code	说明	类型
account-transfer-balance-insufficient-error	账户余额不足	string
base-operation-forbidden	禁止操作（母子账号关系错误时报）	string

子账号余额（汇总）

API Key 权限：读取

母账户查询其下所有子账号的各币种汇总余额

HTTP 请求

- GET /v1/subuser/aggregate-balance

请求参数

无

Response:

```
{
  "status": "ok",
  "data": [
    {
      "currency": "eos",
      "type": "spot",
      "balance": "1954559.809500000000000000"
    },
    {
      "currency": "btc",
      "type": "spot",
      "balance": "0.000000000000000000"
    },
    {
      "currency": "usdt",
      "type": "spot",
      "balance": "2925209.411300000000000000"
    },
    ...
  ]
}
```

响应数据

参数	是否必填	数据类型	长度	说明	取值范围
status	true		-	状态	"OK" or "Error"
data	true	list	-		-

- data

参数	是否必填	数据类型	长度	说明	取值范围
currency	是	string	-	子账号币名	-
type	是	string	-	账户类型	spot：现货账户，point：点卡账户
balance	是	string	-	子账号下该币种所有余额（可用余额和冻结余额的总和）	-

子账号余额

API Key 权限：读取

母账户查询子账号各币种账户余额

HTTP 请求

- GET /v1/account/accounts/{sub-uid}

请求参数

参数	是否必填	数据类型	长度	说明	取值范围
sub-uid	true	long	-	子账号的 UID	-

Response:

```
{
  "status": "ok",
  "data": [
    {
      "id": 9910049,
      "type": "spot",
      "list": [
        {
          "currency": "btc",
          "type": "trade",
          "balance": "1.00"
        },
        {
```

```
        "currency": "eth",
        "type": "trade",
        "balance": "1934.00"
    }
]
},
{
    "id": 9910050,
    "type": "point",
    "list": []
}
]
```

响应数据

参数	是否必填	数据类型	长度	说明	取值范围
id	-	long	-	子账号 UID	-
type	-	string	-	账户类型	spot：现货账户，point：点卡账户
list	-	object	-	-	-

- list

参数	是否必填	数据类型	长度	说明	取值范围
currency	-	string	-	币种	-
type	-	string	-	账户类型	trade：交易账户，frozen：冻结账户
balance	-	decimal	-	账户余额	-

钱包（充值与提现）

i 访问钱包相关的接口需要进行签名认证。

APIv2 充币地址查询

此节点用于查询特定币种（IOTA除外）在其所在区块链中的充币地址

API Key 权限：读取

i 充币地址查询暂不支持IOTA币

HTTP 请求

- GET /v2/account/deposit/address

```
curl "https://api-cloud.huobi.co.kr/v2/account/deposit/address?currency=btc"
```

请求参数

字段名称	是否必需	类型	字段描述	取值范围
currency	true	string	币种	btc, ltc, bch, eth, etc ...(取值参考GET /v1/common/currencys)

Response:

```
{
  "code": 200,
  "data": [
    {
      "currency": "btc",
      "address": "1PSRjPg53cX7hMRYAXGJnL8mqHtzmQgPUs",
      "addressTag": "",
      "chain": "btc"
    }
  ]
}
```

响应数据

字段名称	是否必需	数据类型	字段描述	取值范围
code	true	int	状态码	
message	false	string	错误描述 (如有)	
data	true	object		
{currency	true	string	币种	
address	true	string	充币地址	
addressTag	true	string	充币地址标签	
chain }	true	string	链名称	

状态码

状态码	错误信息	错误场景描述
200	success	请求成功
500	error	系统错误
1002	unauthorized	未授权

状态码	错误信息	错误场景描述
1003	invalid signature	验签失败
2002	invalid field value in "field name"	非法字段取值
2003	missing mandatory field "field name"	强制字段缺失

APIv2 提币额度查询

此节点用于查询各币种提币额度

API Key 权限：读取

HTTP 请求

- GET `/v2/account/withdraw/quota`

```
curl "https://api-cloud.huobi.co.kr/v2/account/withdraw/quota?currency=btc"
```

请求参数

字段名称	是否必需	类型	字段描述	取值范围
currency	true	string	币种	btc, ltc, bch, eth, etc ... (取值参考 <code>GET /v1/common/currencys</code>)

Response:

```
{
  "code": 200,
  "data": {
    {
      "currency": "btc",
      "chains": [
        {
          "chain": "btc",
          "maxWithdrawAmt": "200.00000000",
          "withdrawQuotaPerDay": "200.00000000",
          "remainWithdrawQuotaPerDay": "200.0000000000000000",
          "withdrawQuotaPerYear": "700000.00000000",
          "remainWithdrawQuotaPerYear": "700000.000000000000000000",
          "withdrawQuotaTotal": "700000.00000000",
          "remainWithdrawQuotaTotal": "700000.000000000000000000"
        }
      ]
    }
  ]
}
```

响应数据

字段名称	是否必需	数据类型	字段描述	取值范围
code	true	int	状态码	
message	false	string	错误描述（如有）	
data	true	object		
currency	true	string	币种	
chains	true	object		
{ chain	true	string	链名称	
maxWithdrawAmt	true	string	单次最大提币金额	
withdrawQuotaPerDay	true	string	当日提币额度	
remainWithdrawQuotaPerDay	true	string	当日提币剩余额度	
withdrawQuotaPerYear	true	string	当年提币额度	
remainWithdrawQuotaPerYear	true	string	当年提币剩余额度	
withdrawQuotaTotal	true	string	总提币额度	
remainWithdrawQuotaTotal }	true	string	总提币剩余额度	

状态码

状态码	错误信息	错误场景描述
200	success	请求成功
500	error	系统错误
1002	unauthorized	未授权
1003	invalid signature	验签失败
2002	invalid field value in "field name"	非法字段取值

虚拟币提现

API Key 权限：提币

 仅支持在官网上相应币种 地址列表 中的地址。

HTTP 请求

- POST `/v1/dw/withdraw/api/create`

```
{
  "address": "0xde709f2102306220921060314715629080e2fb77",
  "amount": "0.05",
  "currency": "eth",
  "fee": "0.01"
}
```

请求参数

参数名称	是否必须	类型	描述	取值范围
address	true	string	提现地址	仅支持在官网上相应币种地址列表 中的地址
amount	true	string	提币数量	
currency	true	string	资产类型	btc, ltc, bch, eth, etc ... (取值参考 GET /v1/common/currencys)
fee	true	string	转账手续费	
chain	false	string	取值参考 GET /v2/reference/currencies , 例如提USDT至OMNI时须设置此参数为"usdt", 提USDT至TRX时须设置此参数为"trc20usdt", 其他币种提现无须设置此参数	
addr-tag	false	string	虚拟币共享地址tag, 适用于xrp, xem, bts, steem, eos, xmr	格式, "123"类的整数字符串

Response:

```
{
  "data": 700
}
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
data	false	long	提现 ID	

取消提现

API Key 权限：提币

HTTP 请求

- POST [/v1/dw/withdraw-virtual/{withdraw-id}/cancel](#)

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
withdraw-id	true	long	提现 ID，填在 path 中		

Response:

```
{
  "data": 700
}
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
data	false	long	提现 ID	

充提记录

API Key 权限：读取

查询充提记录

HTTP 请求

- GET `/v1/query/deposit-withdraw`

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
currency	false	string	币种		btc, ltc, bch, eth, etc ... (取值参考GET /v1/common/currencys)
type	true	string	充值或提现		deposit 或 withdraw
from	false	string	查询起始 ID	缺省时，默认值direct相关。当direct为 'prev' 时，from 为1，从旧到新升序返回；当direct为' next '时，from为最新的一条记录的ID，从新到旧降序返回	
size	false	string	查询记录大小	100	1-500
direct	false	string	返回记录排序方向	缺省时，默认为 "prev" （升序）	"prev" （升序） or "next" （降序）

Response:

```
{
  "data":
  [
    {
      "id": 1171,
      "type": "deposit",
      "currency": "xrp",
      "tx-hash": "ed03094b84eafbe4bc16e7ef766ee959885ee5bcb265872baaa9c64e1cf86c2b",
      "amount": 7.457467,
      "address": "rae93V8d2mdoUQHwBDBdM4NHCMehRJAsbm",
      "address-tag": "100040",
      "fee": 0,
      "state": "safe",
      "created-at": 1510912472199,
      "updated-at": 1511145876575
    },
    ...
  ]
}
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
id	true	long		
type	true	string	类型	'deposit', 'withdraw'
currency	true	string	币种	
tx-hash	true	string	交易哈希	
chain	true	string	链名称	
amount	true	float	个数	
address	true	string	地址	
address-tag	true	string	地址标签	
fee	true	float	手续费	
state	true	string	状态	状态参见下表
created-at	true	long	发起时间	
updated-at	true	long	最后更新时间	

- 虚拟币充值状态定义：

状态	描述
unknown	状态未知
confirming	确认中

状态	描述
confirmed	确认中
safe	已完成
orphan	待确认

- 虚拟币提现状态定义：

状态	描述
submitted	已提交
reexamine	审核中
canceled	已撤销
pass	审批通过
reject	审批拒绝
pre-transfer	处理中
wallet-transfer	已汇出
wallet-reject	钱包拒绝
confirmed	区块已确认
confirm-error	区块确认错误
revealed	已撤销

现货

 访问交易相关的接口需要进行签名认证。

下单

API Key 权限：交易

发送一个新订单到火币以进行撮合。

HTTP 请求

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

```
{
  "account-id": "100009",
  "amount": "10.1",
  "price": "100.1",
  "source": "api",
  "symbol": "ethusdt",
  "type": "buy-limit",
  "client-order-id": "a0001"
}
```

请求参数

参数名称	数据类型	是否必需	默认值	描述
account-id	string	true	NA	账户 ID，取值参考 <code>GET /v1/account/accounts</code> 。现货交易使用 ‘spot’ 账户的 account-id；
symbol	string	true	NA	交易对,即btcusdt, ethbtc...（取值参考 <code>GET /v1/common/symbols</code> ）
type	string	true	NA	订单类型，包括buy-market, sell-market, buy-limit, sell-limit, buy-ioc, sell-ioc, buy-limit-maker, sell-limit-maker（说明见下文），buy-stop-limit, sell-stop-limit
amount	string	true	NA	订单交易量（市价买单此字段为订单交易额）
price	string	false	NA	limit order的交易价格
source	string	false	api	现货交易填写 “api”
client-order-id	string	false	NA	用户自编订单号（最大长度64个字符，须在24小时内保持唯一性）

buy-limit-maker

当“下单价格” >= “市场最低卖出价”，订单提交后，系统将拒绝接受此订单；
当“下单价格” < “市场最低卖出价”，提交成功后，此订单将被系统接受。

sell-limit-maker

当“下单价格” <= “市场最高买入价”，订单提交后，系统将拒绝接受此订单；
当“下单价格” > “市场最高买入价”，提交成功后，此订单将被系统接受。

Response:

```
{
  "data": "59378"
}
```

响应数据

返回的主数据对象是一个对应下单单号的字符串。
如client order ID（在24小时内）被复用，节点返回先前订单的client order ID。

撤销订单

API Key 权限：交易

此接口发送一个撤销订单的请求。

❗ 此接口只提交取消请求，实际取消结果需要通过订单状态，撮合状态等接口来确认。

HTTP 请求

- POST `/v1/order/orders/{order-id}/submitcancel`

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
order-id	true	string	订单ID，填在path中		

Response:

```
{
  "data": "59378"
}
```

响应数据

返回的主数据对象是一个对应下单单号的字符串。

错误码

Response:

```
{
  "status": "error",
  "err-code": "order-orderstate-error",
  "err-msg": "订单状态错误",
  "order-state": -1 // 当前订单状态
}
```

返回字段列表中，order-state的可能取值包括 -

order-state	Description
-1	order was already closed in the long past (order state = canceled, partial-canceled, filled, partial-filled)
5	partial-canceled
6	filled
7	canceled
10	cancelling

撤销订单（基于client order ID）

API Key 权限：交易

此接口发送一个撤销订单的请求。

❗ 此接口只提交取消请求，实际取消结果需要通过订单状态，撮合状态等接口来确认。

HTTP 请求

- POST /v1/order/orders/submitCancelClientOrder

```
{
  "client-order-id": "a0001"
}
```

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
client-order-id	true	string	用户自编订单号		

Response:

```
{
  "data": "10"
}
```

响应数据

字段名称	数据类型	描述
data	integer	撤单状态码

Status Code	Description
-1	order was already closed in the long past (order state = canceled, partial-canceled, filled, partial-filled)
0	client-order-id not found
5	partial-canceled
6	filled
7	canceled
10	cancelling

查询当前未成交订单

API Key 权限：读取

查询已提交但是仍未完全成交或未被撤销的订单。

```
{
  "account-id": "100009",
  "symbol": "ethusdt",
  "side": "buy"
}
```

HTTP 请求

- GET `/v1/order/openOrders`

请求参数

参数名称	数据类型	是否必需	默认值	描述
account-id	string	true	NA	账户 ID，取值参考 <code>GET /v1/account/accounts</code> 。现货交易使用 'spot' 账户的 account-id
symbol	string	true	NA	交易对,即btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>)
side	string	false	both	指定只返回某一个方向的订单，可能的值有: buy, sell. 默认两个方向都返回。
from	string	false		查询起始 ID
direct	string	false (如字段'from'已设定，此字段'direct'为必填)		查询方向 (prev - 以起始ID升序检索；next - 以起始ID降序检索)
size	int	false	100	返回订单的数量，最大值500。

Response:

```
{
  "data": [
    {
      "id": 5454937,
      "symbol": "ethusdt",
      "account-id": 30925,
      "amount": "1.000000000000000000",
      "price": "0.453000000000000000",
      "created-at": 1530604762277,
      "type": "sell-limit",
      "filled-amount": "0.0",
      "filled-cash-amount": "0.0",
      "filled-fees": "0.0",
      "source": "web",
    }
  ]
}
```

```
    "state": "submitted"
  }
]
}
```

响应数据

字段名称	数据类型	描述
id	integer	订单id
symbol	string	交易对, 例如btcusdt, ethbtc
price	string	limit order的交易价格
created-at	int	订单创建的调整为新加坡时间的时间戳，单位毫秒
type	string	订单类型
filled-amount	string	订单中已成交部分的数量
filled-cash-amount	string	订单中已成交部分的总价格
filled-fees	string	已交交易手续费总额
source	string	现货交易填写 “api”
state	string	订单状态，包括submitted, partial-filled, cancelling, created
stop-price	string	止盈止损订单触发价格
operator	string	止盈止损订单触发价运算符

批量撤销订单（open orders）

API Key 权限：交易

此接口发送批量撤销订单的请求。

❗ 此接口只提交取消请求，实际取消结果需要通过订单状态，撮合状态等接口来确认。

HTTP 请求

- POST /v1/order/orders/batchCancelOpenOrders

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
------	------	----	----	-----	------

参数名称	是否必须	类型	描述	默认值	取值范围
account-id	true	string	账户ID，取值参考 <code>GET /v1/account/accounts</code>		
symbol	false	string	交易代码列表（最多10个symbols，多个交易代码间以逗号分隔），btcusdt, ethbtc...（取值参考 <code>/v1/common/symbols</code> ）	all	
side	false	string	主动交易方向		"buy" 或 "sell"，缺省将返回所有符合条件尚未成交订单
size	false	int	所需返回记录数	100	[0,100]

Response:

```
{
  "status": "ok",
  "data": {
    "success-count": 2,
    "failed-count": 0,
    "next-id": 5454600
  }
}
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
success-count	true	int	成功取消的订单数	
failed-count	true	int	取消失败的订单数	
next-id	true	long	下一个符合取消条件的订单号	

批量撤销订单

API Key 权限：交易

此接口同时为多个订单（基于id）发送取消请求。

HTTP 请求

- POST `/v1/order/orders/batchcancel`

```
{
  "order-ids": [
    "1", "2", "3"
  ]
}
```

```
    ]
  }
}
```

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
order-ids	true	list	撤销订单ID列表		单次不超过50个订单id

Response:

```
{
  "data": {
    "success": [
      "1",
      "3"
    ],
    "failed": [
      {
        "err-msg": "记录无效",
        "order-id": "2",
        "err-code": "base-record-invalid"
        "order-state": -1 // 当前订单状态
      }
    ]
  }
}
```

响应数据

字段名称	数据类型	描述
data	map	撤单结果

错误码

Response:

```
{
  "status": "ok",
  "data": {
    "success": ["123","456"],
    "failed": [
      {
        "err-msg": "订单状态错误",
        "order-id": "12345678",
        "err-code": "order-orderstate-error",
        "order-state": -1 // 当前订单状态
      }
    ]
  }
}
```

返回字段列表中，order-state的可能取值包括 -

order-state	Description
-1	order was already closed in the long past (order state = canceled, partial-canceled, filled, partial-filled)
5	partial-canceled
6	filled
7	canceled
10	cancelling

查询订单详情

API Key 权限：读取

此接口返回指定订单的最新状态和详情。

HTTP 请求

- GET `/v1/order/orders/{order-id}`

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
order-id	true	string	订单ID，填在path中		

Response:

```
{
  "data":
  {
    "id": 59378,
    "symbol": "ethusdt",
    "account-id": 100009,
    "amount": "10.1000000000",
    "price": "100.1000000000",
    "created-at": 1494901162595,
    "type": "buy-limit",
    "field-amount": "10.1000000000",
    "field-cash-amount": "1011.0100000000",
    "field-fees": "0.0202000000",
    "finished-at": 1494901400468,
    "user-id": 1000,
    "source": "api",
    "state": "filled",
    "canceled-at": 0
  }
}
```

响应数据

字段名称	是否必须	数据类型	描述	取值范围
account-id	true	long	账户 ID	
amount	true	string	订单数量	
canceled-at	false	long	订单撤销时间	
created-at	true	long	订单创建时间	
field-amount	true	string	已成交数量	
field-cash-amount	true	string	已成交总金额	
field-fees	true	string	已成交手续费（买入为币，卖出为钱）	
finished-at	false	long	订单变为终结态的时间，不是成交时间，包含“已撤单”状态	
id	true	long	订单ID	
price	true	string	订单价格	
source	true	string	订单来源	api
state	true	string	订单状态	submitted 已提交, partial-filled 部分成交, partial-canceled 部分成交撤销, filled 完全成交, canceled 已撤销，created
symbol	true	string	交易对	btcusdt, ethbtc, rcneth ...
type	true	string	订单类型	buy-market：市价买, sell-market：市价卖, buy-limit：限价买, sell-limit：限价卖, buy-ioc：IOC买单, sell-ioc：IOC卖单，buy-limit-maker, sell-limit-maker, buy-stop-limit，sell-stop-limit
stop-price	false	string	止盈止损订单触发价格	
operator	false	string	止盈止损订单触发价运算符	gte,lte

查询订单详情（基于client order ID）

API Key 权限：读取

此接口返回指定订单的最新状态和详情。

HTTP 请求

- GET /v1/order/orders/getClientOrder

```
{
  "clientOrderId": "a0001"
}
```

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
clientOrderId	true	string	用户自编订单号		

Response:

```
{
  "data":
  {
    "id": 59378,
    "symbol": "ethusdt",
    "account-id": 100009,
    "amount": "10.1000000000",
    "price": "100.1000000000",
    "created-at": 1494901162595,
    "type": "buy-limit",
    "field-amount": "10.1000000000",
    "field-cash-amount": "1011.0100000000",
    "field-fees": "0.0202000000",
    "finished-at": 1494901400468,
    "user-id": 1000,
    "source": "api",
    "state": "filled",
    "canceled-at": 0
  }
}
```

响应数据

字段名称	是否必须	数据类型	描述	取值范围
account-id	true	long	账户 ID	
amount	true	string	订单数量	
canceled-at	false	long	订单撤销时间	
created-at	true	long	订单创建时间	

字段名称	是否必须	数据类型	描述	取值范围
field-amount	true	string	已成交数量	
field-cash-amount	true	string	已成交总金额	
field-fees	true	string	已成交手续费（买入为币，卖出为钱）	
finished-at	false	long	订单变为终结态的时间，不是成交时间，包含“已撤单”状态	
id	true	long	订单ID	
price	true	string	订单价格	
source	true	string	订单来源	api
state	true	string	订单状态	submitted 已提交, partial-filled 部分成交, partial-canceled 部分成交撤销, filled 完全成交, canceled 已撤销, created
symbol	true	string	交易对	btcusdt, ethbtc, rcneth ...
type	true	string	订单类型	buy-market：市价买, sell-market：市价卖, buy-limit：限价买, sell-limit：限价卖, buy-ioc：IOC买单, sell-ioc：IOC卖单, buy-limit-maker, sell-limit-maker, buy-stop-limit, sell-stop-limit
stop-price	false	string	止盈止损订单触发价格	
operator	false	string	止盈止损订单触发价运算符	gte,lte

如client order ID不存在，返回如下错误信息 { "status": "error", "err-code": "base-record-invalid", "err-msg": "record invalid", "data": null }

成交明细

API Key 权限：读取

此接口返回指定订单的成交明细。

HTTP 请求

- GET `/v1/order/orders/{order-id}/matchresults`

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
------	------	----	----	-----	------

参数名称	是否必须	类型	描述	默认值	取值范围
order-id	true	string	订单ID，填在path中		

Response:

```
{
  "data": [
    {
      "id": 29553,
      "order-id": 59378,
      "match-id": 59335,
      "trade-id": 100282808529,
      "symbol": "ethusdt",
      "type": "buy-limit",
      "source": "api",
      "price": "100.1000000000",
      "filled-amount": "9.1155000000",
      "filled-fees": "0.0182310000",
      "created-at": 1494901400435,
      "role": "maker",
      "filled-points": "0.0",
      "fee-deduct-currency": ""
    }
    ...
  ]
}
```

响应数据

❗ 返回的主数据对象为一个对象数组，其中每一个元件代表一个交易结果。

字段名称	是否必须	数据类型	描述	取值范围
created-at	true	long	成交时间	
filled-amount	true	string	成交数量	
filled-fees	true	string	成交手续费，如果为空或为0，代表使用了其他币种进行了抵扣，可根据filled-points和fee-deduct-currency字段判断	
id	true	long	订单成交记录ID	
match-id	true	long	撮合ID，订单在撮合中执行的顺序ID	
order-id	true	long	订单ID，成交所属订单的ID	

字段名称	是否必须	数据类型	描述	取值范围
trade-id	false	integer	Unique trade ID (NEW)唯一成交编号，成交时产生的唯一编号ID	
price	true	string	成交价格	
source	true	string	订单来源	api
symbol	true	string	交易对	btcusdt, ethbtc, rcneth ...
type	true	string	订单类型	buy-market：市价买, sell-market：市价卖, buy-limit：限价买, sell-limit：限价卖, buy-ioc：IOC买单, sell-ioc：IOC卖单， buy-limit-maker, sell-limit-maker, buy-stop-limit， sell-stop-limit
role	true	string	成交角色	maker,taker
filled-points	true	string	抵扣数量（可为ht或hbpoint）	
fee-deduct-currency	true	string	抵扣类型	如果为空，代表扣除的手续费是原币；如果为"ht"，代表抵扣手续费的是HT；如果为"hbpoint"，代表抵扣手续费的是点卡

搜索历史订单

API Key 权限：读取

此接口基于搜索条件查询历史订单。

HTTP 请求

- GET `/v1/order/orders`

```
{
  "account-id": "100009",
  "amount": "10.1",
  "price": "100.1",
  "source": "api",
  "symbol": "ethusdt",
  "type": "buy-limit"
}
```

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
symbol	true	string	交易对		btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>)

参数名称	是否必须	类型	描述	默认值	取值范围
types	false	string	查询的订单类型组合, 使用','分割		buy-market : 市价买, sell-market : 市价卖, buy-limit : 限价买, sell-limit : 限价卖, buy-ioc : IOC买单, sell-ioc : IOC卖单, buy-limit-maker, sell-limit-maker, buy-stop-limit , sell-stop-limit
start-date	false	string	查询开始日期, 日期格式yyyy-mm-dd。以订单生成时间进行查询	-1d 查询结束日期的前1天	取值范围 [((end-date) – 1), (end-date)] , 查询窗口最大为2天, 窗口平移范围为最近180天, 已完全撤销的历史订单的查询窗口平移范围只有最近7天(state="canceled")
end-date	false	string	查询结束日期, 日期格式yyyy-mm-dd。以订单生成时间进行查询	today	取值范围 [(today-179), today] , 查询窗口最大为2天, 窗口平移范围为最近180天, 已完全撤销的历史订单的查询窗口平移范围只有最近7天 (state="canceled")
states	true	string	查询的订单状态组合, 使用','分割		submitted 已提交, partial-filled 部分成交, partial-canceled 部分成交撤销, filled 完全成交, canceled 已撤销, created
from	false	string	查询起始 ID		
direct	false	string	查询方向		prev 向前, 时间 (或 ID) 正序; next 向后, 时间 (或 ID) 倒序)
size	false	string	查询记录大小	100	[1, 100]

Response:

```
{
  "data": [
    {
      "id": 59378,
      "symbol": "ethusdt",
      "account-id": 100009,
      "amount": "10.1000000000",
      "price": "100.1000000000",
      "created-at": 1494901162595,
      "type": "buy-limit",
      "field-amount": "10.1000000000",
      "field-cash-amount": "1011.0100000000",
      "field-fees": "0.0202000000",
      "finished-at": 1494901400468,
      "user-id": 1000,
      "source": "api",
      "state": "filled",
      "canceled-at": 0
    }
    ...
  ]
}
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
------	------	------	----	------

参数名称	是否必须	数据类型	描述	取值范围
account-id	true	long	账户 ID	
amount	true	string	订单数量	
canceled-at	false	long	接到撤单申请的时间	
created-at	true	long	订单创建时间	
field-amount	true	string	已成交数量	
field-cash-amount	true	string	已成交总金额	
field-fees	true	string	已成交手续费 (买入为基础币，卖出为计价币)	
finished-at	false	long	最后成交时间	
id	true	long	订单ID	
price	true	string	订单价格	
source	true	string	订单来源	api
state	true	string	订单状态	submitted 已提交, partial-filled 部分成交, partial-canceled 部分成交撤销, filled 完全成交, canceled 已撤销 , created
symbol	true	string	交易对	btcusdt, ethbtc, rcneth ...
type	true	string	订单类型	submit-cancel : 已提交撤单申请 ,buy-market : 市价买, sell-market : 市价卖, buy-limit : 限价买, sell-limit : 限价卖, buy-ioc : IOC买单, sell-ioc : IOC卖单 , buy-limit-maker, sell-limit-maker, buy-stop-limit , sell-stop-limit
stop-price	false	string	止盈止损订单触发价格	
operator	false	string	止盈止损订单触发价运算符	gte,lte

START-DATE, END-DATE相关错误码

错误码	对应错误场景
invalid_interval	start date小于end date; 或者 start date 与end date之间的时间间隔大于2天
invalid_start_date	start date是一个180天之前的日期；或者start date是一个未来的日期

错误码	对应错误场景
invalid_end_date	end date 是一个180天之前的日期；或者end date是一个未来的日期

搜索最近48小时内历史订单

API Key 权限：读取

此接口基于搜索条件查询最近48小时内历史订单。

HTTP 请求

- GET `/v1/order/history`

```
{
  "symbol": "btcusdt",
  "start-time": "1556417645419",
  "end-time": "1556533539282",
  "direct": "prev",
  "size": "10"
}
```

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
symbol	false	string	交易对	all	btcusdt, ethbtc... (取值参考GET <code>/v1/common/symbols</code>)
start-time	false	long	查询起始时间（含）	48小时前的时刻	UTC time in millisecond
end-time	false	long	查询结束时间（含）	查询时刻	UTC time in millisecond
direct	false	string	订单查询方向（注：仅在检索出的总条目数量超出size字段限定时起作用；如果检索出的总条目数量在size 字段限定内，direct 字段不起作用。）	next	prev, next
size	false	int	每次返回条目数量	100	[10,1000]

Response:

```
{
  "status": "ok",
  "data": [
    {
```

```
    "id": 31215214553,
    "symbol": "btcusdt",
    "account-id": 4717043,
    "amount": "1.000000000000000000",
    "price": "1.000000000000000000",
    "created-at": 1556533539282,
    "type": "buy-limit",
    "field-amount": "0.0",
    "field-cash-amount": "0.0",
    "field-fees": "0.0",
    "finished-at": 1556533568953,
    "source": "web",
    "state": "canceled",
    "canceled-at": 1556533568911
  }
]
```

响应数据

参数名称	是否必须	数据类型	描述	取值范围
{account-id	true	long	账户 ID	
amount	true	string	订单数量	
canceled-at	false	long	接到撤单申请的时间	
created-at	true	long	订单创建时间	
field-amount	true	string	已成交数量	
field-cash-amount	true	string	已成交总金额	
field-fees	true	string	已成交手续费（买入为基础币，卖出为计价币）	
finished-at	false	long	最后成交时间	
id	true	long	订单ID	
price	true	string	订单价格	
source	true	string	订单来源	api
state	true	string	订单状态	partial-canceled 部分成交撤销, filled 完全成交, canceled 已撤销
symbol	true	string	交易对	btcusdt, ethbtc, rcneth ...

参数名称	是否必须	数据类型	描述	取值范围
stop-price	false	string	止盈止损订单触发价格	
operator	false	string	止盈止损订单触发价运算符	gte,lte
type}	true	string	订单类型	buy-market：市价买, sell-market：市价卖, buy-limit：限价买, sell-limit：限价卖, buy-ioc：IOC买单, sell-ioc：IOC卖单, buy-limit-maker, sell-limit-maker, buy-limit-maker, sell-limit-maker
next-time	false	long	下一查询起始时间（当请求字段“direct”为“prev”时有效），下一查询结束时间（当请求字段“direct”为“next”时有效）。注：仅在检索出的总条目数量超出size字段限定时，此返回字段存在。	UTC time in millisecond

当前和历史成交

API Key 权限：读取

此接口基于搜索条件查询当前和历史成交记录。

HTTP 请求

- GET `/v1/order/matchresults`

请求参数

参数名称	是否必须	类型	描述	默认值	取值范围
symbol	true	string	交易对	NA	btcusdt, ethbtc...（取值参考 <code>GET /v1/common/symbols</code> ）
types	false	string	查询的订单类型组合，使用','分割	all	buy-market：市价买, sell-market：市价卖, buy-limit：限价买, sell-limit：限价卖, buy-ioc：IOC买单, sell-ioc：IOC卖单, buy-limit-maker, sell-limit-maker, buy-stop-limit, sell-stop-limit
start-date	false	string	查询开始日期, 日期格式 yyyy-mm-dd	-1d 查询结束日期的前1天	取值范围 $[(end-date) - 1, (end-date)]$ ，查询窗口最大为2天，窗口平移范围为最近61天
end-date	false	string	查询结束日期, 日期格式 yyyy-mm-dd	today	取值范围 $[(today-60), today]$ ，查询窗口最大为2天，窗口平移范围为最近61天
from	false	string	查询起始 ID	订单成交记录ID（最大值）	

参数名称	是否必须	类型	描述	默认值	取值范围
direct	false	string	查询方向	默认 next，成交记录 ID 由大到小排序	prev 向前，时间（或 ID）正序；next 向后，时间（或 ID）倒序）
size	false	string	查询记录大小	100	[1, 100]

Response:

```
{
  "data": [
    {
      "id": 29553,
      "order-id": 59378,
      "match-id": 59335,
      "symbol": "ethusdt",
      "type": "buy-limit",
      "source": "api",
      "price": "100.1000000000",
      "filled-amount": "9.1150000000",
      "filled-fees": "0.0182310000",
      "created-at": 1494901400435,
      "trade-id": 100282808529,
      "role": taker,
      "filled-points": "0.0",
      "fee-deduct-currency": ""
    }
    ...
  ]
}
```

响应数据

❗ 返回的主数据对象为一个对象数组，其中每一个元件代表一个交易结果。

参数名称	是否必须	数据类型	描述	取值范围
created-at	true	long	成交时间	
filled-amount	true	string	成交数量	
filled-fees	true	string	成交手续费	

参数名称	是否必须	数据类型	描述	取值范围
id	true	long	订单成交记录ID	
match-id	true	long	撮合 ID	
order-id	true	long	订单 ID	
trade-id	false	integer	Unique trade ID (NEW)唯一成交编号	
price	true	string	成交价格	
source	true	string	订单来源	api
symbol	true	string	交易对	btcusdt, ethbtc, rcneth ...
type	true	string	订单类型	buy-market : 市价买, sell-market : 市价卖, buy-limit : 限价买, sell-limit : 限价卖, buy-ioc : IOC买单, sell-ioc : IOC卖单 , buy-limit-maker, sell-limit-maker, buy-stop-limit , sell-stop-limit
role	true	string	成交角色	maker,taker
filled-points	true	string	抵扣数量 (hbpoint)	
fee-deduct-currency	true	string	抵扣类型	hbpoint

START-DATE, END-DATE相关错误码

错误码	对应错误场景
invalid_interval	start date/小于end date; 或者 start date 与end date之间的时间间隔大于2天
invalid_start_date	start date是一个61天之前的日期；或者start date是一个未来的日期
invalid_end_date	end date 是一个61天之前的日期；或者end date是一个未来的日期

获取用户当前手续费率

Api用户查询交易对费率，一次限制最多查10个交易对，子用户的费率和母用户保持一致

API Key 权限：读取

```
curl "https://api-cloud.huobi.co.kr/v1/fee/fee-rate/get?symbols=btcusdt,ethusdt,ltcusdt"
```

HTTP 请求

- GET `/v1/fee/fee-rate/get`

请求参数

参数	数据类型	是否必须	默认值	描述	取值范围
symbols	string	true	NA	交易对，可多填，逗号分隔	btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>)

Response:

```
{
  "status": "ok",
  "data": [
    {
      "symbol": "btcusdt",
      "maker-fee": "0.0001",
      "taker-fee": "0.0002"
    },
    {
      "symbol": "ethusdt",
      "maker-fee": "0.002",
      "taker-fee": "0.002"
    },
    {
      "symbol": "ltcusdt",
      "maker-fee": "0.0015",
      "taker-fee": "0.0018"
    }
  ]
}
```

响应数据

字段名称	数据类型	是否强制	描述
status	string	Y	响应状态
err-code	string	N	响应码
err-msg	string	N	响应信息
data	list	Y	交易对费率列表

LIST

属性	数据类型	说明
symbol	string	交易对
maker-fee	string	挂单手续费率
taker-fee	string	吃单手续费率

响应码

响应码	说明	类型	备注
base-symbol-error	无效的交易对	string	-
base-too-many-symbol	最大支持 10 个交易对	string	-

Websocket行情数据

简介

接入URL

韩国站行情请求地址

wss://api-cloud.huobi.co.kr/ws

请使用中国大陆以外的服务器访问火币 API

数据压缩

WebSocket API 返回的所有数据都进行了 GZIP 压缩，需要 client 在收到数据之后解压。

心跳消息

当用户的Websocket客户端连接到火币Websocket服务器后，服务器会定期（当前设为5秒）向其发送ping消息并包含一整数数值如下：

{ "ping": 1492420473027 } （行情）

当用户的Websocket客户端接收到此心跳消息后，应返回pong消息并包含同一整数数值：

{ "pong": 1492420473027 } （行情）

❗ 当Websocket服务器连续两次发送了`ping`消息却没有收到任何一次`pong`消息返回后，服务器将主动断开与此客户端的连接。

订阅主题

成功建立与Websocket服务器的连接后，Websocket客户端发送如下请求以订阅特定主题：

```
{
  "sub": "market.btcusdt.kline.1min",
  "id": "id1"
}
```

{ "sub": "topic to sub", "id": "id generate by client" }

成功订阅后，Websocket客户端将收到确认：

```
{
  "id": "id1",
```

```
{
  "status": "ok",
  "subbed": "market.btcusdt.kline.1min",
  "ts": 1489474081631
}
```

之后,一旦所订阅的主题有更新,Websocket客户端将收到服务器推送的更新消息(push):

```
{
  "ch": "market.btcusdt.kline.1min",
  "ts": 1489474082831,
  "tick": {
    "id": 1489464480,
    "amount": 0.0,
    "count": 0,
    "open": 7962.62,
    "close": 7962.62,
    "low": 7962.62,
    "high": 7962.62,
    "vol": 0.0
  }
}
```

取消订阅

取消订阅的格式如下:

```
{
  "unsub": "market.btcusdt.trade.detail",
  "id": "id4"
}
```

{ "unsub": "topic to unsub", "id": "id generate by client" }

取消订阅成功确认:

```
{
  "id": "id4",
  "status": "ok",
  "unsubbed": "market.btcusdt.trade.detail",
  "ts": 1494326028889
}
```

请求数据

Websocket服务器同时支持一次性请求数据(pull)。

请求数据的格式如下:

```
{
  "req": "market.ethbtc.kline.1min",
  "id": "id10"
}
```

```
{ "req": "topic to req", "id": "id generate by client" }
```

一次性返回的数据：

```
{
  "status": "ok",
  "rep": "market.btcusdt.kline.1min",
  "data": [
    {
      "amount": 1.6206,
      "count": 3,
      "id": 1494465840,
      "open": 9887.00,
      "close": 9885.00,
      "low": 9885.00,
      "high": 9887.00,
      "vol": 16021.632026
    },
    {
      "amount": 2.2124,
      "count": 6,
      "id": 1494465900,
      "open": 9885.00,
      "close": 9880.00,
      "low": 9880.00,
      "high": 9885.00,
      "vol": 21859.023500
    }
  ]
}
```

K线数据

主题订阅

一旦K线数据产生，Websocket服务器将通过此订阅主题接口推送至客户端：

market.\$symbol\$.kline.\$period\$

订阅请求

```
{
  "sub": "market.ethbtc.kline.1min",
  "id": "id1"
}
```

参数

参数	数据类型	是否必需	描述	取值范围
----	------	------	----	------

参数	数据类型	是否必需	描述	取值范围
symbol	string	true	交易代码	btcusdt, ethbtc... (取值参考 GET /v1/common/symbols)
period	string	true	K线周期	1min, 5min, 15min, 30min, 60min, 4hour, 1day, 1mon, 1week, 1year

Response

```
{
  "id": "id1",
  "status": "ok",
  "subbed": "market.ethbtc.kline.1min",
  "ts": 1489474081631
}
```

Update example

```
{
  "ch": "market.ethbtc.kline.1min",
  "ts": 1489474082831,
  "tick": {
    "id": 1489464480,
    "amount": 0.0,
    "count": 0,
    "open": 7962.62,
    "close": 7962.62,
    "low": 7962.62,
    "high": 7962.62,
    "vol": 0.0
  }
}
```

数据更新字段列表

字段	数据类型	描述
id	integer	unix时间，同时作为K线ID
amount	float	成交量
count	integer	成交笔数
open	float	开盘价
close	float	收盘价（当K线为最晚的一根时，是最新成交价）
low	float	最低价
high	float	最高价
vol	float	成交额, 即 sum(每一笔成交价 * 该笔的成交量)

 当symbol被设为 “hb10” 或 “huobi10” 时，amount，count，vol均为零值。

数据请求

用请求方式一次性获取K线数据需要额外提供以下参数：（每次最多返回300条）

```
{
  "req": "market.$symbol.kline.$period",
  "id": "id generated by client",
  "from": "from time in epoch seconds",
  "to": "to time in epoch seconds"
}
```

参数	数据类型	是否必需	缺省值	描述	取值范围
from	integer	false	1501174800(2017-07-28T00:00:00+08:00)	起始时间 (epoch time in second)	[1501174800, 2556115200]
to	integer	false	2556115200(2050-01-01T00:00:00+08:00)	结束时间 (epoch time in second)	[1501174800, 2556115200] or (\$from, 2556115200] if "from" is set

市场深度行情数据

此主题发送最新市场深度快照。快照频率为每秒1次。

主题订阅

```
market.$symbol.depth.$type
```

Subscribe request

```
{
  "sub": "market.btcusdt.depth.step0",
  "id": "id1"
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
symbol	string	true	NA	交易代码	btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>)
type	string	true	step0	合并深度类型	step0, step1, step2, step3, step4, step5

"type" 合并深度类型

Value	Description
-------	-------------

Value	Description
step0	不合并深度
step1	Aggregation level = precision*10
step2	Aggregation level = precision*100
step3	Aggregation level = precision*1000
step4	Aggregation level = precision*10000
step5	Aggregation level = precision*100000

当type值为 'step0' 时，默认深度为150档; 当type值为 'step1' ， 'step2' ， 'step3' ， 'step4' ， 'step5' 时，默认深度为20档。

Response

```
{
  "id": "id1",
  "status": "ok",
  "subbed": "market.btcusdt.depth.step0",
  "ts": 1489474081631
}
```

Update example

```
{
  "ch": "market.htusdt.depth.step0",
  "ts": 1572362902027,
  "tick": {
    "bids": [
      [3.7721, 344.86], // [price, quote volume]
      [3.7709, 46.66]
    ],
    "asks": [
      [3.7745, 15.44],
      [3.7746, 70.52]
    ],
    "version": 100434317651,
    "ts": 1572362902012
  }
}
```

数据更新字段列表

❗ 在'tick'object下方呈现买盘卖盘深度列表

字段	数据类型	描述
bids	object	当前的所有买单 [price, quote volume]

字段	数据类型	描述
asks	object	当前的所有卖单 [price, quote volume]
version	integer	内部字段
ts	integer	新加坡时间的时间戳，单位毫秒

i 当symbol被设为"hb10"时，amount, count, vol均为零值

数据请求

支持数据请求方式一次性获取市场深度数据：

```
{
  "req": "market.btcusdt.depth.step0",
  "id": "id10"
}
```

买一卖一逐笔行情

当买一价、买一量、卖一价、卖一量，其中任一数据发生变化时，此主题推送逐笔更新。

主题订阅

market.\$symbol.bbo

Subscribe request

```
{
  "sub": "market.btcusdt.bbo",
  "id": "id1"
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
symbol	string	true	NA	交易代码	btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>)

Response

```
{
  "id": "id1",
  "status": "ok",
  "subbed": "market.btcusdt.bbo",
  "ts": 1489474081631
}
```

Update example

```
{
  "ch": "market.btcusdt.bbo",
  "ts": 1489474082831,
  "tick": {
    "symbol": "btcusdt",
    "quoteTime": "1489474082811",
    "bid": "10008.31",
    "bidSize": "0.01",
    "ask": "10009.54",
    "askSize": "0.3"
  }
}
```

数据更新字段列表

字段	数据类型	描述
symbol	string	交易代码
quoteTime	long	盘口更新时间
bid	float	买一价
bidSize	float	买一量
ask	float	卖一价
askSize	float	卖一量

成交明细

主题订阅

此主题提供市场最新成交逐笔明细。

```
market.$symbol.trade.detail
```

Subscribe request

```
{
  "sub": "market.btcusdt.trade.detail",
  "id": "id1"
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
----	------	------	-----	----	------

参数	数据类型	是否必需	缺省值	描述	取值范围
symbol	string	true	NA	交易代码	btcusdt, ethbtc... (取值参考 GET /v1/common/symbols)

Response

```
{
  "id": "id1",
  "status": "ok",
  "subbed": "market.btcusdt.trade.detail",
  "ts": 1489474081631
}
```

Update example

```
{
  "ch": "market.btcusdt.trade.detail",
  "ts": 1489474082831,
  "tick": {
    "id": 14650745135,
    "ts": 1533265950234,
    "data": [
      {
        "amount": 0.0099,
        "ts": 1533265950234,
        "id": 146507451359183894799,
        "tradeId": 102043494568,
        "price": 401.74,
        "direction": "buy"
      }
      // more Trade Detail data here
    ]
  }
}
```

数据更新字段列表

字段	数据类型	描述
id	integer	唯一成交ID (将被废弃)
tradeId	integer	唯一成交ID (NEW)
amount	float	成交量
price	float	成交价
ts	integer	成交时间 (UNIX epoch time in millisecond)
direction	string	成交主动方 (taker的订单方向): 'buy' or 'sell'

数据请求

支持数据请求方式一次性获取成交明细数据（仅能获取最多最近300个成交记录）：

```
{
  "req": "market.btcusdt.trade.detail",
  "id": "id11"
}
```

市场概要

主题订阅

此主题提供24小时内最新市场概要快照。快照频率不超过每秒10次。

market.\$symbol.detail

Subscribe request

```
{
  "sub": "market.btcusdt.detail",
  "id": "id1"
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
symbol	string	true	NA	交易代码	btcusdt, ethbtc... (取值参考GET /v1/common/symbols)

Response

```
{
  "id": "id1",
  "status": "ok",
  "subbed": "market.btcusdt.detail",
  "ts": 1489474081631
}
```

Update example

```
"tick": {
  "amount": 12224.2922,
  "open": 9790.52,
  "close": 10195.00,
  "high": 10300.00,
  "ts": 1494496390000,
  "id": 1494496390,
  "count": 15195,
  "low": 9657.00,
```

```
    "vol":      121906001.754751
  }
```

数据更新字段列表

字段	数据类型	描述
id	integer	unix时间，同时作为消息ID
ts	integer	unix系统时间
amount	float	24小时成交量
count	integer	24小时成交笔数
open	float	24小时开盘价
close	float	最新价
low	float	24小时最低价
high	float	24小时最高价
vol	float	24小时成交额

数据请求

支持数据请求方式一次性获取市场概要数据：

```
{
  "req": "market.btcusdt.detail",
  "id": "id11"
}
```

Websocket资产及订单

简介

接入URL

Websocket资产及订单

```
wss://api-cloud.huobi.co.kr/ws/v1
```

请使用中国大陆以外的服务器访问火币 API。

数据压缩

WebSocket API 返回的所有数据都进行了 GZIP 压缩，需要 client 在收到数据之后解压。

心跳消息

2019/07/08之后

当用户的Websocket客户端连接到火币Websocket服务器后，服务器会定期（当前设为20秒）向其发送ping消息并包含一整数数值如下：

```
{ "op": "ping", "ts": 1492420473027 } (资产订单推送)
```

当用户的Websocket客户端接收到此心跳消息后，应返回pong消息并包含同一整数数值：

```
{ "op": "pong", "ts": 1492420473027 } (资产订单推送)
```

❗ 当Websocket服务器连续三次发送了`ping`消息却没有收到任何一次`pong`消息返回后，服务器将主动断开与此客户端的连接。

2019/07/08之前

当用户的Websocket客户端连接到火币Websocket服务器后，服务器会定期（设为30秒）向其发送ping消息并包含一整数数值如下：

```
{ "op": "ping", "ts": 1492420473027 }
```

当用户的Websocket客户端接收到此心跳消息后，应返回pong消息并包含同一整数数值：

```
{ "op": "pong", "ts": 1492420473027 }
```

❗ 当Websocket服务器连续两次发送了`ping`消息却没有收到任何一次`pong`消息返回后，服务器将主动断开与此客户端的连接。

订阅主题

成功建立与Websocket服务器的连接后，Websocket客户端发送如下请求以订阅特定主题：

```
{
  "op": "operation type, 'sub' for subscription, 'unsub' for unsubscription",
  "topic": "topic to sub",
  "cid": "id generate by client"
}
```

成功订阅后，Websocket客户端将收到确认：

```
{
  "op": "operation type, refer to the operation which triggers this response",
  "cid": "id1",
  "error-code": 0, // 0 for no error
  "topic": "topic to sub if the op is sub",
  "ts": 1489474081631
}
```

之后，一旦所订阅的主题有更新，Websocket客户端将收到服务器推送的更新消息（push）：

```
{
  "op": "notify",
  "topic": "topic of this notify",
  "ts": 1489474082831,
  "data": {
    // data of specific topic update
  }
}
```

```
}  
}
```

取消订阅

取消订阅的格式如下：

```
{  
  "op": "unsub",  
  "topic": "accounts",  
  "cid": "client generated id"  
}
```

取消订阅成功确认：

```
{  
  "op": "unsub",  
  "topic": "accounts",  
  "cid": "id generated by client",  
  "err-code": 0,  
  "ts": 1489474081631  
}
```

请求数据

Websocket服务器同时支持一次性请求数据（pull）。

当与Websocket服务器成功建立连接后，以下三个主题可供用户请求：

- accounts.list
- orders.list
- orders.detail

具体请求方式请见后文。

限频

数据请求（sub）限频规则

限频规则基于API key而不是连接。当sub数量超出限值时，Websocket客户端将收到"too many request"错误码。具体规则如下：

单个连接每秒最多50次sub和50次unsub。单个连接sub总量限制100个，sub总量达到限额后不允许再sub，但每次unsub可以减少sub总量的计数。比如：30个sub后unsub 1个，此时sub总量count为29，还有71个sub总量可用。

数据请求（req）限频规则

限频规则基于API key而不是连接。当请求频率超出限值时，Websocket客户端将收到"too many request"错误码。以下为各主题当前限频设定：

- accounts.list: once every 2 seconds
- orders.list AND orders.detail: once every 5 seconds

鉴权

资产及订单主题鉴权请求数据格式如下：

```
{
  "op": "auth",
  "AccessKeyId": "e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx",
  "SignatureMethod": "HmacSHA256",
  "SignatureVersion": "2",
  "Timestamp": "2017-05-11T15:19:30",
  "Signature": "4F65x5A2bLyMWVQj3Aqp+B4w+ivaA7n50i2SuYtCJ9o=",
}
```

鉴权请求数据格式说明

filed	type	instruction
op	string	必填；操作名称，鉴权固定值为 auth；
cid	string	选填；Client 请求唯一 ID
AccessKeyId	string	必填；您申请的 API Key 中的 AccessKey
SignatureMethod	string	必填；签名方法, 用户计算签名的基于哈希的协议，此处使用 HmacSHA256
SignatureVersion	string	必填；签名协议的版本，此处使用 2
Timestamp	string	必填；时间戳, 您发出请求的时间 (UTC 时区) (UTC 时区) (UTC 时区)。在查询请求中包含此值有助于防止第三方截取您的请求。如：2017-05-11T16:22:06。再次强调是 (UTC 时区)
Signature	string	必填；签名, 计算得出的值，用于确保签名有效和未被篡改

注： - 参考[<https://huobiapi.github.io/docs/spot/v1/cn/#c64cd15fdc>] 生成有效签名 - 签名计算中请求方法固定值为GET]

订阅账户更新

API Key 权限：读取

订阅账户资产变动更新。

主题订阅

accounts

Subscribe request

```
{
  "op": "sub",
  "cid": "40sG903yz80oDFWr",
  "topic": "accounts",
  "model": "0"
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
model	string	false	0	是否包含已冻结余额	1 to include frozen balance, 0 to not

i 如果同时订阅可用和总余额，需要为 0 和 1 各开启一条websocket连接

Response

```
{
  "op": "sub",
  "cid": "40sG903yz80oDFWr",
  "err-code": 0,
  "ts": 1489474081631,
  "topic": "accounts"
}
```

Update example

```
{
  "op": "notify",
  "ts": 1522856623232,
  "topic": "accounts",
  "data": {
    "event": "order.place",
    "list": [
      {
        "account-id": 419013,
        "currency": "usdt",
        "type": "trade",
        "balance": "500009195917.4362872650"
      }
    ]
  }
}
```

数据更新字段列表

字段	数据类型	描述
event	string	资产变化通知相关事件说明，比如订单创建(order.place)、订单成交(order.match)、订单成交退款 (order.refund)、订单撤销(order.cancel)、点卡抵扣交易手续费 (order.fee-refund)、其他资产变化(other)
account-id	integer	账户 id
currency	string	币种
type	string	账户类型, 交易子账户 (trade)

字段	数据类型	描述
balance	string	账户余额 (当订阅model=0时，该余额为可用余额；当订阅model=1时，该余额为总余额)

订阅订单更新

API Key 权限：读取

订阅账户下的订单更新。

主题订阅

orders.\$symbol

Subscribe request

```
{
  "op": "sub",
  "cid": "40sG903yz80oDFWr",
  "topic": "orders.htusdt",
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
symbol	string	true	NA	交易代码	btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>), 支持通配符"*".

Response

```
{
  "op": "sub",
  "cid": "40sG903yz80oDFWr",
  "err-code": 0,
  "ts": 1489474081631,
  "topic": "orders.htusdt"
}
```

Update example

```
{
  "op": "notify",
  "topic": "orders.htusdt",
  "ts": 1522856623232,
  "data": {
    "seq-id": 94984,
    "order-id": 2039498445,
    "symbol": "htusdt",
  }
}
```

```
"account-id": 100077,
"order-amount": "5000.0000000000000000",
"order-price": "1.6621000000000000",
"created-at": 1522858623622,
"order-type": "buy-limit",
"order-source": "api",
"order-state": "filled",
"role": "taker|maker",
"price": "1.6621000000000000",
"filled-amount": "5000.0000000000000000",
"unfilled-amount": "0.0000000000000000",
"filled-cash-amount": "8301.3572800000000000",
"filled-fees": "8.0000000000000000"
}
}
```

数据更新字段列表

字段	数据类型	描述
seq-id	integer	流水号(不连续)
order-id	integer	订单 id
symbol	string	交易对
account-id	string	账户 id
order-amount	string	订单数量
order-price	string	订单价格
created-at	int	订单创建时间 (UNIX epoch time in millisecond)
order-type	string	订单类型, 有效取值: buy-market, sell-market, buy-limit, sell-limit, buy-ioc, sell-ioc, buy-limit-maker, sell-limit-maker,buy-stop-limit,sell-stop-limit
order-source	string	订单来源, 有效取值: sys, web, api, app
order-state	string	订单状态, 有效取值: submitted, partial-filled, filled, canceled, partial-canceled
role	string	成交角色: taker or maker
price	string	成交价格
filled-amount	string	单次成交数量
unfilled-amount	string	单次未成交数量
filled-cash-amount	string	单次成交金额

字段	数据类型	描述
filled-fees	string	单次成交手续费（买入为币，卖出为钱）

订阅订单更新 (NEW)

API Key 权限：读取

相比现有用户订单更新推送主题 “orders.\$symbol”，新增主题 “orders.\$symbol.update” 拥有更低的数据延迟以及更准确的消息顺序。建议API用户订阅此新主题接收订单更新推送，以替代现有订阅主题 “orders.\$symbol”。（现有订阅主题 “orders.\$symbol” 仍将在 Websocket API服务中被保留直至另行通知。）

主题订阅

orders.\$symbol.update

Subscribe request

```
{
  "op": "sub",
  "cid": "40sG903yz80oDFWr",
  "topic": "orders.htusdt.update"
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
symbol	string	true	NA	btcusdt, ethbtc... (取值参考 <code>GET /v1/common/symbols</code>), 支持通配符"*".	

Response

```
{
  "op": "sub",
  "ts": 1489474081631,
  "topic": "orders.htusdt.update",
  "err-code": 0,
  "cid": "40sG903yz80oDFWr"
}
```

Update example

```
{
  "op": "notify",
  "ts": 1522856623232,
  "topic": "orders.htusdt.update",
  "data": {
    "unfilled-amount": "0.000000000000000000",
  }
}
```

```
    "filled-amount": "5000.0000000000000000",
    "price": "1.6621000000000000",
    "order-id": 2039498445,
    "symbol": "htusdt",
    "match-id": 94984,
    "filled-cash-amount": "8301.3572800000000000",
    "role": "taker|maker",
    "order-state": "filled",
    "client-order-id": "a0001",
    "order-type": "buy-limit"
  }
}
```

数据更新字段列表

Field	Data Type	Description
match-id	integer	最近撮合编号（当order-state = submitted, canceled, partial-canceled时，match-id 为消息序列号；当order-state = filled, partial-filled 时，match-id 为最近撮合编号。）
order-id	integer	订单编号
symbol	string	交易代码
order-state	string	订单状态, 有效取值: submitted, partial-filled, filled, canceled, partial-canceled
role	string	最近成交角色（当order-state = submitted, canceled, partial-canceled时，role 为缺省值taker；当order-state = filled, partial-filled 时，role 取值为taker 或maker。）
price	string	最新价（当order-state = submitted 时，price 为订单价格；当order-state = canceled, partial-canceled 时，price 为零；当order-state = filled, partial-filled 时，price 为最近成交价。）
filled-amount	string	最近成交数量
filled-cash-amount	string	最近成交数额
unfilled-amount	string	最近未成交数量（当order-state = submitted 时，unfilled-amount 为原始订单量；当order-state = canceled OR partial-canceled 时，unfilled-amount 为未成交数量；当order-state = filled 时，如果 order-type = buy-market，unfilled-amount 可能为一极小值；如果order-type <> buy-market 时，unfilled-amount 为零；当order-state = partial-filled AND role = taker 时，unfilled-amount 为未成交数量；当order-state = partial-filled AND role = maker 时，unfilled-amount 为未成交数量。）
client-order-id	string	用户自编订单号
order-type	string	订单类型，包括buy-market, sell-market, buy-limit, sell-limit, buy-ioc, sell-ioc, buy-limit-maker, sell-limit-maker,buy-stop-limit,sell-stop-limit

请求用户资产数据

API Key 权限：读取

查询当前用户的所有账户余额数据。

数据请求

accounts.list

Query request

```
{
  "op": "req",
  "cid": "40sG903yz80oDFWr",
  "topic": "accounts.list",
}
```

成功建立和 WebSocket API 的连接之后，向 Server发送如下格式的数据来查询账户数据：

参数	数据类型	描述
op	string	必填；操作名称，固定值为 req
cid	string	选填；Client 请求唯一 ID
topic	string	必填；固定值为accounts.list

返回

Successful

```
{
  "op": "req",
  "topic": "accounts.list",
  "cid": "40sG903yz80oDFWr",
  "err-code": 0,
  "ts": 1489474082831,
  "data": [
    {
      "id": 419013,
      "type": "spot",
      "state": "working",
      "list": [
        {
          "currency": "usdt",
          "type": "trade",
          "balance": "500009195917.4362872650"
        },
        {
          "currency": "usdt",
          "type": "frozen",
          "balance": "9786.6783000000"
        }
      ]
    }
  ],
}
```

```
{
  "id": 35535,
  "type": "point",
  "state": "working",
  "list": [
    {
      "currency": "eth",
      "type": "trade",
      "balance": "499999894616.1302471000"
    },
    {
      "currency": "eth",
      "type": "frozen",
      "balance": "9786.6783000000"
    }
  ]
}
```

Failed

```
{
  "op": "req",
  "topic": "foo.bar",
  "cid": "40sG903yz80oDFWr",
  "err-code": 12001, //Response codes,0 represent success,others value is error,the list of Response codes
  "err-msg": "detail of error message",
  "ts": 1489474081631
}
```

字段	数据类型	描述
{ id	long	账户ID
type	string	账户类型
state	string	账户状态
list	string	账户列表
{currency	string	子账户币种
type	string	子账户类型
balance }}	string	子账户余额

请求当前及历史订单

API Key 权限：读取

根据设定条件查询当前委托、历史委托。

数据请求

order.list

Query request

```
{
  "op": "req",
  "topic": "orders.list",
  "cid": "40sG903yz80oDFWr",
  "symbol": "htusdt",
  "states": "submitted,partial-filled"
}
```

参数

参数	数据类型	是否必需	缺省值	描述	取值范围
account-id	int	true	NA	账户 id	NA
symbol	string	true	NA	交易对	btcusdt, ethbtc... (取值参考GET /v1/common/symbols)
types	string	false	NA	查询的订单类型组合，使用','分割	buy-market, sell-market, buy-limit, sell-limit, buy-ioc, sell-ioc , buy-limit-maker, sell-limit-maker, buy-stop-limit , sell-stop-limit
states	string	true	NA	查询的订单状态组合，使用','分割	submitted, partial-filled, partial-canceled, filled, canceled , created
start-date	string	false	-61d	查询开始日期, 日期格式yyyy-mm-dd	NA
end-date	string	false	today	查询结束日期, 日期格式yyyy-mm-dd	NA
from	string	false	NA	查询起始 ID	NA
direct	string	false	next	查询方向	next, prev
size	string	false	100	查询记录大小	[1, 100]

数据更新字段列表

Successful

```
{
  "op": "req",
  "topic": "orders.list",
  "cid": "40sG903yz80oDFWr",
  "err-code": 0,
  "ts": 1522856623232,
}
```

```
"data": [
  {
    "id": 2039498445,
    "symbol": "htusdt",
    "account-id": 100077,
    "amount": "5000.000000000000000000",
    "price": "1.662100000000000000",
    "created-at": 1522858623622,
    "type": "buy-limit",
    "filled-amount": "5000.000000000000000000",
    "filled-cash-amount": "8301.357280000000000000",
    "filled-fees": "8.000000000000000000",
    "finished-at": 1522858624796,
    "source": "api",
    "state": "filled",
    "canceled-at": 0
  }
]
```

字段	数据类型	描述
id	long	订单ID
symbol	string	交易对
account-id	long	账户ID
amount	string	订单数量
price	string	订单价格
created-at	long	订单创建时间
type	string	订单类型，请参考订单类型说明
filled-amount	string	已成交数量
filled-cash-amount	string	已成交总金额
filled-fees	string	已成交手续费
finished-at	string	最后成交时间
source	string	订单来源，请参考订单来源说明
state	string	订单状态，请参考订单状态说明
cancel-at	long	撤单时间
stop-price	string	止盈止损订单触发价格
operator	string	止盈止损订单触发价运算符

以订单编号请求订单

API Key 权限：读取

以订单编号请求订单数据

数据请求

orders.detail

Query request

```
{
  "op": "req",
  "topic": "orders.detail",
  "order-id": "2039498445",
  "cid": "40sG903yz80oDFWr"
}
```

参数

参数	是否必需	数据类型	描述	缺省值	取值范围
op	true	string	操作名称，固定值为 req		
cid	true	string	Client 请求唯一 ID		
topic	false	string	固定值为 orders.detail		
order-id	true	string	订单ID		

返回

Successful

```
{
  "op": "req",
  "topic": "orders.detail",
  "cid": "40sG903yz80oDFWr",
  "err-code": 0,
  "ts": 1522856623232,
  "data": {
    "id": 2039498445,
    "symbol": "htusdt",
    "account-id": 100077,
    "amount": "5000.000000000000000000",
    "price": "1.662100000000000000",
    "created-at": 1522858623622,
    "type": "buy-limit",
    "filled-amount": "5000.0000000000000000",
    "filled-cash-amount": "8301.3572800000000000",
    "filled-fees": "8.0000000000000000",
    "finished-at": 1522858624796,
  }
}
```

```
{
  "source": "api",
  "state": "filled",
  "canceled-at": 0
}
```

字段	数据类型	描述
id	long	订单ID
symbol	string	交易对
account-id	long	账户ID
amount	string	订单数量
price	string	订单价格
created-at	long	订单创建时间
type	string	订单类型，请参考订单类型说明
filled-amount	string	已成交数量
filled-cash-amount	string	已成交总金额
filled-fees	string	已成交手续费
finished-at	string	最后成交时间
source	string	订单来源，请参考订单来源说明
state	string	订单状态，请参考订单状态说明
cancel-at	long	撤单时间
stop-price	string	止盈止损订单触发价格
operator	string	止盈止损订单触发价运算符

Websocket资产及订单（v2）

简介

接入URL

Websocket资产及订单（v2）

wss://api-cloud.huobi.co.kr/ws/v2

数据压缩

无

心跳消息

当用户的Websocket客户端连接到火币WebSocket服务器后，服务器会定期（当前设为20秒）向其发送Ping消息并包含一整数数值如下：

```
{
  "action": "ping",
  "data": {
    "ts": 1575537778295
  }
}
```

当用户的Websocket客户端接收到此心跳信息后，应返回Pong消息并包含同一整数数值：

```
{
  "action": "ping",
  "data": {
    "ts": 1575537778295 // 使用Ping消息中的ts值
  }
}
```

ACTION的有效取值

有效取值	取值说明
sub	订阅数据
req	数据请求
ping、pong	心跳数据
push	推送数据，服务端发送至客户端数据类型

鉴权

鉴权请求格式如下：

```
{
  "action": "req",
  "ch": "auth",
  "params": {
    "authType": "api",
    "accessKey": "sffd-ddfd-dfdsaf-dfdsafsd",
    "signatureMethod": "HmacSHA256",
    "signatureVersion": "2.1",
    "timestamp": "2019-09-01T18:16:16",
    "signature": "safsfdsjfljljdfsjsjfsdfhsdkjfhklhsd1kfjhlksdfh"
  }
}
```

鉴权成功后返回数据格式如下：

```
{
  "action": "req",
  "code": 200,
  "ch": "auth",
  "data": {}
}
```

参数说明

字段	是否必需	数据类型	描述
action	true	string	Websocket数据操作类型，鉴权固定值为req
ch	true	string	请求主题，鉴权固定值为auth
authType	true	string	鉴权类型，鉴权固定值为api
accessKey	true	string	您申请的API Key中的AccessKey
signatureMethod	true	string	签名方法，用户计算签名寄语哈希的协议，固定值为HmacSHA256
signatureVersion	true	string	签名协议版本，固定值为2.1
timestamp	true	string	时间戳，您发出请求的时间（UTC时间）在查询请求中包含此值有助于防止第三方截取您的请求。如：2017-05-11T16:22:06。再次强调是 (UTC 时区)
signature	true	string	签名, 计算得出的值，用于确保签名有效和未被篡改

签名步骤

v2.1版本签名与v2.0版本签名步骤相似，具体区别如下：

1. 生成参与签名的字符串时，请求方法固定使用GET，请求地址固定为/ws/v2
2. 生成参与签名的固定参数名替换为：accessKey，signatureMethod，signatureVersion，timestamp
3. signatureVersion版本升级为2.1

v2版本签名步骤,您可以点击 [这里](#) 获取。

签名前最后生成的字符串如下：

```
GET\n
api-cloud.huobi.co.kr\n
/ws/v2\n
accessKey=0664b695-rfhfg2mk13-abbf6c5d-49810&signatureMethod=HmacSHA256&signatureVersion=2.1&timestamp=2019-12
```

订阅主题

成功建立与Websocket服务器的连接后，Websocket客户端发送如下请求以订阅特定主题：

```
{
  "action": "sub",
```

```
    "ch": "accounts.update"
  }
```

订阅成功Websocket客户端会接收到如下消息：

```
{
  "action": "sub",
  "code": 200,
  "ch": "accounts.update#0",
  "data": {}
}
```

请求数据

成功建立Websocket服务器的连接后，Websocket客户端发送如下请求用以获取一次性数据：

```
{
  "action": "req",
  "ch": "topic",
}
```

请求成功后Websocket客户端会收到如下消息：

```
{
  "action": "req",
  "ch": "topic",
  "code": 200,
  "data": {} // 请求数据体
}
```

订阅清算后成交明细

API Key 权限：读取

清算后成交明细包含了交易手续费以及交易手续费抵扣等信息，仅当用户订单成交时推送。

订阅主题

```
trade.clearing#${symbol}
```

订阅参数

参数	数据类型	描述
symbol	string	交易代码（支持通配符 * ）

Subscribe request

```
{
  "action": "sub",
  "ch": "trade.clearing#btcusdt"
}
```

Response

```
{
  "action": "sub",
  "code": 200,
  "ch": "trade.clearing#btcusdt",
  "data": {}
}
```

Update example

```
{
  "ch": "trade.clearing#btcusdt",
  "data": {
    "symbol": "btcusdt",
    "orderId": 99998888,
    "tradePrice": "9999.99",
    "tradeVolume": "0.96",
    "orderSide": "buy",
    "aggressor": true,
    "tradeId": 919219323232,
    "tradeTime": 998787897878,
    "transactFee": "19.88",
    " feeDeduct ": "0",
    " feeDeductType": ""
  }
}
```

数据更新字段列表

字段	数据类型	描述
symbol	string	交易代码
orderId	long	订单ID
tradePrice	string	成交价
tradeVolume	string	成交量
orderSide	string	订单方向，有效值：buy, sell
orderType	string	订单类型，有效值：buy-market, sell-market, buy-limit, sell-limit, buy-ioc, sell-ioc, buy-limit-maker, sell-limit-maker, buy-stop-limit, sell-stop-limit

字段	数据类型	描述
aggressor	bool	是否交易主动方，有效值：true, false
tradeId	long	交易ID
tradeTime	long	成交时间，unix time in millisecond
transactFee	string	交易手续费
feeDeduct	string	交易手续费抵扣
feeDeductType	string	交易手续费抵扣类型，有效值：ht，point

订阅账户变更

API Key 权限：读取

订阅账户下的订单更新。

订阅主题

accounts.update#\${mode}

用户可选择以下任一账户变更推送的触发方式

- 1、仅在账户余额发生变动时推送；
- 2、在账户余额发生变动或可用余额发生变动时均推送，且分别推送。

订阅参数

参数	数据类型	描述
mode	integer	推送方式，有效值：0, 1，默认值：0

订阅示例

- 1、不填mode：
accounts.update
仅当账户余额变动时推送；
- 2、填写mode=0：
accounts.update#0
仅当账户余额变动时推送；
- 3、填写mode=1：
accounts.update#1
在账户余额发生变动或可用余额发生变动时均推送且分别推送。

Subscribe request

```
{
  "action": "sub",
```

```
"ch": "accounts.update"
}
```

Response

```
{
  "action": "sub",
  "code": 200,
  "ch": "accounts.update#0",
  "data": {}
}
```

Update example

```
accounts.update#0:
{
  "action": "push",
  "ch": "accounts.update#0",
  "data": {
    "currency": "btc",
    "accountId": 123456,
    "balance": "23.111",
    "changeType": "transfer",
    "accountType": "trade",
    "changeTime": 1568601800000
  }
}

accounts.update#1:
{
  "action": "push",
  "ch": "accounts.update#1",
  "data": {
    "currency": "btc",
    "accountId": 33385,
    "available": "2028.699426619837209087",
    "changeType": "order.match",
    "accountType": "trade",
    "changeTime": 1574393385167
  }
}

{
  "action": "push",
  "ch": "accounts.update#1",
  "data": {
    "currency": "btc",
    "accountId": 33385,
    "balance": "2065.100267619837209301",
    "changeType": "order.match",
    "accountType": "trade",
    "changeTime": 1574393385122
  }
}
```

```
}  
}
```

数据更新字段列表

字段	数据类型	描述
currency	string	币种
accountId	long	账户ID
balance	string	账户余额（仅当账户余额发生变动时推送）
available	string	可用余额（仅当可用余额发生变动时推送）
changeType	string	余额变动类型，有效值：order-place(订单创建)，order-match(订单成交)，order-refund(订单成交退款)，order-cancel(订单撤销)，order-fee-refund(点卡抵扣交易手续费)，other(其他资产变化)
accountType	string	账户类型，有效值：trade, frozen, loan, interest
changeTime	long	余额变动时间，unix time in millisecond