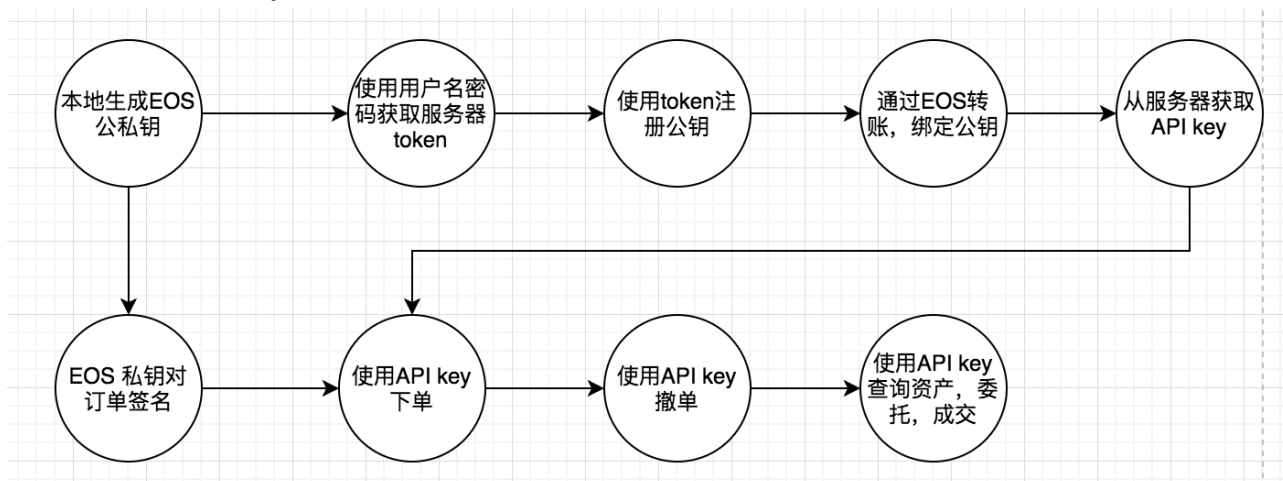


WhaleEx接口文档

安全认证	2
合法请求结构	4
签名运算	4
API接口	5
请求订单ID	6
下单	8
查询历史委托	10
查询当前委托订单	12
撤单	15
批量撤单	15
撤销当前所有委托	16
查询订单详情	17
查询成交	18
查询某个订单的成交明细	22
查询指定账户的余额	24
查询指定账户的余额	27
查询行情	28
获取订单簿	29
查询挖矿情况	31
行情推送Websocket	33

安全认证

WhaleEx的交易API是通过api key来访问。用户可以通过登录token来申请api key，流程如下图。api key不会过期，除非调用删除api key的接口。请妥善保管api key。Token需要使用用户名和密码获取，token存在时效性，所以不建议使用token做api自动化交易。



用户的关键请求，需要使用本地的私钥来签名，WhaleEx使用的是EOS签名方法。

重要提示：API Key与账号安全紧密相关，无论何时都请勿向其它人透露。
获取API key的步骤：

第一步，获取用户token

```
POST https://www.whaleex.com/UAA/oauth/token

// 使用form-data格式
username:<用户手机号码>
password:
4d4f26369171994f3a46776ee2d88494fb9955800a5bb6261c016c4bb9f30b56 (用户密码使用sha256做编码, 保证服务器无法知道用户密码)
grant_type:password
client_id:client
client_secret:secret
clientType:api
source:api
countryCode:CN
```

服务器返回JSON格式的应答。在应答中，字段 access_token 为用户的 token。

第二步，使用token来注册用户本地的eos公钥。

用户可以使用eos的钱包工具，例如scatter，或者使用eosjs-ecc的sdk来生成eos公私钥。eosjs-ecc是一个开源的基于nodejs项目，网址<https://github.com/EOSIO/eosjs-ecc>，它提供eos生产私钥，公钥，签名，验签功能。

生成秘钥对代码示例：

```
const ecc = require('eosjs-ecc')
ecc.randomKey().then(privateKey => {
  console.log('Private Key:\t', privateKey) // wif
  console.log('Public Key:\t',
ecc.privateToPublic(privateKey)) // EOSkey...
})
```

注册公钥请求：

```
POST https://www.whaleex.com/BUSINESS/api/account/pk4api?
pk=<public key>
需要带请求头Authorization: bearer <access_token>
```

返回returnCode为0时为成功。

第三步，用户需要使用eos钱包，通过向合约小额打币的方式，完成公钥在合约的绑定工作。交易所合约账户为whaleextrust, 打币为任意额度，备注使用
bind:<public key>:WhaleEx

查询公钥是否绑定成功：

```
GET https://www.whaleex.com/BUSINESS/api/auth/pk/status?
pk=<public key>
需要带请求头Authorization: bearer <access_token>
```

第四步，获取API Key

```
POST https://www.whaleex.com/BUSINESS/api/user/apiKey
需要带请求头Authorization: bearer <access_token>
```

response: 返回中result中的值为API key。

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": <APIKey>
}
```

合法请求结构

基于安全考虑，除行情API 外的 API 请求都必须进行签名运算。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.whaleex.com。
- API 访问密钥（API Key） 您申请的 API Key。
- 时间戳（timestamp） 您发出请求的时间戳，使用的是unix时间戳。
- 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。请一定注意：
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，下单接口除外，每个方法自带的参数不进行签名认证，即POST请求中需要进行签名运算的只有API Key，pk，timestamp三个参数，其它参数放在body中。下单接口使用专门的签名方式，具体方法查看下单接口说明。
- 签名 签名计算得出的值，用于确保签名有效和未被篡改。

例：

```
https://api.whaleex.com/BUSINESS/api/v1/order?  
APIKey=<APIKey>  
&timestamp=1542075993378  
&Signature=calculated value
```

签名运算

API 请求在通过 Internet 发送的过程中极有可能被篡改。为了确保请求未被更改，我们会要求用户在每个请求中带上签名（行情 API 除外），来校验参数或参数值在传输途中是否发生了更改。

计算签名所需的步骤：

1. 规范要计算签名的请求 因为使用 ECC 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。如果使用 Node，可使用eosjs-ecc。
2. 请求方法（GET 或 POST），后面添加换行符\n。
3. 添加小写的访问地址，后面添加换行符\n。
4. 访问方法的路径，删除前面大写的/BUSINESS，后面添加换行符\n。

按照ASCII码的顺序对参数名进行排序(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'，= 被编码成 '%3D'，如果使用的是 JS，可以用方法 `encodeURIComponent`)。最终组成的签名字符串如下：

```
GET\napi.whaleex.com\n/api/v1/order\nAPIKey=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx&pk=xxxxxxxxxxxxxxxxxxx&timestamp=1542075993378&orderId=1234567890
```

将以上字符串和您签名用的 eos 私钥，传入签名方法，可以得到签名，以 `eosjs-ecc` 为例：

```
const ecc = require('eosjs-ecc')\nvar signature = ecc.sign(data, privateKey);
```

将上述值参数 `Signature` 的取值，添加到 API 请求中。
如果是下单或撤单请求，还需要将 `orderId` 添加到 API 参数中。
最后需要将用户的 `publicKey` 添加到 API 参数中
最终发送到服务器的 API 请求应该为

```
APIKey=e2xxxxxx-99xxxxxx-84xxxxxx-7xxxx&timestamp=1542075993378\n&orderId=1234567890&Signature=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx&pk=yyy\nYYYYYYYYYYYYYYYYYYYYYYYYYYYY
```

API 接口

接口	说明
GET /BUSINESS/api/v1/order/globalIds	请求全局订单 ID，默认取 100 个
POST /BUSINESS/api/v1/order/orders/place	创建并执行订单，下单前需要先从服务器获取 ID。
POST /BUSINESS/api/v1/order/orders/{order-id}/submitcancel	撤销一个订单

POST /BUSINESS/api/v1/order/orders/batchcancel	批量撤销指定订单
POST /BUSINESS/api/v1/order/orders/batchCancelOpenOrders	撤销所有委托订单
GET /BUSINESS/api/v1/order/orders/{order-id}	查询一个订单详情
GET /BUSINESS/api/v1/order/orders	查询历史委托
GET /BUSINESS/api/v1/order/openOrders	查询当前委托订单
GET /BUSINESS/api/v1/order/matchresults	查询成交
GET /BUSINESS/api/v1/order/orders/{order-id}/matchresults	查询某个订单的成交明细
GET /BUSINESS/api/v1/assets	查询指定账户的余额
GET /BUSINESS/api/v1/asset/{currency}	获取制定币种资产
GET /BUSINESS/api/public/v1/ticker/{symbol}	获取行情
GET /BUSINESS/api/public/v1/orderBook/{symbol}	获取订单簿
GET /BUSINESS/api/whale/public/exchange	查询挖矿的状态

请求订单ID

GET /BUSINESS/api/v1/order/globalIds

参数：

名称	是否必须	类型	描述	取值范围
----	------	----	----	------

size	FALSE	int	返回指定数量的orderId，下单时，需要从中选取一个orderId。每个orderId只能使用一次。默认size=100	1-100
remark	FALSE	long	使用桶存储用户的预分配id。每个用户有10个桶。桶生成orderId时，会覆盖原桶里内容。使用remark可以固定住从某一个空闲桶中获取orderId。强烈建议后续的api调用，使用上一次返回的remark。	long

响应

参数	类型	描述
remark	String	orderId的桶虚拟ID
list	Array	orderId的列表

示例：

GET /BUSINESS/api/v1/order/globalIds?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F×tamp=1542202065272&Signature=SIG_K1_KZ69h
pTe9NA6ps7BYe2xhGrbjoHfM4GH9gV9sziNU4CknHWQWdeinspuiJyRass1HSYbKxV
GEBmVaSDrfGKT9UVvForz1a

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
    "remark": "89111215825631040",
    "list": [
      "89376914527631171",
      "89376914527631181",
      "89376914527631191",
      "89376914527631201",
      "89376914527631211",
      "89376914527631221",
      "89376914527631231",
      "89376914531825481",
      "89376914531825491",
      "89376914531825501"
    ]
  }
}
```

```
}  
}
```

下单

POST /BUSINESS/api/v1/order/orders/place

参数：

名称	是否必须	类型	描述	取值范围
orderId	TRUE	long	使用globalId获取的订单id。放在URL的参数中。	long
amount	TRUE	String	限价单表示下单数量，市价买单时表示买多少钱，市价卖单时表示卖多少币	
price	FALSE	String	下单价格，市价单不传该参数	
symbol	TRUE	String	交易对	IQEOS, DICEEOS...
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

响应

retureCode为0时，代表下单成功。result中为下单的ID。

下单接口的签名方式与其它不同，订单需要上链，合约要对订单签名做校验。将以下内容，按照其类型，拼接成字节流，然后进行签名，以IQ_EOS交易对为例

字段按照顺序	类型	买单限价（以IQEOS为例）	卖单限价（以IQEOS为例）	买单市价（以IQEOS为例）	卖单市价（以IQEOS为例）
用户EOS账户	string				
交易所EOS账户	int64	whaleexchang	whaleexchang	whaleexchang	whaleexchang
订单ID	int64				
当前timestamp	int32, 秒级				

支付币 合约名称	string	eosio.token	everipediaiq	eosio.token	everipediaiq
支付币简称	string	EOS	IQ	EOS	IQ
支付数量	int64, 按照链上 精度位 移, 取整 数	amount*price 后小数点向右 位移4位, 向上 取整	amount小数点 向右位移3位	amount小数点 向右位移4位	amount小数点 向右位移3位
获得币 合约名称	string	everipediaiq	eosio.token	everipediaiq	eosio.token
获得币简称	string	IQ	EOS	IQ	EOS
获得数量	int64, 按照链上 精度位 移, 取整 数	amount小数点 向右位移3位	amount*price 后小数点向右 位移4位, 向 下取整	0	0
maker 手续费	int16, 万 分之10	10	10	10	10
taker 手续费	int16, 万 分之10	10	10	10	10

示例:

```
POST /BUSINESS/api/v1/order/orders/place?
APIKey=guMFM9BITeqFwNRv@1-35&orderId=89385080403558920&pk=EOS8315E
f6A3mVrsrmRCdJxuu2Xm32VZFEFTCac16hfJUa8rgiU4F&timestamp=1542267550
741&Signature=SIG_K1_JwsxYjnzVK6YGBvd9Ck1oCzP4Kxq3sv9Wn4ALCB7vvMvZ
FVnRdloipDR9FgjJADc2KHTVe4kJmPLcnCDseGwi3RzBeDSdR
```

```
{
  "amount": "100",
  "price": "0.01",
  "symbol": "IQEBTC",
  "type": "buy-limit"
}
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
```

```
"result":89385080403558920
}
```

查询历史委托

GET /BUSINESS/api/v1/order/orders

参数：

名称	是否必须	类型	描述	取值范围
symbol	FALSE	string	交易对	IQEOS...缺省将返回所有符合条件尚未成交订单
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
start-date	FALSE	string	查询开始日期，日期格式 yyyy-mm-dd	
end-date	FALSE	string	查询结束日期，日期格式 yyyy-mm-dd	
page	FALSE	int	分页	从0开始，默认0
size	FALSE	int	分页大小	默认10

返回

retureCode为0时，代表返回成功。

参数	类型	描述
totalElements	long	数据库中符合条件总个数
totalPages	int	总页数
price	string	下单价格，0为市价单
origQty	string	原始下单量，为quote币的数量，buy-market订单不做参考
execValue	string	成交金额
execAvgPrice	string	成交均价
execQty	string	成交数量，为quote币的数量

status	int	67接受, 74拒单, 88已撤, 83成交
side	int	66买, 83卖
type	int	76限价, 77市价
createTime	long	创建时间

示例:

```
GET /BUSINESS/api/v1/order/orders?
APIKey=guMFM9BITeqFwNRv@1-35&end-
date=2018-11-09&page=0&pk=EOS8315Ef6A3mVrsrmRCdJxuu2Xm32VZFEFTCac1
6hfJUa8rgiU4F&size=10&start-
date=2018-11-07&symbol=IQEBTC&timestamp=1542302020294&Signature=SI
G_K1_K7NTni5aaLR61E2BGqqtAqFWwDSronu4mj9asEzj62im4aTAtQn9nFhKamZ2s
RsRx9nvFBqFoBH6yyo6X8JrzHmMLbxDmq
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
    "content": [
      {
        "orderId": "86957940577406170",
        "symbolId": 2,
        "status": "83",
        "price": "1.000000",
        "origQty": "1.000",
        "execValue": "0.08565000",
        "execAvgPrice": "0.085650",
        "execQty": "1.000",
        "createTime": 1541690589207,
        "type": "76",
        "side": 66
      },
      {
        "orderId": "86957940577406150",
        "symbolId": 2,
        "status": "83",
        "price": "1.000000",
        "origQty": "10.000",
        "execValue": "0.85650000",
        "execAvgPrice": "0.085650",
```

```
        "execQty": "10.000",
        "createTime": 1541689333074,
        "type": "76",
        "side": 66
    },
    ],
    "pageable": {
        "sort": {
            "sorted": true,
            "unsorted": false
        },
        "pageSize": 10,
        "pageNumber": 0,
        "offset": 0,
        "paged": true,
        "unpaged": false
    },
    "totalElements": 2,
    "totalPages": 1,
    "last": true,
    "first": true,
    "sort": {
        "sorted": true,
        "unsorted": false
    },
    "numberOfElements": 2,
    "size": 10,
    "number": 0
}
}
```

查询当前委托订单

GET /BUSINESS/api/v1/order/openOrders

参数:

名称	是否必须	类型	描述	取值范围
symbol	FALSE	string	交易对	IQEOS...缺省将返回所有符合条件尚未成交订单
side	FALSE	string	买卖方向	buy, sell缺省将返回所有符合条件尚未成交订单
page	FALSE	int	分页	从0开始，默认0
size	FALSE	int	分页大小	默认10

返回

retureCode为0时，代表返回成功。

参数	类型	描述
totalElements	long	数据库中符合条件总个数
totalPages	int	总页数
price	string	下单价格，0为市价单
origQty	string	原始下单量，为quote币的数量，buy-market订单不做参考
execValue	string	成交金额
execAvgPrice	string	成交均价
execQty	string	成交数量，为quote币的数量
status	int	67接受，74拒单，88已撤，83成交
side	int	66买，83卖
type	int	76限价，77市价
createTime	long	创建时间

示例：

```
GET /BUSINESS/api/v1/order/openOrders?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&side=buy&symbol=IQEBTC&timestamp=1542284387583
&Signature=SIG_K1_JvdqtuQ8oSSFWoW1W65HvJAQbDCAVxwDfzzk4dBP9z6aWTMT
4aSjWF2LA9EqXoloLywTzsnmW8uZvjYH1qHy4M2cot9qPZ
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
    "content": [
      {
        "orderId": "89119097853633179",
        "symbolId": 2,
        "status": 67,
        "price": "0.010000",
        "origQty": "100.000",
        "execValue": "0",
```

```

        "execAvgPrice": "0",
        "execQty": "0",
        "createTime": 1542204514261,
        "type": "76",
        "resignTime": 0,
        "side": 66,
        "takerFeeRate": 10,
        "makerFeeRate": 10
    },
    {
        "orderId": "89119097853633159",
        "symbolId": 2,
        "status": 67,
        "price": "0.010000",
        "origQty": "100.000",
        "execValue": "0",
        "execAvgPrice": "0",
        "execQty": "0",
        "createTime": 1542204468384,
        "type": "76",
        "resignTime": 0,
        "side": 66,
        "takerFeeRate": 10,
        "makerFeeRate": 10
    }
],
"pageable": {
    "sort": {
        "sorted": true,
        "unsorted": false
    },
    "pageSize": 10,
    "pageNumber": 0,
    "offset": 0,
    "paged": true,
    "unpaged": false
},
"totalElements": 2,
"totalPages": 1,
"last": true,
"first": true,
"sort": {
    "sorted": true,
    "unsorted": false
},
"numberOfElements": 2,
"size": 10,
"number": 0
}

```

撤单

POST /BUSINESS/api/v1/order/orders/{order-id}/submitcancel

参数

名称	是否必须	类型	描述	取值范围
orderId	TRUE	string	订单id, 放在path中	

返回

retureCode为0时, 代表撤单请求发送成功。撤单是否成功, 需要调用查询订单详情的接口。

示例:

```
POST /BUSINESS/api/v1/order/orders/{order-id}/submitcancel?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542202065272&Signature=SIG_K1_KZ69h
pTe9NA6ps7BYe2xhGrbjoHfm4GH9gV9sziNU4CknHWQWdeinspuiJyRass1HSYbKxV
GEBmVaSDrfGKT9UVvForz1a
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": 89385080403558920
}
```

批量撤单

POST /BUSINESS/api/v1/order/orders/batchcancel

参数:

名称	是否必须	类型	描述	取值范围
order-ids	TRUE	string[]	需要撤单的订单ID	

返回

retureCode为0时, 代表撤单请求发送成功。撤单是否成功, 需要调用查询订单详情的接口。

示例:

```
POST /BUSINESS/api/v1/order/orders/batchcancel?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542202065272&Signature=SIG_K1_KZ69h
```

```
pTe9NA6ps7BYe2xhGrbjoHfM4GH9gV9sziNU4CknHWQWdeinspuiJyRasslHSYbKxV
GEBmVaSDrfGKT9UVvForz1a
{
  "order-ids":[
    "1",
    "2",
    "3"
  ]
}
```

=====

```
{
  "returnCode":"0",
  "message":"success",
  "errorCode":null,
  "result":null
}
```

撤销当前所有委托

POST /BUSINESS/api/v1/order/orders/batchCancelOpenOrders

参数:

名称	是否必须	类型	描述	取值范围
symbol	FALSE	string	交易对	IQEOS...缺省将返回所有符合条件尚未成交订单
side	FALSE	string	买卖方向	buy, sell缺省将返回所有符合条件尚未成交订单

返回

retureCode为0时，代表撤单请求发送成功。撤单是否成功，需要调用查询订单详情的接口。

示例:

```
POST /BUSINESS/api/v1/order/orders/batchCancelOpenOrders?
APIKey=guMFM9BITEqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542202065272&Signature=SIG_K1_KZ69h
pTe9NA6ps7BYe2xhGrbjoHfM4GH9gV9sziNU4CknHWQWdeinspuiJyRasslHSYbKxV
GEBmVaSDrfGKT9UVvForz1a
```

=====

```
{
```



```
"returnCode": "0",
"message": "success",
"errorCode": null,
"result": null
}
```

查询订单详情

GET /BUSINESS/api/v1/order/orders/{order-id}

参数:

名称	是否必须	类型	描述	取值范围
orderId	TRUE	string	订单id, 放在path中	

返回

returnCode为0时, 代表返回成功。

参数	类型	描述
price	string	下单价格, 0为市价单
origQty	string	原始下单量, 为quote币的数量, buy-market订单不做参考
execValue	string	成交金额
execAvgPrice	string	成交均价
execQty	string	成交数量, 为quote币的数量
status	int	67接受, 74拒单, 88已撤, 83成交
side	int	66买, 83卖
type	int	76限价, 77市价
createTime	long	创建时间

示例:

```
GET /BUSINESS/api/v1/order/orders/89391252051212110?
APIKey=guMFM9BITeqFwNRv@1-35&pk=E0S8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542202065272&Signature=SIG_K1_KZ69h
pTe9NA6ps7BYe2xhGrbjoHfM4GH9gV9sziNU4CknHWQWdeinspuiJyRass1HSYbKxV
GEBmVaSDrfGKT9UVvForz1a
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
    "orderId": "89391252051212111",
    "symbolId": 2,
    "status": 83,
    "price": "0",
    "origQty": "100.000",
    "execValue": "99.99990000",
    "execAvgPrice": "0.554400",
    "execQty": "180.375",
    "createTime": 1542269774728,
    "type": "77",
    "resignTime": 0,
    "side": 66,
    "takerFeeRate": 10,
    "makerFeeRate": 10
  }
}
```

查询成交

GET /BUSINESS/api/v1/order/matchresults

参数：

名称	是否必须	类型	描述	取值范围
symbol	FALSE	string	交易对	IQEOS...缺省将返回所有符合条件尚未成交订单
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
start-date	FALSE	string	查询开始日期，日期格式yyyy-mm-dd	
end-date	FALSE	string	查询结束日期，日期格式yyyy-mm-dd	
page	FALSE	int	分页	从0开始，默认0
size	FALSE	int	分页大小	默认10

返回

retureCode为0时，代表返回成功。

参数	类型	描述
totalElements	long	数据库中符合条件总个数
totalPages	int	总页数
orderId	string	订单id
execId	string	成交id
price	string	成交价格
quantity	string	成交数量
volume	string	成交金额
feeQty	string	手续费
feeCurrency	string	手续费币种
side	int	66买， 83卖
type	int	77市价， 76限价
liquidityFlag	int	65为maker， 82为taker
timestamp	long	成交时间

示例：

```
GET /BUSINESS/api/v1/order/matchresults?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542285662956&Signature=SIG_K1_KcyWW
S7kZTYgXicPubRuAFjA2ZRL5q43b5tin4wLsy8cpVxtaxstj7WGfd7NFMGmuRVb6Fd
s5eUjTtqPCGWTTfZB99ZX2z
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
```

```
"content":[
  {
    "orderId":"89453946225356310",
    "execId":"59532541722625",
    "type":76,
    "quoteCurrencyName":null,
    "baseCurrencyName":null,
    "price":"0.032420",
    "quantity":"13.000",
    "volume":"0.42146000",
    "side":66,
    "status":null,
    "feeQty":"0.013",
    "feeCurrencyId":4,
    "feeCurrency":"IQ",
    "liquidityFlag":0,
    "timestamp":1542290160571
  },
  {
    "orderId":"89453946225356300",
    "execId":"59528515190785",
    "type":76,
    "quoteCurrencyName":null,
    "baseCurrencyName":null,
    "price":"0.032440",
    "quantity":"11.000",
    "volume":"0.35684000",
    "side":66,
    "status":null,
    "feeQty":"0.011",
    "feeCurrencyId":4,
    "feeCurrency":"IQ",
    "liquidityFlag":0,
    "timestamp":1542289051146
  },
  {
    "orderId":"89453946225356340",
    "execId":"59282091442177",
    "type":76,
    "quoteCurrencyName":null,
    "baseCurrencyName":null,
    "price":"0.544820",
    "quantity":"1000.000",
    "volume":"544.82000000",
    "side":66,
    "status":null,
    "feeQty":"1.000",
    "feeCurrencyId":4,
    "feeCurrency":"IQ",
    "liquidityFlag":0,
    "timestamp":1542285352092
  },
]
```

```
{
  "orderId": "89140894665859720",
  "execId": "56727928078337",
  "type": 76,
  "quoteCurrencyName": null,
  "baseCurrencyName": null,
  "price": "1.000000",
  "quantity": "1.000",
  "volume": "1.00000000",
  "side": 66,
  "status": null,
  "feeQty": "0.001",
  "feeCurrencyId": 4,
  "feeCurrency": "IQ",
  "liquidityFlag": 0,
  "timestamp": 1542209661032
},
{
  "orderId": "86957940577406170",
  "execId": "43778064809985",
  "type": 76,
  "quoteCurrencyName": null,
  "baseCurrencyName": null,
  "price": "0.085650",
  "quantity": "1.000",
  "volume": "0.08565000",
  "side": 66,
  "status": null,
  "feeQty": "0.001",
  "feeCurrencyId": 4,
  "feeCurrency": "IQ",
  "liquidityFlag": 0,
  "timestamp": 1541690589216
},
{
  "orderId": "86957940577406150",
  "execId": "43725183025153",
  "type": 76,
  "quoteCurrencyName": null,
  "baseCurrencyName": null,
  "price": "0.085650",
  "quantity": "10.000",
  "volume": "0.85650000",
  "side": 66,
  "status": null,
  "feeQty": "0.010",
  "feeCurrencyId": 4,
  "feeCurrency": "IQ",
  "liquidityFlag": 0,
  "timestamp": 1541689333083
}
],
```

```
    "pageable":{
      "sort":{
        "unsorted":false,
        "sorted":true
      },
      "offset":0,
      "pageSize":10,
      "pageNumber":0,
      "unpaged":false,
      "paged":true
    },
    "totalPages":1,
    "last":true,
    "totalElements":6,
    "size":10,
    "number":0,
    "numberOfElements":6,
    "first":true,
    "sort":{
      "unsorted":false,
      "sorted":true
    }
  }
}
```

查询某个订单的成交明细

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

参数

名称	是否必须	类型	描述	取值范围
orderId	TRUE	string	订单id, 放在path中	
page	FALSE	int	分页	从0开始, 默认0
size	FALSE	int	分页大小	默认10

返回

retureCode为0时, 代表返回成功。

参数	类型	描述
totalElements	long	数据库中符合条件总个数
totalPages	int	总页数

orderId	string	订单id
execId	string	成交id
price	string	成交价格
quantity	string	成交数量
volume	string	成交金额
feeQty	string	手续费
feeCurrency	string	手续费币种
side	int	66买, 83卖
type	int	77市价, 76限价
liquidityFlag	int	65为maker, 82为taker
timestamp	long	成交时间

示例:

```
GET /BUSINESS/api/v1/order/orders/89453946225356340/matchresults?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542285662956&Signature=SIG_K1_KcyWW
S7kZTYgXicPubRuAFjA2ZRL5q43b5tin4wLsy8cpVxtaxstj7WGfd7NFMGmuRVb6Fd
s5eUjTtqPCGWTTfZB99ZX2z
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
    "content": [
      {
        "orderId": "89453946225356340",
        "execId": "59282091442177",
        "type": 76,
        "quoteCurrencyName": null,
        "baseCurrencyName": null,
        "price": "0.544820",
        "quantity": "1000.000",
```

```
        "volume": "544.82000000",
        "side": 66,
        "status": null,
        "feeQty": "1.000",
        "feeCurrencyId": 4,
        "feeCurrency": "IQ",
        "liquidityFlag": 65,
        "timestamp": 1542285352092
    }
},
"pageable": {
    "sort": {
        "sorted": true,
        "unsorted": false
    },
    "offset": 0,
    "pageSize": 10,
    "pageNumber": 0,
    "unpaged": false,
    "paged": true
},
"totalPages": 1,
"last": true,
"totalElements": 1,
"size": 10,
"number": 0,
"numberOfElements": 1,
"first": true,
"sort": {
    "sorted": true,
    "unsorted": false
}
}
```

查询指定账户的余额

GET /BUSINESS/api/v1/assets

参数：

名称	是否必须	类型	描述	取值范围
page	FALSE	int	分页	从0开始，默认0
size	FALSE	int	分页大小	默认20

返回

retureCode为0时，代表返回成功。

参数	类型	描述
----	----	----

totalElements	long	数据库中符合条件总个数
totalPages	int	总页数
currency	string	资产币名
totalAmount	string	总量
availableAmount	string	可用
frozenAmount	string	冻结
fixedAmount	string	持仓区部分
privatePlacement	string	私募数量

示例:

```
GET /BUSINESS/api/v1/assets?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542287222597&Signature=SIG_K1_JztyZ
zzKbXEBwHnygbhvha6DDNHHnFr44iYFufmF7yt7cd81voHYSdEeZyJwjSYJ3CZAx2x
k65HCWU8CyrYMzQDHQm8Ppn
```

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
    "wal": {
      "freeAmount": "0",
      "stakeAmount": "0",
      "unstakingAmount": "0"
    },
    "summary": {
      "amount": "169735.5773",
      "shortName": "EOS",
      "withdrawNeedVerify": false
    },
    "list": {
      "content": [
        {
          "currency": "EBTC",
          "currencyId": 7,

```

```

    "totalAmount": "9353.23795000",
    "availableAmount": "9352.45965000",
    "frozenAmount": "0.77830000",
    "fixedAmount": "0",
    "privatePlacement": "0"
  },
  {
    "currency": "IQ",
    "currencyId": 4,
    "totalAmount": "101391.183",
    "availableAmount": "101183.163",
    "frozenAmount": "11.000",
    "fixedAmount": "197.020",
    "privatePlacement": "0"
  },
  {
    "currency": "WAL",
    "currencyId": 5,
    "totalAmount": "100600.000",
    "availableAmount": "39.913",
    "frozenAmount": "0",
    "fixedAmount": "560.087",
    "privatePlacement": "100000.000"
  },
  {
    "currency": "EOS",
    "currencyId": 1,
    "totalAmount": "100000.0002",
    "availableAmount": "0.0002",
    "frozenAmount": "0",
    "fixedAmount": "0",
    "privatePlacement": "100000.0000"
  },
  {
    "currency": "ID",
    "currencyId": 2,
    "totalAmount": "0",
    "availableAmount": "0",
    "frozenAmount": "0",
    "fixedAmount": "0",
    "privatePlacement": "0"
  },
  {
    "currency": "BTC",
    "currencyId": 3,
    "totalAmount": "0",
    "availableAmount": "0",
    "frozenAmount": "0",
    "fixedAmount": "0",
    "privatePlacement": "0"
  },
  {

```

```

        "currency": "CANS",
        "currencyId": 6,
        "totalAmount": "0",
        "availableAmount": "0",
        "frozenAmount": "0",
        "fixedAmount": "0",
        "privatePlacement": "0"
    }
},
"pageable": {
    "sort": {
        "sorted": false,
        "unsorted": true
    },
    "offset": 0,
    "pageSize": 20,
    "pageNumber": 0,
    "paged": true,
    "unpaged": false
},
"totalElements": 7,
"totalPages": 1,
"last": true,
"size": 20,
"number": 0,
"first": true,
"numberOfElements": 7,
"sort": {
    "sorted": false,
    "unsorted": true
}
}
}
}

```

查询指定账户的余额

GET /BUSINESS/api/v1/asset/{currency}

参数

名称	是否必须	类型	描述	取值范围
currency	TRUE	币种简称		EOS, IQ...

返回

retureCode为0时，代表返回成功。

参数	类型	描述
currency	string	资产币名

total	string	总量
frozen	string	冻结
fixedAmount	string	持仓区部分

示例：

```
GET /BUSINESS/api/v1/asset/IQ?
APIKey=guMFM9BITeqFwNRv@1-35&pk=EOS8315Ef6A3mVrsrcRCdJxuu2Xm32VZFE
FTCac16hfJUa8rgiU4F&timestamp=1542287645373&Signature=SIG_K1_KkfjQ
6CxX7aqaFn495SrSq2cxFiAFHdnNzWFwdUnYqxHxbwGNAfAvQbcSQaNLd3QPUBJsdC
chdMa24mkZ6pi35WYkAnM7o
```

```
=====

{
  "returnCode": "0",
  "message": "success",
  "result": {
    "currencyId": 4,
    "quotable": false,
    "currency": "IQ",
    "total": "101194.163",
    "frozen": "11.000",
    "fixedAmount": "0"
  }
}
```

查询行情

GET /BUSINESS/api/public/v1/ticker/{symbol}

开放接口，不需要签名，参数中不需要包含文档开头中的安全参数。

参数：

名称	是否必须	类型	描述	取值范围
symbol	TRUE	string	交易对简称	IQEOS, WALEOS...
timestamp	TRUE	long	当前时间戳	

返回

参数	类型	描述
priceChange	string	相比开盘价格变化

priceChangePercent	string	相比开盘价格变化百分比
lastPrice	string	当前价格
quoteVolume	string	quote币的交易总额
baseVolume	string	base币的交易总额
open	string	开盘价
high	string	最高价
low	string	最低价

示例：

GET /BUSINESS/api/public/v1/ticker/IQEBTC?timestamp=1542287645373

=====

```
{
  "lastUpdateId": "59521804304385",
  "timestamp": 1542288160695,
  "symbolId": 2,
  "baseCurrencyId": 0,
  "quoteCurrencyId": 0,
  "priceChange": "-0.455700",
  "priceChangePercent": "-0.4557",
  "lastPrice": "0.544300",
  "lastQty": null,
  "quoteVolume": "32129935.23642966",
  "baseVolume": "58462842.616",
  "open": "1.000000",
  "high": "1.000000",
  "low": "0.500000"
}
```

获取订单簿

GET /BUSINESS/api/public/v1/orderBook/{symbol}

开放接口，不需要签名，参数中不需要包含文档开头中的安全参数。

参数：

名称	是否必须	类型	描述	取值范围
symbol	TRUE	string	交易对简称	IQEOS, WALEOS...
timestamp	TRUE	long	当前时间戳	

level	FALSE	int	档位	默认值10，可取值5， 10， 20
-------	-------	-----	----	--------------------

返回

参数	类型	描述
ask.price	string	卖档价格
ask.quantity	string	卖档数量
bid.price	string	买档价格
bid.quantity	string	买档数量

示例：

GET /BUSINESS/api/public/v1/orderBook/{symbol}?
timestamp=1542287645373

```
=====
{
  "lastUpdateId": "59528515190786",
  "timestamp": 1542289112531,
  "asks": [
    {
      "price": "0.032440",
      "quantity": "58712.534"
    },
    {
      "price": "0.541530",
      "quantity": "103729.521"
    },
    {
      "price": "0.548160",
      "quantity": "246427.814"
    },
    {
      "price": "0.554822",
      "quantity": "12.366"
    },
    {
      "price": "0.555087",
      "quantity": "34.075"
    },
    {
      "price": "0.560812",
      "quantity": "23.195"
    },
    {
```

```

        "price": "0.567838",
        "quantity": "6.576"
    },
    {
        "price": "0.568556",
        "quantity": "29.849"
    },
    {
        "price": "0.576531",
        "quantity": "20.094"
    },
    {
        "price": "0.581306",
        "quantity": "16.222"
    }
],
"bids": [
    {
        "price": "0.032420",
        "quantity": "13.000"
    }
]
}

```

查询挖矿情况

GET /BUSINESS/api/whale/public/exchange

开放接口，不需要签名，参数中不需要包含文档开头中的安全参数。

返回

参数	类型	描述
mineAmount	string	挖矿总量
currentMineAmount	string	当日挖矿量
currentRemainAmount	string	当日挖矿剩余
yesterdayMineAmount	string	昨日挖矿量
currentHourRemainAmount	string	当前小时挖矿剩余
exchangeHourLimitAmount	string	交易所每小时挖矿硬顶
accumulateMineAmount	string	累计挖矿量
totalTradingVolume	string	总交易量
totalTradingValue	string	总交易金额

symbolsRemain.{symbol}.dayLimit	long	交易对每天硬顶
symbolsRemain.{symbol}.hourLimit	long	交易对每小时硬顶
symbolsRemain.{symbol}.totalLimit	long	交易对总硬顶
symbolsRemain.{symbol}.hourRemainAmount	string	交易对当前小时剩余量
symbolsRemain.{symbol}.dayRemainAmount	string	交易对当天剩余量
symbolsRemain.{symbol}.totalRemainAmount	string	交易对总剩余量

示例:

GET /BUSINESS/api/whale/public/exchange

=====

```
{
  "returnCode": "0",
  "message": "success",
  "errorCode": null,
  "result": {
    "lockAmount": "106931.457",
    "mineAmount": "8783330.860",
    "eosExchangeRate": null,
    "currentMineAmount": "5928811.737",
    "currentRemainAmount": "6071188.263",
    "yesterdayMineAmount": "0.000",
    "currentHourRemainAmount": "0",
    "exchangeHourLimitAmount": "1000000",
    "accumulateMineAmount": "8783330.860",
    "totalTradingVolume": "3395881",
    "totalTradingValue": "8829.292",
    "yesterdayRepoFundAmount": "0.0000",
    "totalRepoFundAmount": "5036.8163",
    "symbolsRemain": {
      "CANEOS": {
        "dayLimit": 12000000,
        "hourLimit": 1000000,
        "totalLimit": 3000000000,
        "hourRemainAmount": "0",
        "dayRemainAmount": "6071188.263",
        "totalRemainAmount": "135216669.14"
      },
      "TPTEOS": {
        "dayLimit": 12000000,
        "hourLimit": 1000000,
        "totalLimit": 3000000000,
        "hourRemainAmount": "0",
        "dayRemainAmount": "6071188.263",
```



```

        "totalRemainAmount": "135216669.14"
    },
    "MEETONEEOS": {
        "dayLimit": 12000000,
        "hourLimit": 1000000,
        "totalLimit": 3000000000,
        "hourRemainAmount": "0",
        "dayRemainAmount": "6071188.263",
        "totalRemainAmount": "135216669.14"
    }
},
"yesterdayRepoFundDetails": {
    "CANS" : "83048276",
    "EBTC" : "833023915415"
}
}
}
}

```

行情推送Websocket

WhaleEx提供最新成交和订单簿的推送。我们的使用了stomp协议。
交易所的websocket地址是：<wss://market-gw.whaleex.com/ws/websocket>
下面是nodejs的示例：

```

var Stomp = require('stompjs');
var client=Stomp.overWS("wss://market-gw.whaleex.com/ws/websocket");
client.heartbeat.outgoing = 20000;
client.heartbeat.incoming = 20000;
client.connect('geust', 'guest', function(frame) {
    console.log(frame);
    console.log("connected to Stomp");
    client.subscribe("/IQEOS@depth5", function(message) {
        console.log(message);
    });
    client.subscribe("/IQEOS@trade", function(message) {
        console.log(message);
    });
}, function(error){
    console.log(error);
});

```

topic	内容	消息返回格式
/DICEEOS@depth5	订阅DICEEOS交易对的订单簿，depth后面的数字为订单簿档位，支持5，10，20	如下

/DICEEOS@trade	订阅DICEEOS交易对的最新成交推送	如下
----------------	---------------------	----

订单簿推送，type为“B”，bids数组为买盘，asks数组为卖盘，数组中每一个item为“price:amount”. 格式如下：

```
{
  "type": "B",
  "timestamp": 1542085721967,
  "symbol": "DICEEOS",
  "firstUpdateId": "12435272630272",
  "lastUpdateId": "12435272630272",
  "bids": [
    "0.000092:2511.8696",
    "0.000091:10990.1099",
    "0.000090:78557.1426",
    "0.000089:2103.4102",
    "0.000082:5000.0000",
    "0.000074:3940.5405",
    "0.000072:1417.0000",
    "0.000071:1914.0845",
    "0.000070:170975.7143",
    "0.000069:339296.7656"
  ],
  "asks": [
    "0.000101:9000.0000",
    "0.000102:8383.6805",
    "0.000105:10000.0000",
    "0.000106:6511.2195",
    "0.000107:1841.8205",
    "0.000108:4885.7500",
    "0.000110:15383.5472",
    "0.000112:7475.9057",
    "0.000114:106647.3000",
    "0.000115:101500.0000",
    "0.000118:14945.8062",
    "0.000119:20164.5085"
  ]
}
```

成交推送，type为“T”，bidAsk字段，“A”为taker卖方，“B”为taker买方。格式如下：

```
{
  "type": "T",
  "timestamp": 1542085974687,
  "tradeId": "12435809435650",
  "symbol": "DICEEOS",
  "price": "0.002077",
  "quantity": "139.9594",
  "bidAsk": "A"
}
```

