

# DFuture 开平仓 SDK 简介

本文基于JS 实现

## 一. 与Web3 接口的不同点

1. 增加了专用的开平仓,和查询线上价格的RPC接口, 不使用web3

Heco

```
"https://openoracle_prod_heco.dfuture.com/dev/web/sendOpenPosition"
"https://openoracle_prod_heco.dfuture.com/dev/web/sendClosePosition"
"https://openoracle_prod_heco.dfuture.com/dev/web/getLatesetPrice?symbol=btcusdt"
```

BSC

```
"https://openoracle_prod_bsc.dfuture.com/dev/web/sendOpenPosition"
"https://openoracle_prod_bsc.dfuture.com/dev/web/sendClosePosition"
"https://openoracle_prod_bsc.dfuture.com/dev/web/getLatesetPrice?symbol=btcusdt"
```

2. 增加了accessKey 和 accessSk, 每个client 需要申请自己的key

## 二. 范例

### Sample 地址

<https://github.com/dFuture-finance/dFuture-sdk-demo>

### 接口说明

#### OpenOrder 和 CloseOrder

这是我们提供了两个辅助类来帮助你生成交易数据。

构造参数

```
symbol,
amount,
direction,
acceptablePrice,
approvedUsdt,
parent,
withDiscount,
deadline,
maker,
gasLevel,
couponId,
couponAmount
```

构造参数类型

```
"bytes32",
"uint256",
"int8",
"uint256",
"uint256",
"address",
"bool",
"uint256",
"address",
"uint8",
"uint256",
"uint256"
```

### 参数解释

symbol, 交易标的, 'btc', 'eth'

amount, 交易数量, 单位是手, btc 一手是0.001 btc

direction, 交易方向, 1 做多, -1 做空

acceptablePrice, 能够接受的价格, 0代币任意价格

approveUsdt, approve 的可以被用作保证金的Usdt

parent, 交易账号的上级

withDiscount, 是否有折扣

deadline, 本交易的截止时间戳, 超过这个时间, 交易就无效, 即便上链也会失败

maker, 交易账号

gasLevel, gas price, 单位是gwei

注:

deadline 还包含了nonce 信息, 是该账号在dfuture的交易nonce, 每次开仓平仓都是递增的。它存放在deadline的高64位

## 基本使用方法

```
let openOrder = new OpenOrder(
    utils.formatBytes32String(config.symbol),
    config.handleAmount,
    ORDER_DIRECTION,
    config.ACCEPTABLE_PRICE,
    config.approveUsdt.toString() + e18str,
    config.PARENT_ADDRESS,
    config.WITH_DISCOUNT,
    deadline,
    ordermaker,
    config.GAS_LEVEL,
    1233,
    1
);
let args = await openOrder.toArgs(FUTURE_ADDRESS, makerPrivateKey, web3_rops, config.CHAIN_ID);
const params = {
    "symbol": args[0],
    "amount": args[1],
    "direction": args[2],
    "acceptablePrice": args[3],
    "approvedUsdt": args[4],
    "parent": args[5],
    "withDiscount": args[6],
    "deadline": args[7],
    "maker": args[8],
    "gasLevel": args[9],
    "couponId": args[10],
    "couponAmount": args[11],
    "r": args[13],
    "s": args[14],
    "v": args[12]
};
```

## 三. API签名

介绍API签名前, 需要明确两个签名概念

1. 账号公私钥, 就是普通的bsc, heco, 或者 以太的外部账号, 用来签署交易。
2. api 公私钥, 是用来授权调用dfuture 接口, 查询价格和发送签名后交易的。

本小节, 重点介绍api 公私钥。

### 1. 申请开发者权限获取accesskey

- 申请开发者权限, 平台会生成如下两个参数:
- accessKey (请求key标识): 身份识别
- accessSk (签名私钥): 验证token签名有效性

### 2. 签名验证

- 在所有接口调用时, 在header中增加如下三个字段

Aa Title	≡ 参数名	≡ 说明
----------	-------	------

Aa Title	≡ 参数名	≡ 说明
Untitled	accessKey	1中的身份信息
Untitled	accessTime	请求时本地的unix时间戳（毫秒），如 1616400809000
Untitled	token	签名token，验证请求合法性 签名规范 md5(accessKey&accessTime&accessSk)

- 例子
- 如 accesskey为 abc, accessSk为 def, accessTime 为 1616400809000(单位是毫秒)
- 拼接后字符串: abc&1616400809&def, 注意, '&' 是连接符, 用来连接三个字符串
- md5后生成的token为:
   
5c411238bbc9d3adf20c1b313bc31d4e

### 3. 签名和api 的使用

通过上述签名方法得到了token, 将以下信息作为post 消息, head 的一部分发送

```
{'Content-Type': 'application/json', 'accessKey': 'abc', 'accessTime': '1616400809000', 'token': '5c411238bbc9d3adf20c1b313bc31d4e'}
```

### 请求频率限制

- 系统会根据同一个accessKey在一个自然天内调用次数进行统计, 超过统计上限后当天不再响应请求

## 四. 查询价格

同样需要进行api 签名

btcusdt, 查询usdt 标价的btc的价格, ethusdt, 查询usdt 标价的eth的价格

Heco

```
"https://openoracle_prod_heco.dfutur.com/dev/web/getLatesetPrice?symbol=btcusdt"
```

BSC

```
"https://openoracle_prod_bsc.dfutur.com/dev/web/getLatesetPrice?symbol=btcusdt"
```

## 五. 查询用户和持仓信息

以下服务是由web3 提供, 我们将提供合约地址, 相关接口的abi 和 基本调用方法。

合约地址

Heco

0x917e091cc000012bbd58afFa8E6DbB96fa06cb0a

BSC

0xc67eC5cbcE3E9aB546CF6077Dd2ad519887737BA

### 获取用户持仓

#### 方法

- queryPosition

#### 请求参数

数据类型	描述
address	用户账户的公钥
byte32	请求标的的byte32格式，比如'btc'，就是0x627463

## 响应数据

数据类型	描述
uint104	持仓，单位，手，精度参考合约单位
uint104	持仓价格，单位USDT，18位精度
uint40	最近操作的block number
int8	交易方向， 多单 1， 空单 -1

## 示例

```
//返回 Promise
FPHANDLER.methods.queryPosition("0x859017F0FE975D18A6702415c52D428C6896c123",symbol('btc')).call()
```

## 获取账户信息

### 方法

- queryHolderInfo

## 请求参数

数据类型	描述
address	用户账户的公钥
byte32	请求标的byte32格式，比如'btc'，就是0x627463

## 响应数据

数据类型	描述
uint256	保证金，单位USDT，18位精度
uint256	保证金比率，18位精度
uint256	盈利金额，单位USDT，18位精度
uint256	仓位，单位USDT，18位精度

## 示例

```
//返回 Promise
FPHANDLER.methods.queryHolderInfo(OWNER,symbol('btc')).call();
```

## 六. 备注

### 合约单位

合约简称	BTC/USDT	ETH
交易单位	0.001BTC/手	0.01ETH/手
报价单位	USDT/手	USDT/手



合约币种持续更新中，具体请参考

<https://heco.dfutur.com/help/content/207000850421760>

注：单次最小交易数量\*10^精度 就是最小交易手数

比如 eth 精度2，一手就是0.01，单次最小交易数量0.1，最小交易手数是10

## 七. 线上接口

本节接口均需要通过web3 与链上合约直接交互

### 查询线上接口

```
FPHANDLER.functions.queryIfEmergency().call()
```

## 开仓

### 方法

- openPositionWithDeadLine()

### 请求参数

数据类型	描述
byte32	开仓标的
uint256	开仓数量，单位，手
int8	开仓方向
uint256	最不利情况下，可以接受的价格，单位 USDT/手, 0 任意价格
uint256	已经授权的USDT的金额, 单位USDT，带18位精度
address	上级的地址
bool	是否有折扣
uint	该交易的有效的截止时间戳

### 示例

```
//开仓eth, 100手（相当于1eth），开多，可以接受任意价格，为此授权了20000*1e6（相当于200000USD）
let trxdata = FPHANDLER.methods.openPosition(symbol('eth'),100, 1, 0,
(20000).U6(), ROOT_ADDRESS, 0, 1618396890).encodeABI()
execTrx(trxdata)
```

## 关仓

### 方法

- closePositionWithDeadLine()

### 请求参数

数据类型	描述
byte32	关仓标的
uint256	关仓数量，单位，手
uint256	最不利情况下，可以接受的价格，单位 USDT/手, 0 任意价格
uint	该交易的有效的截止时间戳

### 示例

```
//关掉eth的仓位, 100手，接受任意价格
let trxdata = FPHANDLER.methods.closePositionWithDeadLine(symbol('eth'), 100, 0, 1618396890).encodeABI()
execTrx(trxdata)
```