

1. OpenApi接口文档	2
1.1 /open/api/all_trade-获取全部成交记录	3
1.2 /open/api/cancel_order_all-根据币对取消全部委托单（最多取消两千条，多余两千请循环撤销）	4
1.3 /open/api/cancel_order-取消委托单	5
1.4 /open/api/common/symbols 查询系统支持的所有交易对及精度	5
1.5 /open/api/create_order-创建订单	6
1.6 /open/api/get_allticker 获取所有交易对行情	7
1.7 /open/api/get_records 获取K线数据	8
1.8 /open/api/get_ticker 获取当前行情	9
1.9 /open/api/get_trades 获取行情成交记录	10
1.10 /open/api/market_dept 查询买卖盘深度	11
1.11 /open/api/market-获取各个币对的最新成交价	12
1.12 /open/api/order_info-获取订单详情	12
1.13 /open/api/user/account-资产余额	13
1.14 /open/api/v2/all_order-获取全部委托	14
1.15 /open/api/v2/new_order-获取当前委托	15
1.16 接口调用代码演示（java）	17
1.17 链上私有open-api（需要超管权限）	21
1.17.1 /open/api/exchange_coin-金币兑换功能，业务走赠币	21

OpenAp i 接口文档

1、目录

2、通用规则

1) 签名：
请求参数按照字典排序，然后以keyvalue的形式拼接成字符串string，最后sign=MD5(string+secretKey)。注意：如果请求参数中value为NULL的情况，则在拼接字符串时不计入签名字符串。

例如：

参数如下：

```
{
'symbol': 'ltcbtc',
'api_key': '0816016bb06417f50327e2b557d39aaa',
'sign': 'a169740d0588141ef70b71cf11ff8bf3',
'time': '1522055680'
}
```

拼接完成后：

```
string=api_key0816016bb06417f50327e2b557d39aaasymbolltcbtctime1522055680
sign = MD5(string+secretKey) = MD5(api_key0816016bb06417f50327e2b557d39aaasymbolltcbtctime1522055680xxxxxxxxxxxxxxxxxx)
最终：sign=48105baf432a6626051a72fb160b617e
```

3、post请求参数采用表格式提交数据

```
content-type:application/x-www-form-urlencoded
```

4、错误码

错误码	说明	备注
0	成功	
5	下单失败	
6	数量小于最小值	
7	数量大于最大值	
8	订单取消失败	
9	交易被冻结	
13	对不起，程序出现系统错误，请和网站管理员联系	
19	可用余额不足	
22	订单不存在	
23	缺少交易数量参数	
24	缺少交易价格参数	
100001	系统异常	
100002	系统升级	
100004	请求参数不合法	
100005	参数签名错误	
100007	非法IP	

110002	未知的货币代号	
110003	资金密码错误	
110004	提现被冻结	
110005	可用余额不足	
110020	用户名不存在	
110023	手机号已注册	
110024	邮箱已注册	
110025	帐户被后台管理员锁定	
110032	该用户无权限进行此操作	
110033	充值失败	
110034	提现失败	

/open/api/all_trade-获取全部成交记录

/open/api/all_trade (get): 有成交的订单		
参数	填写类型	说明
symbol	必填	市场标记, btcusdt, 详情看下面
pageSize	选填	页面大小
page	选填	页码
api_key	必填	api_key
time	必填	时间戳
sort	选填	1表示倒序
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	{"如下	

```
{
  "count":22,
  "resultList":[
    {
      "volume":"1.000",
      "side":"BUY",
      "feeCoin":"YLB",
      "price":"0.10000000",
      "fee":"0.16431104",
      "ctime":1510996571195,
      "deal_price":"0.10000000",
      "id":306,
      "type":"买入",
      "bid_user_id":10001,
      "ask_user_id":10001
    },
    {
      "volume":"0.850",
      "side":"BUY",
      "feeCoin":"YLB",
      "price":"0.10000000",
      "fee":"0.13966438",
      "ctime":1510996571190,
      "deal_price":"0.08500000",
      "id":305,
      "type":"买入",
      "bid_user_id":10001,
      "ask_user_id":10001
    },
    {
      "volume":"0.010",
      "side":"BUY",
      "feeCoin":"YLB",
      "price":"0.10000000",
      "fee":"0.00164311",
      "ctime":1510995560344,
      "deal_price":"0.00100000",
      "id":291,
      "type":"买入",
      "bid_user_id":10001,
      "ask_user_id":10001
    }
  ]
}
```

虚拟币编号	xxx-cny	xxx-btc	xxx-usdt
btc	btcnny	-	btcusdt
eth	ethcny	ethbtc	ethusdt
ltc	ltcnny	ltcbtc	ltcusdt
bcc	bccnny	bccbtc	bccusdt
etc	etcnny	etcbtc	etcusdt
usdt	usdtnny	-	-

/open/api/cancel_order_all-根据币对取消全部委托单（最多取消两千条，多余两千请循环撤销）

/open/api/cancel_order_all (post)		
参数	填写类型	说明
symbol	必填	市场标记, ethbtc, 详情看下面
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	""	

虚拟币编号	xxx-cny (xxx101)	xxx-btc (xxx201)
btc	btccny	-
eth	ethcny	ethbtc
ltc	ltccny	ltcbtc
bcc	bcccnny	bccbtc
etc	etccny	etcbtc

/open/api/cancel_order-取消委托单

/open/api/cancel_order (post)		
参数	填写类型	说明
order_id	必填	订单ID
symbol	必填	市场标记, ethbtc, 详情看下面
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	""	

虚拟币编号	xxx-cny (xxx101)	xxx-btc (xxx201)
btc	btccny	-
eth	ethcny	ethbtc
ltc	ltccny	ltcbtc
bcc	bcccnny	bccbtc
etc	etccny	etcbtc

/open/api/common/symbols 查询系统支持的所有交易对及精度

/open/api/common/symbols (get) :		
参数	填写类型	说明
无		

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	<pre>{ "code": "0", "msg": "suc", "data": [{ "symbol": "ethbtc", "count_coin": "btc", "amount_precision": 3, "base_coin": "eth", "price_precision": 8 }, { "symbol": "ltcbtc", "count_coin": "btc", "amount_precision": 2, "base_coin": "ltc", "price_precision": 8 }, { "symbol": "bchbtc", "count_coin": "btc", "amount_precision": 3, "base_coin": "bch", "price_precision": 8 }, { "symbol": "etcbtc", "count_coin": "btc", "amount_precision": 2, "base_coin": "etc", "price_precision": 8 }, { "symbol": "ltceth", "count_coin": "eth", "amount_precision": 2, "base_coin": "ltc", "price_precision": 8 }, { "symbol": "etceth", "count_coin": "eth", "amount_precision": 2, "base_coin": "etc", "price_precision": 8 }] }</pre>	symbol 交易对 base_coin 基础币种 count_coin 计价货币 price_precision 价格精度位数 (0为个位) amount_precision 数量精度位数 (0为个位)

/open/api/create_order-创建订单

/open/api/create_order (post)		
参数	填写类型	说明
side	必填	买卖方向BUY、SELL
type	必填	挂单类型, 1: 限价委托、2: 市价委托

volume	必填	购买数量（多义，复用字段） type=1:表示买卖数量 type=2:买则表示总价格，卖表示总个数 买卖限制 user/me-用户信息
price	选填	委托单价：type=2：不需要此参数
symbol	必填	市场标记，ethbtc
fee_is_user_exchange_coin	选填	0，当交易所有平台币时，此参数表示是否使用平台币支付手续费，0否，1是
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	{"order_id":34343}	成功返回交易ID

虚拟币编号	xxx-cny（xxx101）	xxx-btc（xxx201）
btc	btccny	-
eth	ethcny	ethbtc
ltc	ltccny	ltcbtc
bcc	bcccny	bccbtc
etc	etccny	etcbtc

/open/api/get_allticker 获取所有交易对行情

该接口不进行签名校验

/open/api/get_allticker（get）：获取当前行情		
参数	填写类型	说明
无		

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	如下：	返回值说明 date：返回数据时服务器时间 symbol：交易对（交易对1(base)简称_交易对2(quote)简称） buy：买一价 high：最高价 last：最新成交价 low：最低价 sell：卖一价 vol：成交量（最近的24小时） rose：涨跌幅

```

{
  "date": 1534335607859,
  "ticker": [
    {
      "symbol": "btcusdt",
      "high": 7408.35984546,
      "vol": 0.01,
      "last": 1,
      "low": 7408.35984546,
      "buy": "3700.00000000",
      "sell": "7408.35984546",
      "rose": 0
    },
    {
      "symbol": "ethusdt",
      "high": 535.96,
      "vol": 6366.8591,
      "last": 20,
      "low": 279.57,
      "rose": -0.44564773
    },
    {
      "symbol": "bchusdt",
      "high": 12,
      "vol": 100,
      "last": 1,
      "low": 12,
      "buy": "11.00",
      "sell": "9.00",
      "rose": 0
    },
    {
      "symbol": "ethbtc",
      "high": 1,
      "vol": 281261,
      "last": 0.1,
      "low": 0.044039,
      "buy": "0.044049",
      "sell": "0.044049",
      "rose": -0.00022701
    },
    {
      "symbol": "nobbtc",
      "high": 0.007419,
      "vol": 1998,
      "last": 1,
      "low": 0.007419,
      "sell": "0.00741900",
      "rose": 0
    },
    {
      "symbol": "ltceth",
      "last": 0.18519949,
      "buy": "1.00000000",
      "sell": "0.18328001"
    }
  ]
}

```


/open/api/get_records 获取K线数据

该接口不进行签名校验

/open/api/get_records (get)：获取K线数据		
参数	填写类型	说明
symbol	必填	市场标记，bchbtc，详情看下面
period	必填	单位为分钟，比喻1分钟则为1，一天则为1440

虚拟币编号	xxx-cny	xxx-btc	xxx-usdt
bch	bcccnym	bchbtc	bchusdt
btc	btccnym	-	btcusdt
etc	etccnym	etcbtc	etcusdt
eth	ethcnym	ethbtc	ethusdt
ltc	ltccnym	ltcbtc	ltcusdt

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	如下：	

```
[
  [
    1514445780, //时间戳
    1.12, //开盘价
    1.12, //最高
    1.12, //最低
    1.12, //收盘价
    0 //成交量
  ],
  [
    1514445840,
    1.12,
    1.12,
    1.12,
    1.12,
    0
  ],
  [
    1514445900,
    1.12,
    1.12,
    1.12,
    1.12,
    0
  ]
]
```

/open/api/get_ticker 获取当前行情

该接口不进行签名校验

/open/api/get_ticker (get)：获取当前行情		
参数	填写类型	说明
symbol	必填	市场标记，btcusdt，详情看下面

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	如下：	

```
{
  "high": 1, //最高值
  "vol": 10232.26315789, //交易量
  "last": 173.60263169, //最新成交价
  "low": 0.01, //最低值
  "buy": "0.01000000", //买一价
  "sell": "1.12345680", //卖一价
  "rose": -0.44564773, //涨跌幅
  "time": 1514448473626
}
```

虚拟币编号	xxx-cny	xxx-btc	xxx-usdt
bcc	bcccnym	bccbtc	bccusdt
btc	btccnym	-	btcusdt
etc	etccnym	etcbtc	etcusdt
eth	ethcnym	ethbtc	ethusdt
ltc	ltccnym	ltcbtc	ltcusdt

/open/api/get_trades 获取行情成交记录

该接口不进行签名校验

/open/api/get_trades (get)：获取行情成交记录		
参数	填写类型	说明
symbol	必填	市场标记，bchbtc，详情看下面

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	如下：	

```
[
  {
    "amount": 0.55, //成交量
    "price": 0.18519949, //成交价
    "id": 447121,
    "type": "buy" //买卖type, 买为buy, 买sell
  },
  {
    "amount": 16.45,
    "price": 0.18335468,
    "id": 447120,
    "type": "buy"
  },
  {
    "amount": 2,
    "price": 0.18335468,
    "id": 447119,
    "type": "buy"
  },
  {
    "amount": 2.92,
    "price": 0.183324003,
    "id": 447118,
    "type": "sell"
  }
]
```

虚拟币编号	xxx-cny	xxx-btc	xxx-usdt
bch	bcccnym	bchbtc	bchusdt
btc	btccnym	-	btcusdt
etc	etccnym	etcbtc	etcusdt
eth	ethcnym	ethbtc	ethusdt
ltc	ltccnym	ltcbtc	ltcusdt

/open/api/market_dept 查询买卖盘深度

改接口不进行签名校验

/open/api/market_dept (get):

参数	填写类型	说明
symbol	必填	市场标记, ethbtc, 详情看下面
type	必填	深度类型, step0, step1, step2 (合并深度0-2); step0时, 精度最高

返回

字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	见下面代码:	

```
{
  "tick": {
    "asks": [//卖盘
      {22112.22, 0.9332},
      {22112.21, 0.2},
      {22112.21, 0.2},
      {22112.21, 0.2},
      {22112.21, 0.2},
    ],
    "bids": [//买盘
      {22111.22, 0.9332},
      {22111.21, 0.2},
      {22112.21, 0.2},
      {22112.21, 0.2},
      {22112.21, 0.2},
    ]
  }
}
```

/open/api/market-获取各个币对的最新成交价

/open/api/market (get):

参数	填写类型	说明
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回

字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	{"btcusdt":15000,"ethusdt":800}	

虚拟币编号	xxx-cny	xxx-btc	xxx-usdt
btc	btccny	-	btcusdt
eth	ethcny	ethbtc	ethusdt
ltc	ltccny	ltcbtc	ltcusdt
bcc	bccny	bccbtc	bccusdt
etc	etccny	etcbtc	etcusdt

jira:<http://jira.365os.com/browse/BIKI-13>去掉 api_key和sign的校验

/open/api/order_info-获取订单详情

/open/api/order_info (get):

参数	填写类型	说明
order_id	必填	订单ID
symbol	必填	市场标记, ethbtc, 详情看下面

api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	{ "order_info" : {}, "trade_list": [] }	

```
{
  "order_info": {
    "id": 343,
    "side": "sell",
    "side_msg": "卖出",
    "created_at": "09-22 12:22",
    "price": 222.33,
    "volume": 222.33,
    "deal_volume": 222.33,
    "total_price": 222.33,
    "fee": 222.33,
    "avg_price": 222.33
  }
  "trade_list": [
    {
      "id": 343,
      "created_at": "09-22 12:22",
      "price": 222.33,
      "volume": 222.33,
      "deal_price": 222.33,
      "fee": 222.33
    },
    {
      "id": 345,
      "created_at": "09-22 12:22",
      "price": 222.33,
      "volume": 222.33,
      "deal_price": 222.33,
      "fee": 222.33
    }
  ]
}
```

/open/api/user/account-资产余额

/open/api/user/account (get)		
参数	填写类型	说明
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		

字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	{ "total_asset":432323.23, "coin_list":[{"coin":"btc","normal":32323.233,"locked":32323.233,"btcValuat in":112.33}, {"coin":"ltc","normal":32323.233,"locked":32323.233,"btcValuat in":112.33}, {"coin":"bch","normal":32323.233,"locked":32323.233,"btcValuat in":112.33},] }	total_asset:总资产 normal:余额账户 locked:冻结账户 btcValuat in: BTC估值

/open/api/v2/all_order-获取全部委托

旧接口/open/api/all_order 仍保留, 但已经不建议使用

v2版本变化: 去掉了结果返回值中的tradeList成交记录, 提升效率; 如果需要单一订单的成交信息, 可以使用/open/api/order_info 接口单独去查

/open/api/v2/all_order (get): 已成交和已取消		
参数	填写类型	说明
symbol	必填	市场标记, btcusdt, 详情看下面
pageSize	选填	页面大小
page	选填	页码
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	如下:	

```
{
  "count":10,
  "orderList":[
    {
      "side":"BUY",
      "total_price":"0.10000000",
      "created_at":1510993841000,
      "avg_price":"0.10000000",
      "countCoin":"btc",
      "source":1,
      "type":1,
      "side_msg":"买入",
      "volume":"1.000",
      "price":"0.10000000",
      "source_msg":"WEB",
      "status_msg":"完全成交",
      "deal_volume":"1.00000000",
      "id":424,
      "remain_volume":"0.00000000",
      "baseCoin":"eth",
      "status":2
    },
    {
      "side":"SELL",
      "total_price":"0.09900000",
      "created_at":1510993715000,
      "avg_price":"0.10000000",
      "countCoin":"btc",
      "source":1,
      "type":1,
      "side_msg":"卖出",
      "volume":"1.000",
      "price":"0.09900000",
      "source_msg":"WEB",
      "status_msg":"完全成交",
      "deal_volume":"1.00000000",
      "id":423,
      "remain_volume":"0.00000000",
      "baseCoin":"eth",
      "status":2
    }
  ]
}
```

虚拟币编号	xxx-cny	xxx-btc	xxx-usdt
btc	btccny	-	btcusdt
eth	ethcny	ethbtc	ethusdt
ltc	ltccny	ltcbtc	ltcusdt
bcc	bcccnny	bccbtc	bccusdt
etc	etccny	etcbtc	etcusdt
usdt	usdtcny	-	-

/open/api/v2/new_order-获取当前委托

旧接口/open/api/new_order 仍保留, 但已经不建议使用

v2版本变化：去掉了结果返回值中的tradeList成交记录, 提升效率; 如果需要单一订单的成交信息, 可以使用/open/api/order_info接口单独去查

/open/api/v2/new_order (get)：包括未成交和正在成交的委托		
参数	填写类型	说明
symbol	必填	市场标记, btcusdt, 详情看下面
pageSize	选填	页面大小
page	选填	页码
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	如下图	订单状态(status)说明： INIT(0, "初始订单，未成交未进入盘口"), NEW_(1, "新订单，未成交进入盘口"), FILLED(2, "完全成交"), PART_FILLED(3, "部分成交"), CANCELED(4, "已撤单"), PENDING_CANCEL(5, "待撤单"), EXPIRED(6, "异常订单")；


```
{
  "count":10,
  "resultList":[
    {
      "side":"BUY",
      "total_price":"0.10000000",
      "created_at":1510993841000,
      "avg_price":"0.10000000",
      "countCoin":"btc",
      "source":1,
      "type":1,
      "side_msg":"买入",
      "volume":"1.000",
      "price":"0.10000000",
      "source_msg":"WEB",
      "status_msg":"完全成交",
      "deal_volume":"1.00000000",
      "id":424,
      "remain_volume":"0.00000000",
      "baseCoin":"eth",
      "status":2
    },
    {
      "side":"SELL",
      "total_price":"0.09900000",
      "created_at":1510993715000,
      "avg_price":"0.10000000",
      "countCoin":"btc",
      "source":1,
      "type":1,
      "side_msg":"卖出",
      "volume":"1.000",
      "price":"0.09900000",
      "source_msg":"WEB",
      "status_msg":"完全成交",
      "deal_volume":"1.00000000",
      "id":423,
      "remain_volume":"0.00000000",
      "baseCoin":"eth",
      "status":2
    }
  ]
}
```

虚拟币编号	xxx-cny	xxx-btc	xxx-usdt
btc	btccny	-	btcusdt
eth	ethcny	ethbtc	ethusdt
ltc	ltccny	ltcbtc	ltcusdt
bcc	bcccny	bccbtc	bccusdt
etc	etccny	etcbtc	etcusdt
usdt	usdtcny	-	~/web/new_order~获取当前委托

接口调用代码演示（java）

```
package test;
import java.io.InputStream;
import java.io.UnsupportedEncodingException;
```

```

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Date;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;
import java.util.TreeMap;
import org.apache.commons.httpclient.HttpClient;
import org.apache.commons.httpclient.methods.PostMethod;
import org.apache.commons.io.IOUtils;
/**
 * openapi的post接口调用实例
 * 已/open/api/cancel_order 为例
 *
 */
public class OpenApiTest {
/**
 *
 * @param args
 */
public static void main(String[] args) {
/** 4个变量, api_key, secret_key */
String api_key = "60c88d91c8e33a4b4d42777596e0e8c";
String secret_key = "23937296ab9205219646aac913d06e05";
Long order_id = 542129L;
String symbol = "btcusdt";

/** 封装需要签名的参数 */
TreeMap<String, String> params = new TreeMap<String, String>();
params.put("order_id", order_id.toString());
params.put("symbol", symbol);
params.put("api_key", api_key);
params.put("time", new Date().getTime()+"");

/** 拼接签名字符串, md5签名 */
StringBuilder result = new StringBuilder();
Set<Entry<String, String>> entrys = params.entrySet();
for (Entry<String, String> param : entrys) {
/** 去掉签名字段 */
if(param.getKey().equals("sign")){
continue;
}

/** 空参数不参与签名 */
if(param.getValue()!=null) {
result.append(param.getKey());
result.append(param.getValue().toString());
}
}
result.append(secret_key);
String sign = getMD5(result.toString());
params.put("sign", sign);

/** http请求 */
String resultJson = post("https://[网站api访问域名]/exchange-open-api/open/api/cancel_order",
params);
System.out.println(resultJson);
}

/**

```

```

* 通过post来提交数据，带参数的方法
*
* @param url 请求地址
* @param params 参数
* @return
*/
public static String post(String url, Map<String, String> params) {
    String str = null;
    try {
        HttpClient client = new HttpClient();
        PostMethod method = new PostMethod(url);
        //设定请求头的样式
        method.setRequestHeader("Content-Type", "application/x-www-form-urlencoded;charset=utf-8");
        if (params != null && params.size() > 0) {
            for (Map.Entry<String, String> entry : params.entrySet()) {
                method.setParameter(entry.getKey(), entry.getValue());
            }
        }
        int code = client.executeMethod(method);
        if (code >= 200 && code < 300) {
            InputStream in = method.getResponseBodyAsStream();
            str = IOUtils.toString(in);
        }
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return str;
}

/**
* 获取String的MD5值
*
* @param info 字符串
* @return 该字符串的MD5值
*/
public static String getMD5(String info) {
    try {
        //获取 MessageDigest 对象，参数为 MD5 字符串，表示这是一个 MD5 算法（其他还有 SHA1 算法等）：
        MessageDigest md5 = MessageDigest.getInstance("MD5");
        //update(byte[])方法，输入原数据
        //类似StringBuilder对象的append()方法，追加模式，属于一个累计更改的过程
        md5.update(info.getBytes("UTF-8"));
        //digest()被调用后，MessageDigest对象就被重置，即不能连续再次调用该方法计算原数据的MD5值。可以手动调用reset()方法重置输入源。
        //digest()返回值16位长度的哈希值，由byte[]承接
        byte[] md5Array = md5.digest();
        //byte[]通常会转化为十六进制的32位长度的字符串来使用，本文会介绍三种常用的转换方法
        return bytesToHex(md5Array);
    } catch (NoSuchAlgorithmException e) {
        return "";
    } catch (UnsupportedEncodingException e) {
        return "";
    }
}

private static String bytesToHex(byte[] md5Array) {
    StringBuilder strBuilder = new StringBuilder();
    for (int i = 0; i < md5Array.length; i++) {
        int temp = 0xff & md5Array[i];
        String hexString = Integer.toHexString(temp);
    }
}

```

```
if (hexString.length() == 1) { //如果是十六进制的0f，默认只显示f，此时要补上0
    stringBuilder.append("0").append(hexString);
} else {
    stringBuilder.append(hexString);
}
}
```

```
return strBuilder.toString();
}
}
```

链上私有open-api（需要超管权限）

1. 目录：

[/open/api/present_coin](#)-金币兑换功能，业务走赠币

[/open/api/exchange_coin](#)-金币兑换功能，业务走赠币

需求描述.txt特殊逻辑：

1、同一exchange_id不能重复赠币（调多次接口，只赠一次）

2、此接口禁止通用api-key调用，需要使用管理员权限的api调用

/open/api/exchange_coin (POST)：		
参数	填写类型	说明
uid	必填	用户id
amount	必填	金币兑换金额
symbol	必填	金币兑换币种
exchange_id	必填	金币兑换业务id
api_key	必填	api_key
time	必填	时间戳
sign	必填	签名

返回		
字段	实例	解释
code	0	
msg	"suc"	code>0失败
data	{"convert_id":1006}	成功返回兑换记录id

返回码描述		
返回码code	返回信息msg	解释
0	suc	处理成功
100001	system error	系统异常
100004	request parameter illegal	请求参数不合法
100005	request sign illegal	参数签名错误
100007	ip illegal	非法IP
110032	no authority for user	该用户无权限进行此操作
110020	Wrong username or password	用户名不存在
110040	The current currency is not yet open to exchange	当前币种尚未开放金币兑换