

流量管理平台API文档

1.请求和响应报文
2.API参数说明
2.1. 签名算法
2.2. 签名工具类
3.消息单元定义
3.1. 单号码充流量
3.1.1 单号码充流量(测试用接口)
3.2. 推送状态报告
3.3. 查询状态报告
4.错误定义

1.请求和响应报文

接口协议	
说明	描述
网络协议	HTTP
请求方法	POST
请求格式	application/json
应答格式	JSON
字符集	UTF8
ApiKey	请咨询商务联系人员或查看平台中用户基本信息页

请注意:提交请求的HTTP头部分中,ContentType请指定为application/json

2.API参数说明

参数分为系统参数及业务参数，系统参数为区分客户信息及安全鉴权使用，业务参数为接口服务的具体选项。

名称	类型	是否必须	描述
apiKey	String	Y	客户第三方身份标识
securityKey	String	Y	客户接口签名认证用密钥
timeStamp	String	Y	当前时间戳 格式:yyyyMMddHHmmss
sign	String	Y	对指定参数的加密签名

apiKey 与 securityKey

apiKey是管理平台API分配给客户端厂商身份的唯一标识，在客户端厂商申请调用客户端API系统API开发权限时，由管理平台管理员分配。与apiKey同时分配的还有securityKey（密钥）,请接入用户登录到平台在用户账户页获取 apiKey。

timeStamp

当前时间戳 格式为yyyyMMddHHmmss 如20160626014303

sign

签名，客户将接口列表中的参数按照指定方式进行签名,签名方式使用SHA1,具体签名方案参见文档签名算法;由客户平台按照规范签名后传入。

2.1. 签名算法

签名生成规则是根据用户请求参数、值(不包括sig本身)结合分配给各客户端厂商的密钥，使用SHA1算法，生成数字签名信息；签名生成时，首先按照参数名（KEY）做升序排序，然后对参数名和参数值（使用utf-8编码，特别注意包含中文的参数值的编码）联接成字符串；然后在最后加上密码串生成签名

例如:签名是经过密码串与请求参数数据结合而成的。下面以订购接口为例：
假设订购接口的参数，该客户端获得的密钥是 securityKey=test0002

```
String apiKey="ryan001"  
String timeStamp="20170222184924"  
String phone="13128758237"  
String productCode="FU0310010M"  
String notifyUrl="http://localhost:8888/callme/call"  
String cstmOrderNo="FD00012876"  
String sign （需要根据以上值生成签名）
```

对以上参数（去除sign）
1.首先key升序排序：
apiKey,cstmOrderNo,notifyUrl,phone,productCode,timeStamp
2.再按照key获取value并将key和value拼接：
apiKeyryan001cstmOrderNoFD00012876notifyUrlhttp://localhost:8888/callme/callphone13128758237productCodeFU0310010MtimeStamp20170222184924
(http://localhost:8888/callme/callphone13128758237productCodeFU0310010MtimeStamp20170222184924)
2.1 如notifyUrl不提供,则采用空字符串
apiKeyryan001cstmOrderNoFD00012876notifyUrlcallphone13128758237productCodeFU0310010MtimeStamp20170222184924
3.将该客户端厂商的密钥拼接在 第2步得到了的字符串后面：
appkeyxiaowangcallbackUrlhttp://localhost:8888/callme/callcstmOrderNo123457phoneNo13128758237productldfu0340003gtimeStamp20160626014459test0002
(http://localhost:8888/callme/callcstmOrderNo123457phoneNo13128758237productldfu0340003gtimeStamp20160626014459test0002)
4.最后第3步得到的字符串进行SHA1签名，生成签名：
sign = ecabb01c075582027658d869f374690cf04fdab1

针对查询状态报告接口的举例

```
String apiKey="dis123456"  
String timeStamp="20170619163014"  
String order_no="11497863173539102435"  
String sign （需要根据以上值生成签名）
```

对以上参数（去除sign）
1.首先key升序排序：
apiKey,order_no,timeStamp
2.再按照key获取value并将key和value拼接：

apiKeydis123456order_no11497863173539102435timeStamp20170619163014
3.将该客户端厂商的秘钥拼接在 第2步得了到的字符串后面：
apiKeydis123456order_no11497863173539102435timeStamp20170619163014test0002
4.最后第3步得到的字符串进行SHA1签名，生成签名：
sign = 1cddb8e5376bb3abd9691c142d29d4fdac6cf8a9

2.2. 签名工具类

```
/**
 * SHA加密参考
 */
public class SignatureUtil {

    private Decrypt signUtil = new Decrypt();

    public String sign(String str) {
        return signUtil.shaEncrypt(str);
    }

    private class Decrypt {
        public final String KEY_SHA = "SHA";
        char[] Digit = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };

        /**
         * SHA加密
         */
        public String shaEncrypt(String inputStr) {
            byte[] inputData = inputStr.getBytes();
            String returnString = "";
            try {
                inputData = encryptSHA(inputData);
                for (int i = 0; i < inputData.length; i++) {
                    returnString += byteToHexString(inputData[i]);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
            return returnString;
        }

        private String byteToHexString(byte ib) {
            char[] ob = new char[2];
            ob[0] = Digit[(ib >>> 4) & 0X0F];
            ob[1] = Digit[ib & 0X0F];

            String s = new String(ob);

            return s;
        }

        /**
         * SHA加密字节
         */
        public byte[] encryptSHA(byte[] data) throws Exception {
            MessageDigest sha = MessageDigest.getInstance(KEY_SHA);
            sha.update(data);
            return sha.digest();
        }
    }
}
```

3.消息单元定义

3.1. 单号码充流量

API 请求地址

http://host/open-api/rest/recharge

请求		
元素编码	元素名称	说明
apiKey	商户唯一标识	
timeStamp	时间戳	格式为yyyyMMddHHmmss 如20160626014303
sign	签名	签名方案参见文档签名算法
phone	手机号码	11位充值号码
productCode	套餐编码	流量包
notifyUrl	回调地址	小于200个字符(可为空)
cstmOrderNo	商户订单号	商户系统内部的订单号,64个字符内、可包含字母，不能重复

响应		
元素编码	元素名称	说明
code	接口状态编码	0000为成功 0001为失败
msg	描述	
data	返回业务数据	code为0001时，无data数据
orderNo	服务方系统订单流水号	32位
cstmOrderNo	调用方流水号	
status	业务状态	0：订单已受理 8失败
errorDesc	错误描述	当status为8时不为空

接口范例

```
请求报文：
{
  "apiKey": "huawei",
  "notifyUrl": "http://localhost:8888/callme/call",
  "cstmOrderNo": "123457",
  "phone": "13128758237",
  "productCode": "FU03010050M",
  "sign": "8753669045cb847dc1631b12a335c1e7acc295df",
  "timeStamp": "20160626014459"
}
返回报文：
{
  "code": "0000",
  "msg": "成功",
  "data": {
    "status": "0",
    "orderNo": "12334354545454",
    "cstmOrderNo": "13512345631",
    "errorDesc": "受理中"
  }
}
{
  "code": "0001",
  "msg": "账号余额不足"
}
```

3.1.1 单号码充流量(测试用接口)

API 请求地址

http://host/open-api/test/recharge

请求及参数参见3.1
此接口仅用于调试接口请求参数,不会产生实际订单数据

3.2. 推送状态报告

接口用户提供接收http post推送的url地址。接收到推送数据解析成功后响应ok，即为推送成功，否则每分钟尝试重新发送，尝试三次。推送数据格式为json数组。

对返回结果标记为处理中的订单,请以后继明确状态同步为准
接口返回系统异常错误或者连接超时错误, 请按照受理中状态处理

请求		
元素编码	元素名称	说明
orderNo	订单流水号	
cstmOrderNo	客户流水号	
status	状态	0007：订购成功，0008：订购失败，0009：订单处理中
msg	描述	失败描述

请求		
元素编码	元素名称	说明
code	接口状态	ok
msg	描述	

接口范例

```
请求报文：
{
  "status": "0007",
  "orderNo": "123343545454",
  "cstmOrderNo": "13512345631",
  "msg ": "订购成功"
}
返回报文：
{
  "code": "ok",
  "msg": "接收成功"
}
```

3.3. 查询状态报告

API 请求地址

http://host/open-api/rest/status

提供渠道主动查询某一订单状态的能力

请求		
元素编码	元素名称	说明
apiKey	商户唯一标识	
orderNo	订单流水号	
timeStamp	时间戳	格式为yyyyMMddHHmmss 如20160626014303

请求		
sign	签名	签名方案参见文档签名算法

请求		
元素编码	元素名称	说明
code	接口状态	
msg	描述	

接口范例

```
请求报文：
{
  "apiKey": "dis123456",
  "timeStamp": "20170619163014",
  "order_no": "11497863173539102435",
  "sign": "38d88b0c18eab68ba4f3fa19057832d3d65368d5"
}
返回报文：
{
  "code": "0007",
  "msg": "订购成功"
}
```

4.错误定义

错误码	描述
提交阶段	
0000	受理成功
0001	受理失败,提交订单错误见msg字段
0008	受理失败,提交订单错误见msg字段
状态报告阶段	
0007	订购成功
0008	订购失败,提交订单错误见msg字段
0009	订单处理中,等待后续状态推送

...read more →