API 调用方法说明

API 是基于 HTTP 协议来调用的，开发者需根据接口说明来封装 HTTP 请求进行调用。接口调用数据参数采取 HTTP 方式调用，请求数据需要加密加签，数据加密采用 AES 加密和双向 RSA 加密算法，加签采用 RSA 签名，详见本文档 4、5 两个小节。

1. 接入流程

1) 壹账通为调用方分配 orgCode，channelId。
2) 壹账通为调用方分配{服务端 RSA 公钥}。
3) 调用方自行产生{客户端 RSA 公私钥对}，并将{客户端 RSA 公钥}反馈给壹账通。
4) 调用方请求 API。

2. 接口地址

生产公网：https://ipms.pingan.com/openapi/{serviceName}/{method}
url 参数说明：
{serviceName}-服务名称，详细参考《壹账通业务服务接口说明-xxx 接口》
{method}-方法名称，详细参考《壹账通业务服务接口说明-xxx 接口》

3. 公共参数

post 请求参数：

字段名称	类型	必须	说明
orgCode	String	是	机构 ID，由壹账通分配
channelId	String	是	渠道 ID，由壹账通分配
requestId	String	是	接口请求 ID，长度最长为 64 位，数字和字符串的组合，客户端请求唯一标识（建议用 UUID）,客户端生成，每次接口请求的 requestId 都是不同的
encodeKey	String	是	AES 密钥的密文，产生方式：AES 秘钥明文为调用方随机产生 16 位长度字符串，并且使用{服务端 RSA 公钥}加密产生
sign	String	是	签名（对除 sign 外所有非 null 字段，以参数字段名称进行字母排序进行加签），签名实例参考本文 5 小节
signMethod	String	否	签名的摘要算法，默认值为：RSAWITHSHA256
timestamp	String	是	yyyy-MM-dd HH:mm:ss,时区 GMT+8，linux 时间戳，精确到毫秒（System.currentTime）
format	String	否	接口报文响应格式，默认为 json
version	String	是	接口版本号，默认传 1.0
requestData	String	是	json 格式业务请求数据的 AES 加密密文，其中入参格式详细

			参考《壹账通业务服务接口说明-xxx 接口》，加密说明请参考本文 1.4 小节

返回参数：

字段名称	类型	必须	说明
responseCode	String	是	响应码
responseMessage	String	是	响应码说明
responseData	String	是	响应数据

4. 加密算法

1.AES key 随机产生 16 位长度字符串；
2.对接口的业务参数 json 格式数据使用 AES 加密，生成 requestData 的值
3.AES key 采用 RSA 公钥加密，生成 encodeKey 的值
4.AES 加密明文示例：{"productId":"xxx","productCode":"xxxx"...}；
5.RSA 公钥由壹账通颁发
6.返回报文解密

5. 签名算法

签名使用所有通用入参及业务入参按顺序排序后使用 sha256 进行加签，网关使用相同的方式对签名进行校验。

```
public static String sign(Map<String, String> params) {
    Map<String, String> signDatas = new TreeMap<String, String>();
    signDatas.put("orgCode", params.get("orgCode"));
    signDatas.put("channelId", params.get("channelId"));
    signDatas.put("requestId", params.get("requestId"));
    signDatas.put("signMethod", params.get("signMethod"));
    signDatas.put("timestamp", params.get("timestamp"));
    signDatas.put("format", params.get("format"));
    signDatas.put("version", params.get("version"));
    signDatas.put("encodeKey", params.get("encodeKey"));
    signDatas.put("requestData", params.get("requestData"));

    String sign = RSAUtils.signwithsha256(signDatas.toString(), CLIENT_PRIVATE_KEY, "UTF-8");
    return sign;
}
```

6. 调用示例

URL: https://ipms.pingan.com/openapi/serviceName/method
POST 请求数据

```
{
```

```
    "encodeKey": "U3G5cleIBTD78WMh8MmR83OzHej7NgxjcD/yvoiyWCC9cf4eO7PWqDHQVBVCx6koW4azJvKILgvE4LlaYDMLd9I8+OY2KmtIwsUnjz6xHTV6NkBIaej1eglGY1LuOyRs4YUuuwQ5kXVrNVt6DLaFDi2sV5tJ/uVuV7l0riHAZdQ=",
    "format": "json",
    "orgCode": "API",
    "requestData": "D+DVKuWDQeCeoIsYd5GAC4dNkbD...2T69o8VYhYLSpvbv+aNxDg==",
    "requestId": "1545618329533",
    "version": "1.0",
    "signMethod": "SHA1WithRSA",
    "timestamp": "2018-12-24 10:25:29",
    "sign": "gSmIc6/TUMqmli9boddvzr458xN5Qpp5RysR03+ceKzAkdySAsOspSFeCKZoBYaqWWWjZwQS26zxIZ/FBVkUqS4r8UmH5cR+0QaNtYvqhjD+d/APSb+inB2DKddKcwD8gTigOTr0CqjO6hXnn51yFxl8WhvogmDcsVbOtGsx8mE=",
    "channelId": "1"
}
```


正确返回示例
Content-type: application/json;charset=UTF-8


```
{
    "responseCode ": "000000",
    "responseMessage": "success",
    "responseData ": "/Ln280NAA95uImZ0BJdps+MI+GexMFvBJ9MVrBRO3RY="
}
```


4.7 错误返回示例


```
{
    "responseCode ": "999999",
    "responseMessage": "failed",
    "responseData ": ""
}
```


7. 公共层面错误码

错误码	错误码描述	解决方案
999999	未知错误	
910001	非法参数	
910004	请求参数为空	
900013	验证签名失败	
940003	机构号或者渠道号不正确	
940004	请求太频繁	