

# 接入说明文档

版 本 号: V2.0.4

编制日期: 2020-06-12

更改日期: 2020-06-12

## 目录

1. 前言.....	3
1.1. 简介.....	3
1.2. 业务流程.....	3
2. Android 接入 .....	4
2.1. 安装.....	4
2.1.1. 环境准备.....	4
2.1.2. 安装 SDK.....	4
2.2. 快速入门.....	5
2.2.1. 初始化.....	5
2.2.2. 示例.....	5
2.3. 异常处理.....	6
3. IOS 接入 .....	6
3.1. 安装.....	6
3.1.1. 环境准备.....	6
3.1.2. 安装 SDK.....	6
3.2. 快速入门.....	9
3.2.1. 初始化.....	9
3.2.2. 示例.....	10
3.3. 编译打包.....	10
4. 服务器接入.....	10
4.1. 数据鉴权.....	10
4.2. 接口说明.....	11
4.3. 主动请求类接口.....	11
4.3.1. 请求说明.....	11
4.3.2. 加盟报备.....	12
4.3.3. 子工号报备.....	13
4.3.4. 删除子工号.....	14
4.3.5. 修改手机号成功.....	15
4.3.6. 退加盟.....	16
4.3.7. 开空卡订单查询.....	16
4.3.8. 激活卡订单查询.....	19
4.3.9. 成卡订单查询.....	21
4.3.10. 查询用户信息.....	23
4.4. 通知类接口.....	25
4.4.1. 通知策略.....	25
4.4.2. 通知格式.....	25
4.4.3. 返回码说明.....	34
4.4.4. 异常处理.....	34

# 1. 前言

本文档适用于 SDK 调用、服务器接入。

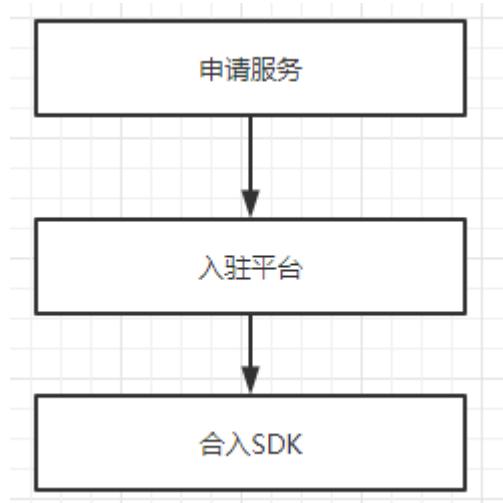
## 1.1. 简介

使用 SDK 之前,您需要:

- 了解并开通服务(联系客服)。
- 获得 appId、appKey,配置数据信息。

本文档详细介绍了一下 SDK 的安装、开发操作、参数、示例和常见问题处理, 以及服务器接入接口。

## 1.2. 业务流程



## 2. Android 接入

### 2.1. 安装

#### 2.1.1. 环境准备

本项目使用 Android Studio 进行开发，依赖 Android Support V4 库、RecyclerView、RxJava2、ZXing、Kalle 为支援库

#### 2.1.2. 安装 SDK

在主模块目录下的 libs 目录(如果没有则创建 libs 目录)下导入核心库 KMSdk.aar，以及  
依赖库 AlipaySdk.aar (如果项目已经依赖请忽略)、WbCloudFaceReflectLiveSdk.aar(腾讯  
云活体识别)、 WbCloudNormal.aar(腾讯云活体识别)。

1、项目 build.gradle 中添加 repositories

```
repositories {  
    flatDir {  
        dirs('libs')  
    }  
}
```

2.项目 build.gradle 中添加 dependencies:

```
implementation 'com.android.support:appcompat-v7:28.0.0'  
implementation 'com.android.support:recyclerview-v7:28.0.0'  
implementation "io.reactivex.rxjava2:rxjava:2.2.12"  
implementation "io.reactivex.rxjava2:rxandroid:2.1.1"  
implementation 'com.yanzhenjie:kalle:0.1.7'  
implementation 'com.google.zxing:core:3.3.3'  
implementation 'me.dkzwm.widget.fet:core:0.0.8'  
implementation 'me.dkzwm.widget.srl:core:1.6.6.8'  
implementation 'me.dkzwm.widget.srl:ext-material:1.6.6.8'
```

The screenshot shows the Android Studio interface with the project tree on the left and the build.gradle file on the right.

**Project Tree:**

- app
- build
- libs
- WbCloudFaceReflectLiveSdk.aar
- WbCloudNormal.aar
- src
- main
- java
- com.yuantel.kamenglib.demc
- Application
- MainActivity
- res
- drawable
- drawable-v24
- layout
- mipmap-anydpi-v26
- mipmap-hdpi
- mipmap-mdpi
- mipmap-xhdpi
- mipmap-xxhdpi
- mipmap-xxxhdpi
- values

**build.gradle (partial content):**

```
repositories {  
    flatDir {  
        dirs('libs')  
    }  
}  
  
dependencies {  
    implementation(name: 'KMSdk', ext: 'aar')  
    implementation(name: 'AlipaySdk', ext: 'aar')  
    implementation(name: 'WbCloudFaceReflectLiveSdk', ext: 'aar')  
    implementation(name: 'WbCloudNormal', ext: 'aar')  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support:recyclerview-v7:28.0.0'  
    implementation "io.reactivex.rxjava2:rxjava:2.2.12"  
    implementation "io.reactivex.rxjava2:rxandroid:2.1.1"  
    implementation 'com.yanzhenjie:kalle:0.1.7'  
    implementation 'com.google.zxing:core:3.3.3'  
    implementation 'me.dkzwm.widget.fet:core:0.0.8'  
    implementation 'me.dkzwm.widget.srl:core:1.6.6.8'  
    implementation 'me.dkzwm.widget.srl:ext-material:1.6.6.8'  
}
```

## 2.2. 快速入门

### 2.2.1. 初始化

```
//创建 SDK 实例  
final KMLiteSdk kmSdk = KMLiteSdkFactory.create();  
  
//注册相关资源  
//userId 为调用方 userId  
//userPhone 为调用方 user 对应手机号码  
//appId 和 appKey 为平台分配的资源  
kmSdk.register(context, userId, userPhone, appId, appKey);
```

### 2.2.2. 示例

```
new Thread() {  
    @Override  
    public void run() {  
        //构建 SDK  
        final KMLiteSdk kmSdk = KMLiteSdkFactory.create();  
        //注册相关资源  
        //userId 为调用方 userId  
        //userPhone 为调用方 user 对应手机号码  
        //appId 和 appKey 为平台分配的资源  
        kmSdk.register(context, userId, userPhone, appId, appKey);  
        //调起 SDK 界面  
        kmSdk.launch(context, new OnSdkUIFinishedListener() {  
            @Override
```

```
    public void onFinishned() {
        //从 SDK 返回到 APP，可选择直接解除注册
        kmSdk.unregister(context);
    }
};

}.start();
```

## 2.3. 异常处理

SDK 可能抛出`InitializationFailedException`异常（初始化异常），导致的原因可能有  
1、`Context`（上下文）`、`userId`、`userPhone`、`appId`、`appKey`为空。  
2、未进行资源注册就调用`launch`方法。

# 3. IOS 接入

## 3.1. 安装

### 3.1.1. 环境准备

Mac OS 10.12 以上，Xcode8 以上，最低支持 iOS10

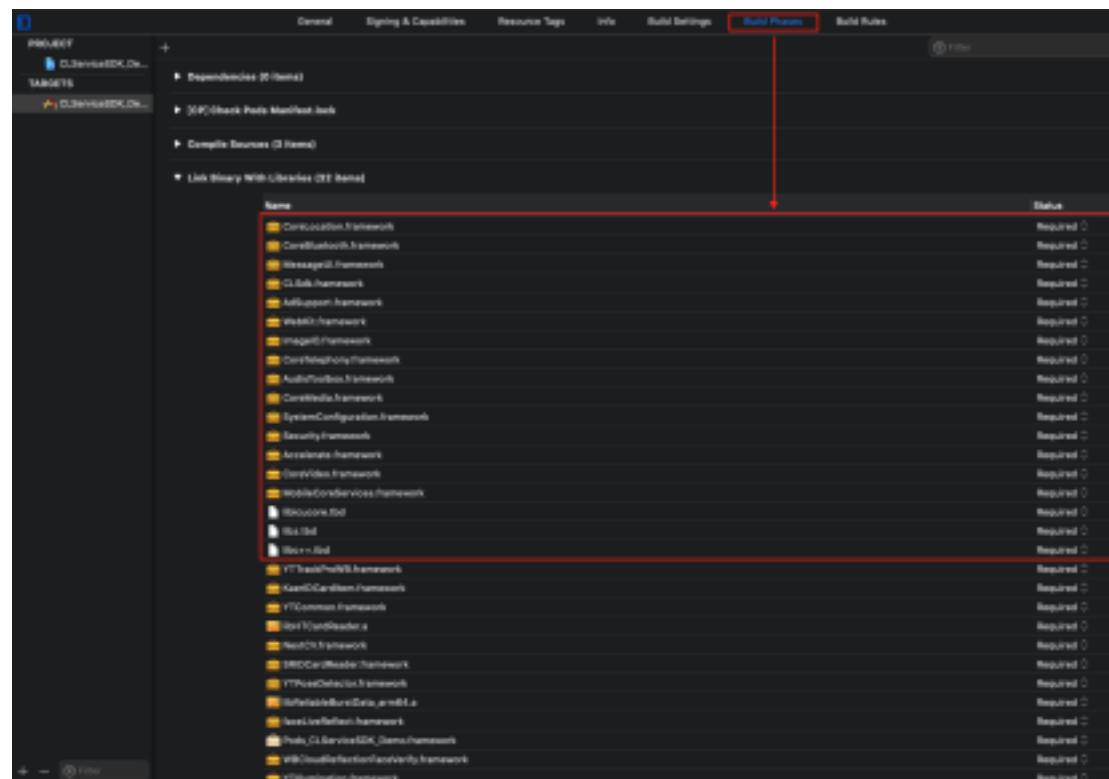
### 3.1.2. 安装 SDK

导入卡盟 iOS SDK 相关文件，并勾选 Copy Items if needed 和 Create groups，包括：

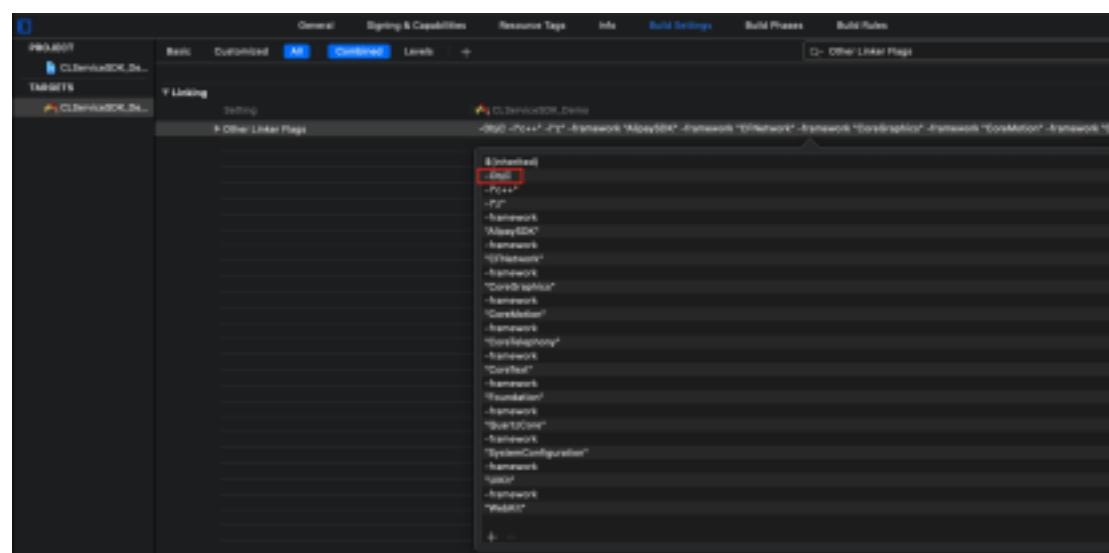
- 1) 卡盟业务相关： CLSdk.framework 、 CLSdk.bundle 、 CLDeviceService.framework 、 CLDeviceService.bundle;
- 2) 腾讯人脸识别 SDK : faceLiveReflect.framework 、 NextCV.framework 、 rapidnet\_ios.framework 、 WBCloudReflectionFaceVerify.framework 、 YTCommon.framework 、 YTILumination.framework 、 YTPoseDetector.framework 、 YTTrackProWB.framework 、 detector\_wb.bundle 、 PEDat\_wb.bundle 、 ufa\_wb.bundle 、 WBCloudReflectionFaceVerify.bundle;

- 3) 支付宝 SDK: AlipaySDK.framework、AlipaySDK.bundle; （接入方可自行按照支付宝开发文档集成，我方不提供支付宝 SDK 相关文件）
- 4) 蓝牙设备方提供的 SDK:
  - ① HT 设备: libHTCardReader.a;
  - ② 卡尔设备: KaerIDCardItem.framework;
  - ③ 森瑞设备: SRIDCardReader.framework、libReliableBurstData\_arm64.a;

添加依赖库:



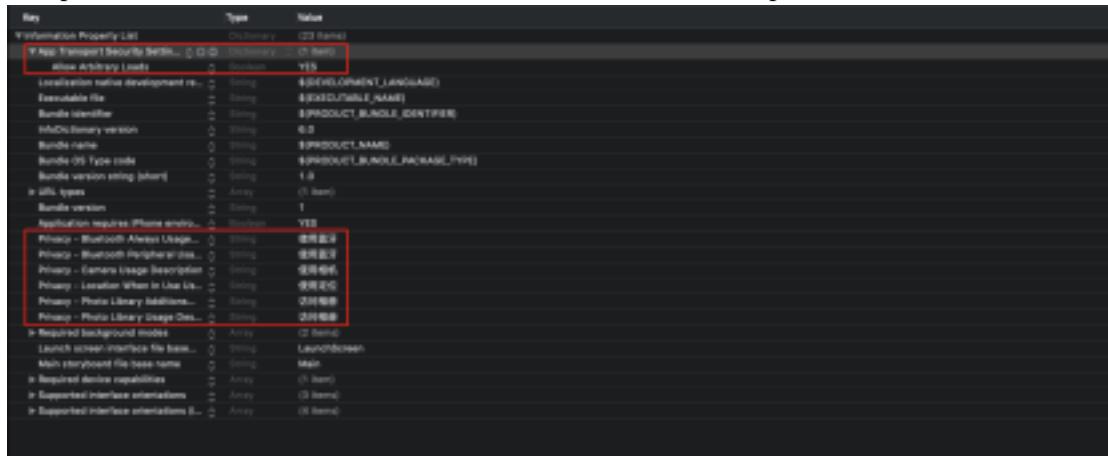
编译设置添加”-Objc”:



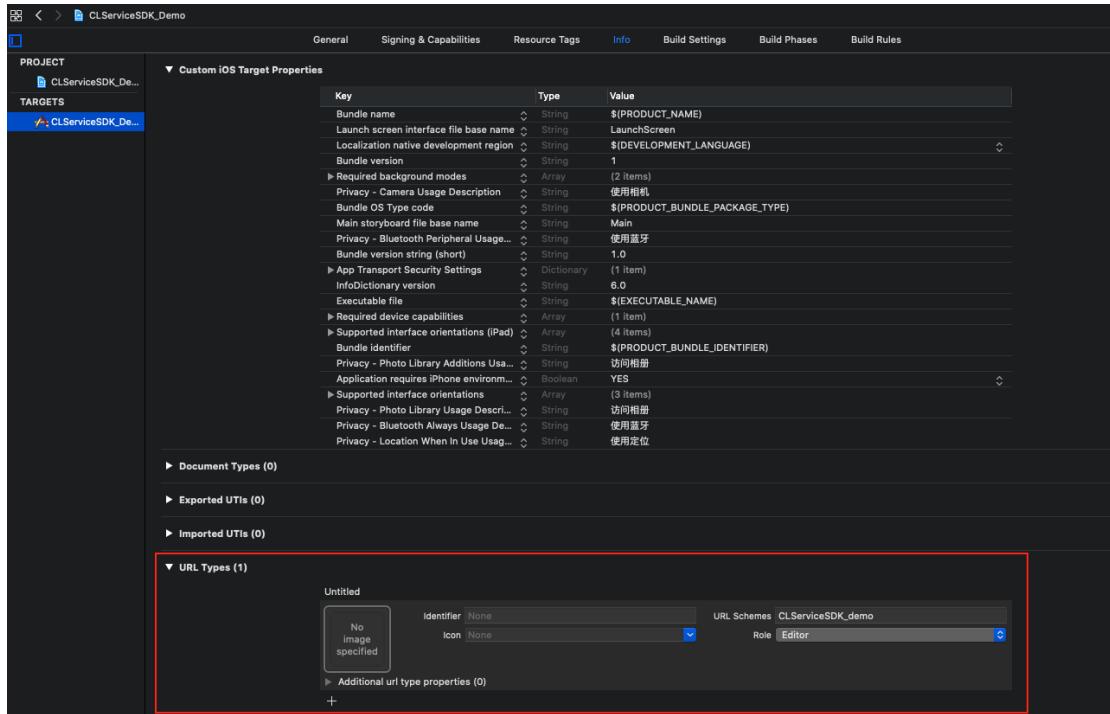
添加后台模式:



Info.plist 添加蓝牙、定位、相机、相册的访问权限，并允许 http 协议:



URL types 中设置 URL Schemes（按照支付宝开发文档的要求设置即可）：



## 3.2. 快速入门

### 3.2.1. 初始化

(一) 注册卡盟 SDK 服务:

```
// 这个appScheme为集成支付宝SDK时在URL-Types设置的URL
Schemes, 如果填写的不一致, 将导致支付完成后无法返回接入方的app
[CLService registerWithAppId:@"your app id"
    appKey:@"your app key"
    appScheme:@"your appScheme"];
```

(二) 调起卡盟 SDK:

```
// userId 接入方平台的用户id
// userPhone 接入方平台的用户手机号
[CLService startWithUserId:@"your user id"
    userPhone:@"your user phone number"];
```

(三) 处理支付宝支付结果:

```
- (BOOL)application:(UIApplication *)app openURL:(NSURL *)url
options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> *)options {
    if ([[url.host isEqualToString:@"safepay"]]) {
        // 支付跳转支付宝钱包进行支付，处理支付结果
        [[AlipaySDK defaultService] processOrderWithPaymentResult:url
            standbyCallback:^(NSDictionary *resultDic) {
                /*
                 * 接入方自己的处理
                 */
                // 将resultDic透传给卡盟SDK
                [CLService processAlipayResult:resultDic];
            }];
    }
}
```

### 3.2.2. 示例

[https://github.com/JiangHanfeng/CLServiceSDK\\_Demo](https://github.com/JiangHanfeng/CLServiceSDK_Demo)

## 3.3. 编译打包

Xcode - Product - Archive

# 4. 服务器接入

## 4.1. 数据鉴权

第三方与远特服务器之间的交互，均采用 HTTP 协议，

HTTP 方法：POST；

Content-type：application/json;charset=UTF-8；

接口均需要带上鉴权参数 sign，否则视为非法请求。

sign 参数生成方式：

第一步：

设所有发送或者接收到的数据为集合 M，将集合 M 内非空参数值的参数按照参数名 ASCII 码从小到大排序（字典序），使用 URL 键值对的格式（即 key1=value1&key2=value2…）拼接成字符串 stringA。

特别注意以下重要规则：

- ◆ 参数名 ASCII 码从小到大排序（字典序）；
- ◆ 如果参数的值为空不参与签名；

- ◆ 参数名区分大小写；

第二步：

在 stringA 最后拼接上 appKey 得到 stringSignTemp 字符串；

第三步：

对 stringSignTemp 进行 MD5 大写加密，得到的值即为 sign 的值。

eg: 参数如下：

appId: aabbcc

nonceStr: 123321

orderId: xxxxxx

第一步：对参数按照 key=value 的格式，并按照参数名 ASCII 字典序排序如下：

stringA="appid=aabbcc&nonceStr=123321&orderId=xxxxxx";

第二步：拼接 API 密钥：

stringSignTemp = appId=aabbcc&nonceStr=123321&orderId=xxxxxx&key=yourkey

第三步：MD5 大写

sign = "F7145F2C278718507452D1F95D94932A"

key 参数值请开通服务后申请

## 4.2. 接口说明

接口方法	接口作用描述
/ums/s/user/saveUserData	加盟报备
/ums/s/user/sdkAppAddJobNumSuccess	子工号报备
/ums/s/user/sdkAppRemoveJobNumSuccess	删除子工号
/ums/s/user/updatePhoneSync	修改手机号码
/bes/s/notify/merchant/exit	退加盟通知
/bes/s/query/appOpencard	开卡订单查询
/bes/s/query/makeAdultOrder	制卡订单查询
通知类 url 需要将 url 告知远特配置数据 (只支持 http)	用于接口回调数据通知

## 4.3. 主动请求类接口

### 4.3.1. 请求说明

远特给第三方提供多个 api 接口，接口返回均为 json 格式字符串，主要有两种使用场景：

1. 第三方向远特同步数据；

2. 第三方请求远特查询数据。

设定远特域名为: YUANTE\_URL

字段类型: str(字符串类型), int(数字类型)

注意: 以下接口中, 请求示例中 sign 只是示例数据, 并不是按实际规则生成的

### 4.3.2. 加盟报备

接口说明: 商户审核通过后, 同步用户信息

请求地址: /ums/s/user/saveUserData

请求参数:

```
{  
    "phone": "电话号码",  
    "userId": "userId",  
    "userName": "姓名",  
    "appid": "appid",  
    "sign": "签名",  
    "nonceStr": "32 位随机字符串",  
    "merchantType": "商户类型, 1 企业, 2 个人",  
    "merchantName": "商户名称 (个人商户有值, 企业商户时为空字符串)",  
    "contactName": "联系人姓名 (企业商户有值, 个人商户时为空字符串)",  
    "buzCode": "营业执照号或身份证号",  
    "companyName": "门店名称 (企业商户有值, 个人商户时为空字符串)"  
}
```

请求参数说明:

参数	必填	参数类型	说明
phone	是	String	用户 11 位手机号码
userId	是	String	用户生成的用户 ID
userName	是	String	用户姓名
appid	是	String	平台分配的 id
sign	是	String	
nonceStr	是	String	32 位随机字符串
merchantType	是	String	商户类型, 1 企业, 2 个人
merchantName	是	String	商户名称 (个人商户有值, 当类型为企业商户时 为空字符串)
contactName	是	String	联系人姓名 (企业商户有值, 当类型为个人商户 时为空字符串)
buzCode	是	String	营业执照号或身份证号

companyName	是	String	门店名称（企业商户有值，当类型为个人商户时为空字符串）
-------------	---	--------	-----------------------------

返回参数：

```
{
  "code": "200",
  "msg": "success"
}
```

参数说明：

code		响应码。响应码说明参加文档末尾处的响应码说明
msg		响应消息

### 4.3.3. 子工号报备

接口说明：添加子工号成功，同步数据

请求地址：/ums/s/user/sdkAppAddJobNumSuccess

请求参数：

```
{
  "createTime": "创建时间",
  "userId": "子工号 id",
  "phone": "子工号电话号码",
  "xnMainUserId": "SDK 侧主工号 userId",
  "cityCode": "城市编码",
  "cityName": "城市名称",
  "merchantType": "商户类型，1企业，2个人",
  "appid": "appid",
  "userName": "子工号身份证姓名",
  "idCardAddress": "子工号身份证地址",
  "idCardPeriod": "子工号身份证有效期",
  "sign": "签名",
  "nonceStr": "32位随机字符串",
}
```

请求参数说明：

参数	必填	参数类型	说明
createTime	是	Long	创建时间 13位时间戳
userId	是	String	子工号 id
phone	是	String	子工号电话号码
xnMainUserId	是	String	SDK 侧主工号 userId

cityCode	是	String	城市编码
cityName	是	String	城市名称
merchantType	是	int	商户类型, 1 企业, 2 个人
appid	是	String	appid
userName	是	String	子工号身份证姓名
idCardAddress	是	String	子工号身份证地址
idCardPeriod	是	String	子工号身份证有效期
sign	是	String	
nonceStr	是	String	32 位随机字符串

返回参数:

```
{
  "code": "200",
  "msg": "success"
}
```

参数说明:

msg		响应码。响应码说明参见文档末尾处的响应码说明
code		响应消息。参见文档末尾处的响应消息说明

#### 4.3.4. 删除子工号

请求地址: /ums/s/user/updatePhoneSync

请求参数:

```
{
  "removeUserId": "被删账号的 userid",
  "appid": "appid",
  "sign": "签名",
  "nonceStr": "32 位随机字符串",
  "appid": "appid"
}
```

请求参数说明:

参数	必填	参数类型	说明
removeUserId	是	String	被删除账号的 userid
sign	是	String	签名
nonceStr	是	String	32 位随机字符串

appid	是	String	appid
-------	---	--------	-------

返回参数：

```
{
  "code": "200",
  "msg": "success"
}
```

参数说明：

msg		响应码。响应码说明参加文档末尾处的响应码说明
code		响应消息。参见文档末尾处的响应消息说明

### 4.3.5. 修改手机号成功

请求地址： /ums/s/user/updatePhoneSync

请求参数：

```
{
  "newPhone": "SDK 侧被删账号的 userid",
  "userId": "appid",
  "sign": "签名",
  "nonceStr": "32 位随机字符传",
  "appid": "32 位随机字符传"
}
```

请求参数说明：

参数	必填	参数类型	说明
newPhone	是	String	更新后的手机号
sign	是	String	签名
nonceStr	是	String	32 位随机字符串
Appid	是	String	Appid
userId	是	String	SDK 侧 userid

返回参数：

```
{
  "code": "200",
  "msg": "success"
}
```

参数说明：

msg		响应码。响应码说明参加文档末尾处的响应码说明
code		响应消息。参见文档末尾处的响应消息说明

#### 4.3.6. 退加盟

- 使用场景：已经在远特报备过的商户在第三方要退加盟，第三方同步该商户信息给远特，远特将该商户注销。
- 请求 URL: /bes/s/notify/merchant/exit
- 请求类型: POST, application/json
- 请求参数

参数名称	必填	长度	类型	说明
appId	是		str	远特分配给第三方的 appId
nonceStr	是	32	str	随机字符串
sign	是	32	str	签名
sdkUserId	是		str	第三方的用户 id

请求示例：

```
{  
    "appId": "xxx",  
    "nonceStr": "12454545452121213568565451ssffgs",  
    "sdkUserId": "XU1810xxx",  
    "sign": "69F223073DD722C10379C70302BE6E62"  
}
```

- 返回参数：

参数名称	必填	长度	说明
code	是		详情参考状态码描述
message	是		结果说明

#### 4.3.7. 开空卡订单查询

- 使用场景：第三方查询开空卡订单相关信息。
- 请求 url: /bes/s/query/appOpencard
- 请求类型: POST, application/json
- 请求参数：

参数名称	必填	长度	类型	说明
appId	是		str	远特分配给第三方的 appId
nonceStr	是	32	str	随机字符串
sign	是	32	str	签名
sdkUserId	是		str	第三方的用户 id
sysOrderId	否		str	开卡订单号

dealerId	否		str	渠道 id
phoneNum	否		str	开卡号码
startTime	否		int	开始时间,13位时间戳
endTime	否		int	结束时间,13位时间戳
page	是		int	查询页码, 首页传 1
pageSize	是		int	查询数据条数

注意: sysOrderId、phoneNum、sdkUserId、dealerId 不能同时为空

请求示例:

```
{
    "appId": "sdk",
    "dealerId": "A100315169",
    "endTime": 1510000000000,//13位时间戳
    "nonceStr": "xxxxxxxx",
    "page": 1,
    "pageSize": 10,
    "phoneNum": "17000000000", //11位手机号码
    "sdkUserId": "X181010205224834105",
    "sign": "69F223073DD722C10379C70302BE6E62",
    "startTime": 1500000000000,//13位时间戳
    "sysOrderId": "xxx"
}
```

5. 返回参数:

参数名称	上级参数	说明
code		详情参考状态码描述
msg		结果说明
data		
orderSize	data	满足条件订单总数
orderList	data	订单数据组
sysOrderId	orderList	订单号
...	orderList	返回参数参考开卡订单通知

返回示例:

```
{
    "code": "200",
    "msg": "请求成功",
    "data": {
        "orderList": [
            {
                "activityId": 1000,
                "actualCardMoney": 0,
                "actualMoney": 0,
                "actualPrestoreMoney": 0,
                "appType": 10,
                "beforePrestore": 30000,
                "bigNumberLevel": 0,
                ...
            }
        ]
    }
}
```

```
        "bizType": 4,
        "cardMoney": 0,
        "cardType": 1,
        "changePacks": 0,
        "companyName": "张三公司",
        "createTime": 1544526126823,
        "dealerId": "A100316156",
        "dealerIdMake": "A100316156",
        "deductionMoney": 0,
        "detailType": 2,
        "firstCharge": 0,
        "firstChargeActual": 0,
        "home": "银川",
        "iccid": "xxxx01188410440xxxxx",
        "initialMoney": 0,
        "insuranceIncome": 0,
        "makeDetailType": 2,
        "modifyTime": 1544526433890,
        "monopolyType": 0,
        "optionalPackage": "1352755|1353272",
        "orderStatus": 2,
        "packageId": 1353295,
        "papersCode": "511xxxxxxxxxxxxxxX",
        "payType": 2,
        "payUser": 0,
        "phoneActCode": 1001,
        "phoneNumber": 17045921019,
        "prestoreDeduct": 0,
        "prestoreMoney": 30000,
        "privilegeType": 1,
        "sysActivationOrderId": "CK18120215072503593",
        "sysOrderId": "K18121119020660944",
        "terminalType": 1,
        "totalMoney": 30000,
        "transactionIdPay": 20121215,
        "userId": "X181122180353315365",
        "userIdMake": "X181122180353315365",
        "userName": "张三"
    }
],
"orderSize": 1
}
}
```

#### 4.3.8. 激活卡订单查询

6. 使用场景：第三方查询激活卡订单相关信息。

7. 请求 url: /bes/s/query/appActivecard

8. 请求类型: POST, application/json

9. 请求参数:

参数名称	必填	长度	类型	说明
appId	是		str	远特分配给第三方的 appId
nonceStr	是	32	str	随机字符串
sign	是	32	str	签名
sdkUserId	是		str	第三方的用户 id
sysOrderId	否		str	开卡订单号
dealerId	否		str	渠道 id
phoneNum	否		str	开卡号码
startTime	否		int	开始时间,13位时间戳
endTime	否		int	结束时间,13位时间戳
page	是		int	查询页码, 首页传 1
pageSize	是		int	查询数据条数

注意: sysOrderId、phoneNum、sdkUserId、dealerId 不能同时为空

请求示例:

```
{  
    "appId": "sdk",  
    "dealerId": "A100315169",  
    "endTime": 1510000000000, //13位时间戳  
    "nonceStr": "xxxxxxxx",  
    "page": 1,  
    "pageSize": 10,  
    "phoneNum": "17000000000", //11位手机号码  
    "sdkUserId": "X181010205224834105",  
    "sign": "69F223073DD722C10379C70302BE6E62",  
    "startTime": 1500000000000, //13位时间戳  
    "sysOrderId": "xxx"  
}
```

10. 返回参数:

参数名称	上级参数	说明
code		详情参考状态码描述
msg		结果说明
data		
orderSize	data	满足条件订单总数
orderList	data	订单数据组
sysOrderId	orderList	订单号
...	orderList	返回参数参考开卡订单通知

返回示例:

```
{  
    "code": "200",  
    "msg": "请求成功",  
    "data": {  
        "orderList": [  
            {  
                "activityId": 1000,  
                "actualCardMoney": 0,  
                "actualMoney": 0,  
                "actualPrestoreMoney": 0,  
                "appType": 10,  
                "beforePrestore": 30000,  
                "bigNumberLevel": 0,  
                "bizType": 4,  
                "cardMoney": 0,  
                "cardType": 1,  
                "changePacks": 0,  
                "companyName": "张三公司",  
                "createTime": 1544526126823,  
                "dealerId": "A100316156",  
                "dealerIdMake": "A100316156",  
                "deductionMoney": 0,  
                "detailType": 2,  
                "firstCharge": 0,  
                "firstChargeActual": 0,  
                "home": "银川",  
                "iccid": "xxxx01188410440xxxxx",  
                "initialMoney": 0,  
                "insuranceIncome": 0,  
                "makeDetailType": 2,  
                "modifyTime": 1544526433890,  
                "monopolyType": 0,  
                "optionalPackage": "1352755|1353272",  
                "orderStatus": 2,  
                "packageId": 1353295,  
                "papersCode": "511xxxxxxxxxxxxxxX",  
                "payType": 2,  
                "payUser": 0,  
                "phoneActCode": 1001,  
                "phoneNumber": 17045921019,  
                "prestoreDeduct": 0,  
                "prestoreMoney": 30000,  
                "privilegeType": 1,  
                "sysActivationOrderId": "CK18120215072503593",  
            }  
        ]  
    }  
}
```

```

        "sysOrderId": "K18121119020660944",
        "terminalType": 1,
        "totalMoney": 30000,
        "transactionIdPay": 20121215,
        "userId": "X181122180353315365",
        "userIdMake": "X181122180353315365",
        "userName": "张三"
    }
],
"orderSize": 1
}
}

```

#### 4.3.9. 成卡订单查询

1. 使用场景：第三方查询成卡订单相关信息。
2. 请求 url: /bes/s/query/makeAdultOrder
3. 请求类型: POST, application/json
4. 请求参数:

参数名称	必填	长度	类型	说明
appId	是		str	远特分配给第三方的 appId
nonceStr	是	32	str	随机字符串
sign	是	32	str	签名
sdkUserId	是		str	第三方的用户 id
sysOrderId	否		str	订单号
dealerId	否		str	渠道 id
phoneNum	否		str	制卡号码
startTime	否		int	开始时间,13 位时间戳
endTime	否		int	结束时间,13 位时间戳
page	是		int	查询页码, 首页传 1
pageSize	是		int	查询数据条数

注意: sysOrderId、phoneNum、sdkUserId、dealerId 不能同时为空

请求示例:

```
{
    "appId": "sdk",
    "dealerId": "A100315169",
    "endTime": 1510000000000,//13 位时间戳
    "nonceStr": "xxx",
    "page": 1,
    "pageSize": 10,
```

```

"phoneNum": "170000000000",//11位手机号码
"sdkUserId": "X181010205224834105",
"sign": "69F223073DD722C10379C70302BE6E62",
"startTime": 1500000000000000, //13位时间戳
"sysOrderId": "K20041017025100022"
}

```

5. 返回参数:

参数名称	上级参数	说明
code		详情参考状态码描述
msg		结果说明
data		
orderSize	data	满足条件订单总数
orderList	data	订单数据组
sysOrderId	orderList	订单号
...	orderList	返回参数参考制卡订单通知

返回示例:

```
{
  "code": "200",
  "msg": "请求成功",
  "data": {
    "orderList": [
      {
        "actualCardMoney": 0,
        "actualMoney": 30000,
        "actualPrestoreMoney": 30000,
        "appType": 9,
        "beforePrestore": 0,
        "bigNumberLevel": 0,
        "cardMoney": 0,
        "changePacks": 0,
        "companyName": "张三公司",
        "createTime": 1583895035195,
        "dealerId": "A100379687",
        "deductionMoney": 0,
        "detailType": 2,
        "deviceId": "CK20031110503500037",
        "home": "石家庄",
        "iccid": "xxxx01188410440xxxxx",
        "idcardAddress": "成都市武侯区***",
        "idcardName": "李*",
        "idcardNum": "511325****112X",
        "idcardPeriod": "2017.11.11-2037.11.11",
        "initialMoney": 0,
        "insuranceIncome": 0,
        ...
      }
    ]
  }
}
```

```

        "makeEndTime": 1583895947091,
        "modifyTime": 0,
        "monopolyType": 0,
        "optionalPackage": "1353305|1353317",
        "orderStatus": 2,
        "packageId": 1353295,
        "payType": 1,
        "phoneActCode": 1001,
        "phoneNumber": 17041630717,
        "prestoreDeduct": 0,
        "prestoreMoney": 30000,
        "privilegeType": 2,
        "safePhone": 17092510000,
        "sysOrderId": "CK20031110503500037",
        "terminalType": 1,
        "totalMoney": 30000,
        "transactionIdPay": 20031110503500036,
        "userId": "XU181010185742505342",
        "username": "张三"
    },
],
"orderSize": 1
}
}

```

#### 4.3.10. 查询用户信息

1、此请求可以通过 userId 向卡盟平台查询商户(主工号、子工号)信息，商户审核有结果后才返回信息。

2、请求地址：/bes/s/query/userInfoQuery

3、请求参数：

```
{
    "sdkUserId": "单点结算侧 userId",
    "sign": "签名",
    "nonceStr": "32 位随机字符传",
    "appid": "32 位随机字符传"

}
```

请求参数说明：

参数	必填	参数类型	说明
----	----	------	----

sdkUserId	是	String	单点结算侧 userId
-----------	---	--------	--------------

4、返回参数：

```
{
    "code": "200",
    "msg": "success",
    "data": {
        "message": "{\"businessLicence\":\"365255555288\",
                     \"period\":\"20150225-20250225\",
                     \"address\":\"湖北省武汉市 xxx\",
                     \"identityCardName\":\"xxx\",
                     \"companyName\":\"四川 xxxx 公司\",
                     \"identityCard\":\"420702xxx7862\",
                     \"merchantType\":1,
                     \"userId\":\"xxx\",
                     \"identityCardAddress\":\"湖北省鄂州市 xxx\"}",
        "syncResult": "同步数据发送成功"
    }
}
```

参数说明：

msg		响应码。响应码说明参加文档末尾处的响应码说明
code		响应消息。参见文档末尾处的响应消息说明
data		返回数据
businessLicence	String	商户营业执照号，个人=证件号码，企业=营业执照，子工号=主工号商户信息
period	String	身份证有效期（企业存使用人信息，子工号存证件有效期）
address	String	商户门店地址,个人=证件地址，企业=门店地址，子工号=主工号商户信息
identityCardName	String	身份证姓名（企业存使用人信息，子工号存证件姓名）
companyName	String	网点或公司名称：个人=证件项目，企业=公司名称，子工号=主工号商户信息
identityCard	String	身份证号码（企业存使用人信息，子工号存证件号码）
merchantType	String	商户属性 1 企业 2 个人
userId	String	用户 userid
identityCardAddress	String	身份证地址（企业存使用人信息，子工号存证件地址）
syncResult	String	同步数据发送结果描述

## 4.4. 通知类接口

### 4.4.1. 通知策略

待服务开通后,第三方提供给远特一个接收数据的 URL 地址, 远特把所有通知数据均发向该地址, 数据格式为 JSON 字符格式, 该请求 body 中有 type 字段标明通知业务类型。

第三方收到远特通知后, 返回 json 格式字符串, 原特方根据返回 json 中参数 code 的值判断第三方是否处理成功。当且仅当 json 中 code 为 200 时, 远特认为第三方处理成功, 否则远特将启动重发机制(每次失败后间隔 10~60s 重发请求, 失败 5 次后不再重发请求)。

第三方返回示例: {"code":"200"}

注意: 鉴权方式参见 数据鉴权

### 4.4.2. 通知格式

公共参数, 远特每个通知中都含有一下参数

参数名称	必填	长度	说明
appId	是		远特分配给第三方的 appId
nonceStr	是	32	随机字符串
sign	是	32	签名
type	是		通知类型, eg: CARD_ACTIVE(卡激活)
uniqueId	是		唯一标识
data	是		业务相关参数, type 不同, data 内数据不同

#### 4.4.2.1 通知数据

通知接口 type、data 说明

##### type: REGISTER\_ORDER\_SUCCESS

说明: 商户审核(包括主工号与子工号)通过成为远特商户时触发此调用。

参数	参数类型	说明
appid	String	Appid
sign	String	签名

nonceStr	String	32 位随机字符串
type	String	操作类型: REGISTER_ORDER_SUCCESS
uniqueId	String	唯一码
data	JSONObject	同步数据
address	String	商户门店地址
companyName	String	网点或公司名称
businessLicence	String	商户营业执照号
identityCard	String	身份证号码 (企业存使用人信息)
identityCardName	String	身份证姓名 (企业存使用人信息)
identityCardAddress	String	身份证地址 (企业存使用人信息)
period	String	身份证有效期 (企业存使用人信息)
userId	String	单点结算侧 userid
merchantType	Int	商户属性 1 企业 2 个人
isMain	Int	1 主账号 0 子账号

示例:

```
{
    "appid": "appid",
    "nonceStr": "32 位随机字符串",
    "sign": "签名",
    "type": "REGISTER_ORDER_SUCCESS",
    "uniqueId": "唯一码",
    "data": {
        "address":"",
        "companyName":"",
        "businessLicence":"",
        "identityCard":"",
        "identityCardName":"",
        "identityCardAddress":"",
        "period":"",
        "userId":"",
        "merchantType":""
    }
}
```

## type: APP\_MAKEADULT\_SUCCESS

说明: 制成功一张成卡

data:

sysOrderId	String	成卡订单号
------------	--------	-------

transactionIdPay	String	第三方支付订单号
phoneNumber	String	号码
createTime	Long	订单创建时间 13位时间戳
home	String	号码城市
prestoreMoney	Long	预存面额, 分
actualPrestoreMoney	Long	实际支付预存, 分
cardMoney	Long	号码占用费, 分
actualCardMoney	Long	实际支付号码占用费, 分
totalMoney	Long	总面额, 分
actualMoney	Long	实际支付总额, 分
orderStatus	Integer	订单状态 2成功
payType	Integer	支付方式 1:远特账户支付 2:微信支付 3: 支付宝支付
modifyTime	String	订单最新时间, 13位时间戳
userId	String	sdk 方用户 id
dealerId	String	商户 id
packageId	String	套餐标识
optionalPackage	String	可选包标识, 分割
iccid	String	卡 iccid
terminalType	Integer	终端类型。1: ios, 2: android 0: 未知类型
bigNumberLevel	Integer	号码大等级 0为普号 1-9为对应等级靓号 如1级靓号
privilegeType	Integer	优惠类型, 1: 优惠模式, 2: 反佣模式, 3: 混合模式
deductionMoney	Integer	使用抵扣总金额, 分
prestoreDeduct	Integer	抵扣预存, 分
username	String	操作者姓名
companyName	String	商户姓名
monopolyType	Integer	0普号 1大众专营号 2专属专营号
deviceId	String	写卡设备号
appType	Integer	9app 11制卡中心
changePacks	Integer	1专营号升空卡套餐变更, 0未变更
beforePrestore	Integer	号码默认预存, 分, 只有专营号类别有值
safePhone	String	短信校验号码
phoneActCode	Integer	选号费活动 code
initialMoney	Long	号码占用费原价, 分
makeEndTime	Long	制卡结束时间
detailType	Integer	制卡方 detailType
idcardName	String	证件姓名
idcardNum	String	证件号
idcardAddress	String	证件地址
idcardPeriod	String	证件有效期
insuranceIncome	Long	保险返佣, 分

示例：

{

```
"appId": "sdk",
"nonceStr": "gALTvBOMAhRCfFEHbVawgiEwH5NFHZsJ",
"sign": "0D434FF90A88411CC1A64AA25135B758",
"type": "APP_MAKEADULT_SUCCESS",
"uniqueId": "CK20051415040300004",
"data": {
    "insuranceIncome": 0,
    "companyName": "张三",
    "idcardPeriod": "2017.11.11-2037.11.11",
    "orderStatus": 2,
    "initialMoney": 17600,
    "prestoreDeduct": 0,
    "idcardName": "张**",
    "deviceId": "",
    "phoneActCode": 1001,
    "actualCardMoney": 15800,
    "detailType": 5,
    "idcardNum": "511325****112X",
    "iccid": "89860118841044071337",
    "modifyTime": 1589439983090,
    "payType": 1,
    "appType": 9,
    "transactionIdPay": "",
    "actualPrestoreMoney": 30000,
    "safePhone": "17090060002",
    "prestoreMoney": 30000,
    "actualMoney": 45800,
    "dealerId": "A7jkU9B7Ec5",
    "deductionMoney": 0,
    "packageId": "1353356",
    "totalMoney": 45800,
    "bigNumberLevel": 6,
    "makeEndTime": 1589439983090,
    "changePacks": 0,
    "userId": "YU200514110912867224",
    "home": "上海",
    "idcardAddress": "成都市武侯区***",
    "optionalPackage": "1353254|1353253",
    "terminalType": 1,
    "monopolyType": 0,
    "phoneNumber": "17191503335",
    "cardMoney": 15800,
    "createTime": 1589439843355,
    "beforePrestore": 0,
```

```

    "privilegeType": 2,
    "sysOrderId": "CK20051415040300004",
    "username": "张三"
}
}

```

## **type: APP\_OPENCARD\_SUCCESS**

说明: app 开卡成功,包含开空卡、激活成卡

data:

sysOrderId	String	开卡订单号
transactionIdPay	String	第三方支付订单号
phoneNumber	String	号码
beforePrestore	Integer	卡默认预存, 分; 如果是成卡, 值为成卡预存, 如果是空卡, 值为号码默认预存
createTime	Long	订单创建时间 13 位时间戳
home	String	号码城市
prestoreMoney	Long	预存面额, 分
actualPrestoreMoney	Long	实际支付预存, 分*100
cardMoney	Long	号码占用费, 分
actualCardMoney	Long	实际支付号码占用费, 分*100
firstCharge	Long	首充预存面额, 分
firstChargeActual	Long	实际支付首充金额, 分*100
totalMoney	Long	总面额, 分
actualMoney	Long	实际支付总额, 分*100
orderStatus	Integer	订单状态 2 成功
modifyTime	String	订单最新时间, 13 位时间戳
userId	String	sdk 方用户 id
dealerId	String	商户 id
packageId	String	套餐标识
optionalPackage	String	可选包标识, 分割
iccid	String	卡 iccid
activityId	String	活动 id
payType	Integer	支付方式 1:远特账户支付 2:微信支付 3: 支付宝支付
terminalType	Integer	终端类型。1: ios, 2: android 0: 未知类型
bigNumberLevel	Integer	号码大等级 0 为普号 1-9 为对应等级靓号 如 1 级靓号
privilegeType	Integer	优惠类型, 1: 优惠模式, 2: 返佣模式, 3: 混合模式
userName	String	身份证姓名, 已脱敏
papersCode	String	证件号码,已脱敏
payUser	Integer	0 商家支付 1 用户扫码支付

deductionMoney	Integer	使用抵扣总金额, 分
prestoreDeduct	Integer	抵扣预存, 分
monopolyType	Integer	0 普号 1 大众专营号 2 专属专营号
appType	Integer	1 卡盟 app5 远特通服 6 远微商城 7 信时空公众号 9 单点结算 app10 小程序
username	String	操作人姓名
companyName	String	商户名称
initialMoney	Long	号码占用费原价, 分, 有活动时才有值, 只做展示用
detailType	Integer	激活方的 detailType
bizType	Integer	操作类型 4 成卡激活 6 开空卡
sysActivationOrderId	String	激活时成卡订单号
changePacks	Integer	1 跟默认套餐有变更 0 未变更套餐
makeDetailType	Integer	制卡方的 detailType
userIdMake	String	制卡操作人 sdk 方的 userId
dealerIdMake	String	制卡操作商户 Id
userName	String	证件姓名
papersCode	String	证件号码
cardType	Integer	1 远特卡 2 联通卡
phoneActCode	Integer	选号费活动 code
insuranceIncome	Long	保险返佣, 分

示例：

```
{
    "appId": "sdk",
    "nonceStr": "UVVAaJPCerSp4Z48tREWH4ii04S9Njz",
    "sign": "392E48982E9823FE692ACB7B921BD784",
    "type": "APP_OPENCARD_SUCCESS",
    "uniqueId": "K20051415121800006",
    "data": {
        "insuranceIncome": 0,
        "payUser": 0,
        "bizType": 6,
        "dealerIdMake": "",
        "companyName": "张三",
        "orderStatus": 2,
        "initialMoney": 0,
        "prestoreDeduct": 0,
        "phoneActCode": 0,
        "actualCardMoney": 0,
        "detailType": 5,
        "activityId": "",
        "papersCode": "511325****112X",
        "iccid": "89860118841044071337",
        "modifyTime": 1589440548702,
    }
}
```

```

    "payType": 1,
    "appType": 9,
    "transactionIdPay": "",
    "actualPrestoreMoney": 2900000,
    "sysActivationOrderId": "",
    "firstChargeActual": 985000,
    "prestoreMoney": 30000,
    "actualMoney": 3885000,
    "makeDetailType": 0,
    "dealerId": "A7jkU9B7Ec5",
    "deductionMoney": 0,
    "cardType": 1,
    "packageId": "1353357",
    "totalMoney": 40000,
    "firstCharge": 10000,
    "bigNumberLevel": 0,
    "changePacks": 1,
    "userName": "张**",
    "userId": "YU200514110912867224",
    "home": "龙岩",
    "optionalPackage": "1353255|1353256",
    "terminalType": 1,
    "monopolyType": 1,
    "phoneNumber": "16712450246",
    "cardMoney": 0,
    "createTime": 1589440338350,
    "userIdMake": "",
    "beforePrestore": 1000,
    "privilegeType": 2,
    "sysOrderId": "K20051415121800006",
    "username": "张三"
}
}

```

## **type: TAS\_ACTIVE\_SUCCESS**

说明：远特卡激活

data:

sysOrderId	String	开卡订单号
payTransactionId	String	第三方支付订单号
phoneNumber	String	号码

beforePrestore	Integer	卡默认预存, 分; 如果是成卡, 值为成卡预存, 如果是空卡, 值为号码默认预存
createTime	Long	订单创建时间 13 位时间戳
phoneHome	String	号码城市
prestoreMoney	Long	预存面额, 分
actualPrestoreMoney	Long	实际支付预存, 分
cardMoney	Long	号码占用费, 分
actualCardMoney	Long	实际支付号码占用费, 分
firstCharge	Long	首充预存面额, 分
firstChargeActual	Long	实际支付首充金额, 分
totalMoney	Long	总面额, 分
actualMoney	Long	实际支付总额, 分
orderStatus	Integer	订单状态 2 成功
packageId	String	套餐标识
optionalPackage	String	可选包标识, 分割
terminalType	Integer	终端类型。1: ios, 2: android 0: 未知类型
modifyTime	String	订单最新时间, 13 位时间戳
userId	String	sdk 方用户 id
dealerId	String	商户 id
iccid	String	卡 iccid
bigNumberLevel	Integer	号码大等级 0 为普号 1-9 为对应等级靓号 如 1 级靓号
userName	String	证件姓名
papersCode	String	证件号码
monopolyType	Integer	0 普号 1 大众专营号 2 专属专营号
initialMoney	Long	号码占用费原价, 分
bizType	Integer	操作类型 4 成卡激活
sysActivationOrderId	String	激活时成卡订单号
changePacks	Integer	1 有变更套餐 0 没有变更
makeDetailType	Integer	制卡方的 detailType
userIdMake	String	制卡操作人 sdk 方的 userId
dealerIdMake	String	制卡操作商户 Id
userName	String	证件姓名
papersCode	String	证件号码
privilegeType	Integer	优惠类型, 1: 优惠模式, 2: 返佣模式, 3: 返佣模式

示例：

```
{
    "appId": "sdk",
    "nonceStr": "nFYZyHiN6wDMJjDQSiVXaA3eJ87L0HoA",
    "sign": "862309C40EB81640A678ABF271D3BA69",
    "type": "TAS_ACTIVE_SUCCESS",
    "uniqueId": "TF20051416201707695",
```

```
"data": {  
    "bizType": 4,  
    "dealerIdMake": "A7jkU9B7Ec5",  
    "phoneHome": "上海",  
    "orderStatus": 2,  
    "initialMoney": 0,  
    "payTransactionId": "",  
    "phoneActCode": 0,  
    "actualCardMoney": 0,  
    "papersCode": "511325****112X",  
    "iccid": "89860118841044071337",  
    "modifyTime": 1589444907045,  
    "payType": 0,  
    "actualPrestoreMoney": 0,  
    "sysActivationOrderId": "CK20051415040300004",  
    "firstChargeActual": 0,  
    "prestoreMoney": 30000,  
    "actualMoney": 0,  
    "makeDetailType": 5,  
    "dealerId": "A7jkU9B7Ec5",  
    "packageId": "1353356",  
    "totalMoney": 45800,  
    "firstCharge": 0,  
    "bigNumberLevel": 6,  
    "changePacks": 0,  
    "userName": "张**",  
    "userId": "Y200514110912867224",  
    "optionalPackage": "1353254|1353253",  
    "terminalType": 2,  
    "monopolyType": 0,  
    "phoneNumber": "17191503335",  
    "cardMoney": 15800,  
    "createTime": 1589444417672,  
    "userIdMake": "Y200514110912867224",  
    "beforePrestore": 30000,  
    "privilegeType": 2,  
    "sysOrderId": "TF20051416201707695"  
}  
}
```

#### **4.4.3. 返回码说明**

code	状态说明
200	请求处理成功
600	系统内部错误
607	参数错误
645	用户信息异常
725	接口请求频率过快

#### **4.4.4. 异常处理**

若未收到通知,请使用对应接口主动查询数据