



鑫云+电子钱包 SDK 使用手册

版本 V1.1

文档编号	NJCB- OPEN-DZQB-01	保密等级	机密
作者	王康、李俊康	最后修改日期	
审核人		最后审批日期	

文档修订记录

编号	章节名称	修订内容简述	修订日期	修订前版本号	修订后版本号	修订人
1						
2						
3						
4						
5						
6						
7						
8						

目 录

1、SDK 集成流程	4
1.1 IOS 端	4
1.1.1 示例代码	5
1.1.2 返回码列表	6
1.2 ANDROID 端	11
1.2.1 示例代码	13
1.2.2 返回码列表	13
2、SDK 调用说明	18
2.1 IOS 端	18
2.1.1 初始化	18
2.1.2 钱包	19
2.1.3 支付	21
2.1.4 系统版本号	24
2.1.5 当前手机 IP	24
2.1.6 设备唯一标识	25
2.1.7 bundleid	25
2.1.8 手机系统版本号	25
2.2 ANDROID 端	25
2.2.1 初始化	26
2.2.2 支付	26
2.2.3 钱包	27

1、SDK 集成流程

[查看账户明文](#)

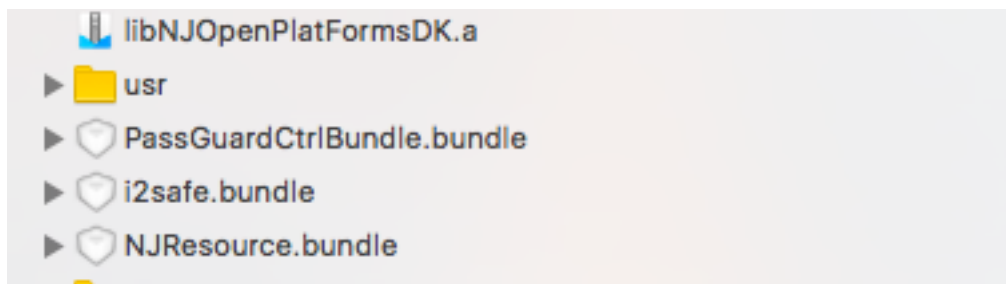
1.1 IOS 端

1. 新建工程

在 XCODE 中新建你的工程，支持的最低 iOS 版本为 iOS 8，iOS10+ 在项目设置中打开 Keychain sharing;

2. 引入南京银行鑫云+外联开放平台











将南京银行鑫云+外联开放平台 iOS SDK 中包含的 NJSdk 文件夹内的所有文件拖入到工程内



3. 添加依赖库

在 Linked Frameworks and Libraries 中引入

JavaScriptCore.framework
CoreTelephony.framework
AdSupport.framework
SystemConfiguration.framework
Security.framework
AudioToolbox.framework
CFNetwork.framework
Coremotion.framework
libstdc++.6.0.9.tbd
libstdc++.tbd
libz.tbd

Name	Status
 AdSupport.framework	Required ⬆
 SystemConfiguration.framework	Required ⬆
 CFNetwork.framework	Required ⬆
 CoreMotion.framework	Required ⬆
 libstdc++.tbd	Required ⬆
 libstdc++.6.0.9.tbd	Required ⬆
 AudioToolbox.framework	Required ⬆
 Security.framework	Required ⬆
 libz.tbd	Required ⬆
 JavaScriptCore.framework	Required ⬆
+ — Drag to reorder frameworks	

4. 出于安全性控制，我们在 key chain 中记录了设备 ID 等信息，请 XCODE8 以上的开发者打开 key chain 权限：

Keychain Sharing

5. 修改 build setting 添加 other linker flags 属性：-ObjC

▼ Linking	
Setting	NJSDK-SourceCode
Other Linker Flags	-ObjC
Quote Linker Arguments	Yes ⬆

6. 添加头文件

在需要使用南京银行鑫云+外联开放平台 SDK 的文件中#import "NJSDK.h"、#import "NJCallBackModel.h"并增加协议<NJRespDelegate>。若使用到支付，请 import 支付的 model，#import "NJPaymentInfo.h"，#import "TalkingData.h"，#import "NJCustomPresentationVc.h"。

1.1.1 示例代码

```
#import "NJCallBackModel.h"
#import "NJPaymentInfo.h"
#import "NJSDK.h"//此类为支付，钱包功能入口
#import "NJCallBackModel.h" //此类为回调类，要接受 SDK 的响应信息，要遵守协议

@interface viewController : <NJRespDelegate>@end
// 实现文件#import "viewController.h"@interface viewController ()
@implementation viewController
```

```

- (void)viewDidLoad {
    [super viewDidLoad];
    //成为代理
    NJCallBackModel *model=[NJCallBackModel defaultManager];
    model.NJdelegate=self;
}

// 实现代理方法
- (void)getRespCallBackFromNJ: (NJCallBackModel*) resp {

    if (resp.key== NJ_REQ_KEY_PAYMENT_RESULT) {
        if ([[resp.callBackResult objectForKey:NJSDK_RETURNCODE] isEqual:
NJSDK_PAYMENT_SUCCESS]) {
            ” //这里写支付成功后将要执行的代码
            NSLog(@"支付成功");
        }
    }
}

@end

```

1.1.2 返回码列表

NJSDK_RETURNCODE //returnCode 字段名

NJSDK_SUCCESS //可根据此宏定义来判断响应信息是否正确

NJSDK_PAYMENT_SUCCESS//可根据此宏来判断支付是否成功

NJSDK_NETFAILURE //网络异常

NJSDK_SERVICE_ERROR // 服务异常

NJSDK_PARAMET_ERROR //必传参数没传，可能是未初始化 SDK，或未验证开发者

NJSDK_USERCANCEL_REQUEST //用户取消操作

错误码表

- sdk 返回编码列表

returnCode	returnMsg
999999	系统异常

returnCode	returnMsg
000000	交易成功
010001	服务调用异常
000000	交易成功
000055	失败
009999	系统繁忙，请稍后重试
002000	签名错误
003000	参数错误
009990	操作数据库失败
009992	银行验证未通过
010000	交易码有误
010002	没有收到对账文件
010003	账户余额不足
010004	查询不到付款联行号对应的省行联行号
010005	收款账号币种与交易币种不一致
010006	收款行名称为空且无法查询
010007	交易失败被银行退回
010008	开始和截止日期的间隔不能超过 3 月
010009	货币码有误：当前只允许人民币交易
010010	收款金额不能为空或者超过指定长度
010011	账号不存在
010012	账号户名不符
010013	原批次号不存在
010014	原流水号不存在
010015	商户号错误
010016	流水号重复
010017	账户状态异常
010018	时间戳校验失败
010019	对账日期错误

returnCode	returnMsg
010020	订单号（批次号）重复
010021	当前时间不支持该交易
010022	联行号输入错误
010023	收款行地址错误
010024	账户已销户
010025	账户已冻结
010026	超出授信额度
010027	该交易已发送，不允许撤销
010028	原交易不存在或已经撤销
010029	单笔交易金额超限
010031	收款行处理失败
010032	渠道不可用
010033	不支持该银行卡
010035	查无此单
020001	UserName 字段不允许为空
020012	该客户已开立电子账户
000000	交易成功
030016	流水号重复
030040	流水号参数没有设置
000000	交易成功
030221	身份证件信息未查到
030222	身份证件信息不一致
030122	身份证照片太小，无法识别，请重新上传
040001	输入信息[法人代码]不能为空！
040002	交易密码错误
040003	密码已锁定
040004	客户帐号余额不足
040005	查询日期范围过大, 请重新输入起止日期！

returnCode	returnMsg
040006	起始日期不能大于终止日期
050011	该账户尚未进行签约
060005	短信发送失败
000000	交易成功
060007	手机号码格式不对
000000	交易成功
070008	OCR 影像识别失败
100024	请求报文不能为空
100025	请求报文未上送手机号
100026	请求 esb 服务异常
100027	请求报文未上送验证码
100028	验证码不存在或已失效，请重新申请
100029	验证码已失效，请重新申请
100030	验证码校验不通过
100031	短信次数已到达阈值！
110002	没有收到对账文件
110006	读取非结构化配置文件失败
110007	读取 config 配置文件失败
110008	channelID 与 path 不能为空
110009	ChannelID 不能为空
110010	OCR 接口调用异常
110029	文件上传失败
110030	文件接口异常
110031	文件输出流关闭异常
0200009	非绑定卡不能进行充值或提现
020013	系统繁忙，请重试
020011	账号不存在
000008	该活动不存在

returnCode	returnMsg
000009	当前活动已关闭
000010	当前不在活动期
000011	该商户未参与当前活动
000012	客户当日参与活动次数达到限制
000013	客户参与活动总次数达到限制
000014	活动余额不足或名额已满
000015	活动审计表保存失败
010024	该商户未参与活动
010025	该商户暂无可用活动
010010	该应用对应的服务未申请
0100011	token 为必填项
0100012	报文中无签名
0100013	验签失败
0100014	该用户未实名认证!
0100015	该用户未签约!
0100016	该应用未审核!
010020	商户号验证失败
010021	验证交易平台编号失败
0100017	SDK 信息认证失败
0100018	加密机加签失败
1020001	Union 加密异常
010023	该商户未进行实名认证
010022	加解密算法类型不支持
1020002	未获取到报文加密参数
010007	请求参数不合法, 服务调用异常
010008	未识别的卡片信息
010006	营销数量更新失败
010009	用户已注册

1.2 Android 端

1. 配置

1.1 需要引入提供的文件

外部工程库：uilibrary

1.2 需要的依赖包

jar 包：njsdk_proguard.jar

1.3 需要权限

在 AndroidManifest.xml 文件中添加

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission
android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>

<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.RUN_INSTRUMENTATION" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission
android:name="android.permission.INTERACT_ACROSS_USERS_FULL" />
<uses-permission
android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
<uses-permission android:name="android.permission.VIBRATE" />

<activity
    android:name="szxx.sdk.ui.SDKH5Activity"
    android:exported="false"
    android:screenOrientation="portrait"/>

    <activity
        android:name="com.i2trust.auth.sdk.activity.FaceDetectActivity"
        android:theme="@android:style/Theme.Holo.NoActionBar"
        android:label="@string/ctid_str_main"
        android:screenOrientation="portrait" >
    </activity>
    <activity
        android:name="com.hisign.CTID.facelivedetection.CTIDLiveDetectActivity"
```

```

        android:configChanges="screenSize|keyboardHidden|orientation"
        android:launchMode="standard"
        android:screenOrientation="portrait" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
        </intent-filter>
    </activity>
</provider>
    <provider
        android:name="android.support.v4.content.FileProvider"
        android:authorities="你的应用包名.FileProvider"
        android:exported="false"
        android:grantUriPermissions="true">
        <meta-data
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/rc_file_path"/>
        </provider>

    <meta-data
        android:name="TD_APP_ID"
        android:value="@string/appid"/>
    <meta-data
        android:name="TD_CHANNEL_ID"
        android:value="@string/channelid"/>

```

1.4 混淆包时需要添加的 keep

```

-keep class cn.passguard.** { *; }
-keep class com.a.a.a.** { *; }
-keep class com.alibaba.fastjson.** { *; }
-keep class com.hisign.CTID.** { *; }
-keep class com.i2trust.auth.sdk.** { *; }
-keep class org.apache.http.** { *; }
-keep class org.bouncycastle.** { *; }
-keep class szxx.sdk.** { *; }
-dontwarn com.tendcloud.tenddata.**
-keep class com.tendcloud.**{*;}
-keep public class com.tendcloud.tenddata.**{
    public protected *;
}
-keepclassmembers class com.tendcloud.tenddata.**{
    public void *(*);
}
-keep class com.talkingdata.sdk.TalkingDataSDK{

```

```
public *;  
}  
-keep class com.apptalkingdata.**{  
    *;  
}
```

1.2.1 示例代码

```
//初始化 设定 appID 在 application 的 onCreate 中初始化  
SDK.instance().initSDK(this, "e77717e_830f_4056_9ef4_d432320c701d");  
  
//支付  
PayUtil.pay(Activity activity, String merchant, String idNo, String userName,  
String phoneNumber, Map<String, Object> payInfo, OnPayListener  
listener, String signCert, String signData)  
//我的钱包  
PayUtil.wallet(Activity activity, String merchant, String idNo, String userName,  
String phoneNumber, String signCert, String signData);
```

1.2.2 返回码列表

错误码表

returnCode	returnMsg
sdk000001	未初始化 SDK
sdk000002	未验证开发者
sdk010001	用户已开通钱包
sdk010002	用户未开通钱包
sdk010003	未调用一键登录
sdk010004	openID 不匹配
sdk010005	personUnionID 不匹配
sdk010006	身份证号格式不正确
sdk010007	姓名格式不正确
sdk010008	手机号格式不正确
sdk010009	银行卡号格式不正确

returnCode	returnMsg
sdk010010	openID 为空
sdk010011	personUnionID 为空
sdk010012	用户验证已通过
sdk010013	用户验证未通过
sdk020001	充值金额为空
sdk020002	提现金额为空
sdk020003	支付金额为空
sdk020004	支付金额必须等于订单金额减优惠金额
sdk020005	金额格式错误
sdk090001	网络错误

sdk 返回编码列表

returnCode	returnMsg
999999	系统异常
000000	交易成功
010001	服务调用异常
000000	交易成功
000055	失败
009999	系统繁忙，请稍后重试
002000	签名错误
003000	参数错误
009990	操作数据库失败
009992	银行验证未通过
010000	交易码有误
010002	没有收到对账文件
010003	账户余额不足
010004	查询不到付款联行号对应的省行联行号
010005	收款账号币种与交易币种不一致
010006	收款行名称为空且无法查询

returnCode	returnMsg
010007	交易失败被银行退回
010008	开始和截止日期的间隔不能超过 3 月
010009	货币码有误：当前只允许人民币交易
010010	收款金额不能为空或者超过指定长度
010011	账号不存在
010012	账号户名不符
010013	原批次号不存在
010014	原流水号不存在
010015	商户号错误
010016	流水号重复
010017	账户状态异常
010018	时间戳校验失败
010019	对账日期错误
010020	订单号（批次号）重复
010021	当前时间不支持该交易
010022	联行号输入错误
010023	收款行地址错误
010024	账户已销户
010025	账户已冻结
010026	超出授信额度
010027	该交易已发送，不允许撤销
010028	原交易不存在或已经撤销
010029	单笔交易金额超限
010031	收款行处理失败
010032	渠道不可用
010033	不支持该银行卡
010035	查无此单
020001	UserName 字段不允许为空

returnCode	returnMsg
020012	该客户已开立电子账户
000000	交易成功
030016	流水号重复
030040	流水号参数没有设置
000000	交易成功
030221	身份证件信息未查到
030222	身份证件信息不一致
030122	身份证照片太小，无法识别，请重新上传
040001	输入信息[法人代码]不能为空！
040002	交易密码错误
040003	密码已锁定
040004	客户帐号余额不足
040005	查询日期范围过大, 请重新输入起止日期！
040006	起始日期不能大于终止日期
050011	该账户尚未进行签约
060005	短信发送失败
000000	交易成功
060007	手机号码格式不对
000000	交易成功
070008	OCR 影像识别失败
100024	请求报文不能为空
100025	请求报文未上送手机号
100026	请求 esb 服务异常
100027	请求报文未上送验证码
100028	验证码不存在或已失效，请重新申请
100029	验证码已失效，请重新申请
100030	验证码校验不通过
100031	短信次数已到达阈值！

returnCode	returnMsg
110002	没有收到对账文件
110006	读取非结构化配置文件失败
110007	读取 config 配置文件失败
110008	channelID 与 path 不能为空
110009	ChannelID 不能为空
110010	OCR 接口调用异常
110029	文件上传失败
110030	文件接口异常
110031	文件输出流关闭异常
0200009	非绑定卡不能进行充值或提现
020013	系统繁忙，请重试
020011	账号不存在
000008	该活动不存在
000009	当前活动已关闭
000010	当前不在活动期
000011	该商户未参与当前活动
000012	客户当日参与活动次数达到限制
000013	客户参与活动总次数达到限制
000014	活动余额不足或名额已满
000015	活动审计表保存失败
010024	该商户未参与活动
010025	该商户暂无可用活动
010010	该应用对应的服务未申请
0100011	token 为必填项
0100012	报文中无签名
0100013	验签失败
0100014	该用户未实名认证！
0100015	该用户未签约！

returnCode	returnMsg
0100016	该应用未审核!
010020	商户号验证失败
010021	验证交易平台编号失败
0100017	SDK 信息认证失败
0100018	加密机加签失败
1020001	Union 加密异常
010023	该商户未进行实名认证
010022	加解密算法类型不支持
1020002	未获取到报文加密参数
010007	请求参数不合法, 服务调用异常
010008	未识别的卡片信息
010006	营销数量更新失败
010009	用户已注册

2、SDK 调用说明

2.1 IOS 端

2.1.1 初始化

在 AppDelegate 中添加#import "TalkingData.h",#, import "NJCallBackModel.h"头文件, 设置 SDK 运行所需参数, 示例代码:

```
@interface AppDelegate ()<NJRespDelegate> //遵守回调协议
```

```
@end
```

```
设置参数
```

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary
```

```
*)launchOptions {
```

```
    //设置支付回调
```

```
    NJCallBackModel* backModel = [NJCallBackModel defaultManager];
```

```
    backModel.NJdelegate=self;
```

```
    //收集错误日志,早于埋点初始化之前调用
```

```
[TalkingData setExceptionReportEnabled:YES];
//埋点
[TalkingData sessionStarted:@"XXX" withChannelId:@"XXX"];
//初始化 SDK
[NJSDK initNJSDKWithAppID:@"xxx" SDKDesc:@"这是 SDK 描述" SDKUUID:[NJSDK getSDKUUID]
SDKVersion:[NJSDK getSDKVersion] MachannelID: @"xxx" ShopNumber: @"xxx"];
return YES;
}
#pragma mark-实现回调协议方法
-(void)getRespCallBackFromNJ:(NJCallBackModel *)resp{

    if (resp.key ==NJ_REQ_KEY_PAYMENT_RESULT) {
        if ([resp.callBackResult[@"returnCode"] isEqualToString:NJSDK_PAYMENT_SUCCESS]) {
            NSLog(@"支付成功");
            //这里执行支付成功后的事情
        }else{
            NSLog(@"南京支付失败");
        }
    }
}
```

2.1.2 钱包

引入#import "NJSDK.h", 调用下面方法进入钱包页面

申明

```
/**
```

钱包

@param delegate 控制器

@param userName 用户姓名

@param IDNO 用户身份证号

@param userPhoneNumber 用户手机号

@param signCert 加密的证书信息

@param signData 签值

@param iphoneIP 手机当前 IP

*/

```
+(void)myPurseWithDelegate:(UIViewController<UIViewControllerTransitioningDelegate> *)delegate
    UserName:(NSString*)userName IDNO:(NSString*)IDNO
    UserPhoneNumber:(NSString*)userPhoneNumber SignCert:(NSString*)signCert
    SignData:(NSString*)signData IphoneIP:(NSString*)iphoneIP;
```

参数

delegate

一个继承 `UIViewControllerTransitioningDelegate` 的 `ViewController`，会从此 `viewController` `present` 出支付页面，请在此 `view Controller` 中实现如下方法：

```
- (UIPresentationController *)
presentationControllerForPresentedViewController:
(UIViewController *)presented
presentingViewController:
(UIViewController *)presenting
sourceViewController:
(UIViewController *)source{
    return
    [[NJCustomPresentationVc alloc]
    initWithPresentedViewController:presented
    presentingViewController:presenting];
}
```

username

支付者姓名

IDNO

支付者身份证号

userPhoneNumber

支付者手机号

signCert

加密的证书信息

signData

签值

iphoneIP

手机当前环境下的 IP，通过+(NSString*)getiPhoneIP 方法获得。**Ps:**此方法只适用于本平台。

2.1.3 支付

引入#import "NJSDK.h", #import "NJPaymentInfo.h", #import "NJCustomPresentationVc.h"

调用下面方法进入支付页面

申明

```
/**

支付

@param paymentInfo 支付的订单信息

@param delegate 控制器

@param userName 用户姓名

@param IDNO 用户身份证号

@param userPhoneNumber 用户手机号

@param signCert 加密的证书信息

@param signData 签值

@param iphoneIP 手机当前 IP

*/

+(void)payOrderWithPaymentInfo:(NJPaymentInfo *)paymentInfo
delegate:(UIViewController<UIViewControllerTransitioningDelegate> *)delegate
UserName:(NSString*)userName IDNO:(NSString*)IDNO
UserPhoneNumber:(NSString*)userPhoneNumber SignCert:(NSString*)signCert
SignData:(NSString*)signData IphoneIP:(NSString*)iphoneIP;
```

参数

paymentInfo

@property(nonatomic, copy) NSString *TrxAmt;//金额

@property(nonatomic, copy) NSString *PyeeNm;//收款方名称

@property(nonatomic, copy) NSString *PyeeAcctId;//收款方账户

@property(nonatomic, copy) NSString *PyeeAcctTp;//收款方账户类型

@property(nonatomic, copy) NSString *PyeeAcctIssrId;//收款方账户所属机构标识

@property(nonatomic, copy) NSString *OrdrId;//订单编号

@property(nonatomic, copy) NSString *OrdrDesc;//订单的简要描述

@property(nonatomic, copy) NSString *MrchntNm;//商户名称

@property(nonatomic, copy) NSString *MrchntNo;//商户编码

@property(nonatomic, copy) NSString *MrchntTp;//商户类型

@property(nonatomic, copy) NSString *MrchntCertTp;//商户证件类型

@property(nonatomic, copy) NSString *MrchntCertId;//商户证件编码

@property(nonatomic, copy) NSString *MrchntCtgyCd;//商户行业类别

@property(nonatomic, copy) NSString *MrchntPltfrmNm;//网络交易平台名

@property(nonatomic, copy) NSString *InstgId;// 非银行支付机构标识

@property(nonatomic, copy) NSString *InstgAcctId;// 非银行支付机构?备付金账户

@property(nonatomic, copy) NSString *InstgAcctIssrId;// 备付金账户所属? 机构标识

delegate

遵守 UINavigationControllerTransitioningDelegate 协议的 ViewController，会从此 viewController present 出支付页面，请在此 view Controller 中实现如下方法：

```
- (UIPresentationController *)
presentationControllerForPresentedViewController:
(UIViewController *)presented
```

```
presentingViewController:

(UIViewController *)presenting

sourceViewController:

(UIViewController *)source{

    return

    [[NJCustomPresentationVc alloc]

    initWithPresentedViewController:presented

    presentingViewController:presenting];

}
```

若不实现此方法将会收到错误信息。正常展示页面效果如下：



username

支付者姓名

IDNO

支付者身份证号

userPhoneNumber

支付者手机号

signCert

加密的证书信息

signData

签值

iphoneIP

手机当前环境下的 IP，通过 `+(NSString*)getiPhoneIP` 方法获得。Ps:此方法只适用于本平台。

返回

key

NJ_REQ_KEY_PAYMENT_RESULT

callbackResult

此返回对象为 `id` 类型的对象，以 JSON 格式说明：

```
{
    returnCode: "", // 结果代码
    returnMsg: "", // 结果信息
}
```

2.1.4 系统版本号

声明

/**

获得 SDK 版本号

@return sdk 版本号

***/**

+(NSString*)getSDKVersion;

2.1.5 当前手机 IP


```
声明
/**
    获得当前 IP

    @return 当前网络 IP
    */
+ (NSString*)getSDKiPhoneIP;
```

2.1.6 设备唯一标识

```
声明
/**
    获得设备唯一标识

    @return UUID
    */
+ (NSString*)getSDKUUID;
```

2.1.7 bundleid

```
声明
/**
    获得 bundleid

    @return bundleid
    */
+ (NSString*)getSDKApplicationID;
```

2.1.8 手机系统版本号

```
声明
/**
    获得系统版本号

    @return 系统版本号
    */
+ (NSString*)getSDKOsVersion;
```

2.2 Android 端

signCert、signData

为动态生成的签值和证书，会提供生成方式

2.2.1 初始化

在使用 sdk 功能之前，必须先调用初始化接口 `initSDK`，让 sdk 完成初始化配置，在应用生命周期中只需初始化一次即可。

初始化 SDK 配置(`initSDK`)在 `application` 的 `oncreate` 中

输入

```
/**
 *
 * @param AppId    app 识别号
 * @param context 当前 activity 的 context
 */ SDK.instance().initSDK(this, "e77717e_830f_4056_9ef4_d432320c701d");
```

2.2.2 支付

调用支付

```
Map<String, Object> orderInfo = new TreeMap<>();
    orderInfo.put("TrxAmt", e1t); // 支付金额
    orderInfo.put("PyeeNm", e2t); // 收款方
    orderInfo.put("PyeeAcctId", e3t); // 收款方账户
    orderInfo.put("PyeeAcctTp", "0"); // 收款方账户类型
    orderInfo.put("PyeeAcctIssrId", "6879"); // 收款方账户所属机构标识
    orderInfo.put("OrdrId", e4t); // 订单 id
    orderInfo.put("OrdrDesc", e5t); // 订单描述
    JSONObject MrchntInfo = new JSONObject();
    MrchntInfo.put("MrchntCtgyCd", "2564"); // 商户行业类别
    MrchntInfo.put("MrchntCertId", "222403198806160012"); // 商户证件编码
    MrchntInfo.put("MrchntTp", "1"); // 商户类型
    MrchntInfo.put("MrchntNo", "2108699897"); // 商户编码
    MrchntInfo.put("MrchntNm", "京东"); // 商户名称
    MrchntInfo.put("MrchntCertTp", "01"); // 商户证件类型
    //    MrchntInfo.put("MrchntPltfrmNm", "888"); // 网络交易平台名称
    orderInfo.put("MrchntInfo", MrchntInfo);
    JSONObject InstgInfo = new JSONObject();
    InstgInfo.put("InstgAcctIssrId", "888"); // 备付金账户所属机构标识
```

```
InstgInfo.put("InstgAcctId", "888");//非银行支付机构备付金账户
InstgInfo.put("InstgId", "888");//非银行支付机构标识
orderInfo.put("InstgInfo", InstgInfo);
PayUtil.pay(this, str1, str2, str3, str4, orderInfo, new OnPayListener()
{
    @Override
    public void onPayResult(Map<String, Object> payResult) {
        Message message = Message.obtain();
        message.obj = payResult;
        mHandler.sendMessage(message);
    }
}, signCert, signData);
```

2.2.3 钱包

进入我的钱包

```
PayUtil.wallet(this, str1, str2, str3, str4, signCert, signData);
```