**苏宁银行金融开放平台SDK接入使用手册**

目录

[一、操作步骤 1](#_Toc4093801)

[二、API请求 2](#_Toc4093802)

[三、签名验签,H5请求 3](#_Toc4093803)

[四、系统技术说明 4](#_Toc4093804)

[五、附件 6](#_Toc4093805)

### 一、操作步骤

#### 1.1 导入SDK加包

苏宁银行方会将SDK源码以及jar包同时提供给开发者，若开发者修改了源码需要重新打包生成jar包。

在你的工程中导入jar包，名字为 snb-fsos-openapi-0.0.2.jar。Maven工程，配置信息为：

|  |
| --- |
| <dependency>  <groupId> com.snb.fsos</groupId>  <artifactId> snb-fsos-openapi</artifactId>  <version>0.0.2</version> </dependency> |

#### 2.2 使用DEMO类发送测试请求

以下以API接口测试DEMO类 TestSnbApiHelper介绍，测试您的环境与苏宁银行测试环境网络是否联通。

1） 使用SnbSignUtil.setEnv(InitData.ENV.sit)初始化SDK连接到苏宁银行金融开放平台系统的SIT测试环境。

2）使用SnbSignUtil.setKeyMetadata("11111111", "11111111", null);初始化您自己的私钥证书路径以及证书密码。若初始化路径为null，则SDK取初始默认路径。（详见源码）

3）使用SnbSignUtil.initApi("DEMO", "BJLJ001");初始化苏宁银行分配给您的‘应用编号’和‘渠道编号’。

4）SnbRequestBean为发送数据的包装类，包含了接口文档中的公共报文头和业务数据payload部分。公共报文头中除了‘流水号’和‘交易码’字段之外，其他 字段SDK都已经封装好了，无需设置。

流水号填入您系统为此交易分配的流水号，若您系统没有生成流水号能力请联系苏宁银行开发人员解决。

‘交易码’字段值详见接口文档中定义的值。测试网络连通性交易，该值为固定值‘snb.connect.test’。该交易同时验签了您发送过来的报文数据格式是否正确，签名私钥是否匹配。

5）使用具体业务数据Bean封装您的业务数据然后set到SnbRequestBean中，测网络连通性交易，不校验该业务数据Bean。

6) 调用SnbApiHelper中send方法生成随机秘钥，使用您的私钥签名数据，使用苏宁银行测试公钥加密数据，生成最终请求报文。再使用org.apache.http.client.HttpClient 发送请求到苏宁银行。（若您系统已经有了发送Http请求的方法，可以替换jar包中发送数据的部分代码）

该方法同时验签了苏宁银行返回的业务数据，验证通过后，将数据转化成Map。

7）根据苏宁银行返回的业务数据Map,进行您后续的业务流程。

### 二、API请求

**示例代码:**

**package** com.snb.fsos.testdemo;  
**import** com.snb.fsos.bean.BalanceQuery;  
**import** com.snb.fsos.bean.SnbRequestBean;  
**import** com.snb.fsos.util.InitData;  
**import** com.snb.fsos.util.SnbApiHelper;  
**import** com.snb.fsos.util.SnbSignUtil;  
**import** java.util.Map;  
**public class** TestSnbApiHelper {  
 **public static void** main(String[] args){  
 *// 初始设置一次即可 ，三个方法顺序不能变动↓*  SnbSignUtil.setEnv(InitData.ENV.sit);  
 SnbSignUtil.setKeyMetadata("11111111", "11111111", null);  
 SnbSignUtil.initApi("DEMO", "BJLJ001");  
 *// 初始设置一次即可 ↑*  SnbRequestBean snbRequestBean = new SnbRequestBean();  
 snbRequestBean.setChannelSerialNo("12323232000081114");// 唯一交易流水号  
 snbRequestBean.setTransCode("snb.connect.test");// 交易码  
  
 // 具体业务数据（payload）。根据不同的接口,调用包com.snb.fsos.bean下面的具体bean。  
// 如果包下面没有符合实际使用的Bean,需要自定义bean，该bean继承PayLoadBean即可。  
 BalanceQuery bq = new BalanceQuery();  
 bq.setElectAcct("111111212121223512");  
 bq.setMerchantId("1111");  
 // 具体业务数据  
  
 snbRequestBean.setPayLoadBean(bq);  
 try {  
 Map res = SnbApiHelper.send(snbRequestBean);  
 if("-1".equals(res.get("respCode"))){  
 System.out.println("客户端检查请求参数出错[" + res.get("respMsg") + "]");  
 }else{  
 // DO  
 System.out.println("返回数据:" + res);  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

### 三、签名验签,H5请求

**package** com.snb.fsos.testdemo;  
**import** com.snb.fsos.util.InitData;  
**import** com.snb.fsos.util.SnbSignUtil;  
**import** java.util.HashMap;**public class** TestSignData {  
 **public static void** main(String[] args){  
 *// 初始设置一次即可, 三个方法顺序不能变动 ↓*  SnbSignUtil.setEnv(InitData.ENV.sit);  
 // 设置商户秘钥路径，以及密码。文件路径为Null时取默认路径 InitDta.MerPriKeyPath。  
 SnbSignUtil.setKeyMetadata("11111111", "11111111", null);  
 SnbSignUtil.init("9111000077255233120001", "ZCXC0001", "ZCXC0001", "100002", "402");  
 // 初始设置一次即可 ↑  
 HashMap payload = new HashMap();  
 payload.put("channelSerialNo", "123456789000127");// 唯一流水号  
 payload.put("transCode", "snb.tranCode");// 交易码  
  
 // 业务数据,根据具体接口设置字段名称以及相关值  
 payload.put("idNo", "440105199910107051");  
 payload.put("bindMobile","18620127971");  
 payload.put("userName", "尹一");  
 payload.put("merchantUserNo", "511423198902140138");  
 try {  
 String url = SnbSignUtil.signData(payload);  
 System.out.println("url: " + url);  
 // 重定向到以下 URL地址即可  
 String rurl = "redirect:" + url;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

### 四、系统技术说明

#### 4.1 H5请求调用摄像头

APP开发人员需要注意，H5外输产品需要具备调用摄像头的权限，在安卓系统中需要通过代码赋予WebChrome调用摄像头的能力。安卓5.0以前系统需要实现openFileChooser函数。5.0及以后，重写onShowFileChooser函数。以下为示例代码：

private class MyWebChromeClient extends WebChromeClient {

public void openFileChooser(ValueCallback<Uri> uploadMsg, String acceptType) {

mFileChooser = uploadMsg;

}

public void openFileChooser(ValueCallback<Uri> uploadMsg) {

mFileChooser = uploadMsg;

}

public void openFileChooser(ValueCallback<Uri> uploadMsg, String acceptType, String capture) {

mFileChooser = uploadMsg;

}

@Override

public boolean onShowFileChooser(WebView webView, ValueCallback<Uri[]> filePathCa

llback, FileChooserParams fileChooserParams) {

mShowFileChooser = filePathCallback;

return true;

}

}

#### 4.2 苏宁银行API中公私钥使用简述

请求分为发送数据者和接收数据者，发送数据者（以下简称A）有自己的私钥证书(pfx格式文件)以及对应的公钥证书（cer格式文件），接收数据者（以下简称B）也有自己的私钥证书和对应的公钥证书。

发送数据时，A使用B的公钥生成随机秘钥Key，然后使用自己的A私钥证书对发送报文签名，最终将数据发送给B。B接收到A发送过来的数据后，使用自己B的私钥证书解密，然后使用A的公钥验签。

A和B通过线下的方式，分别将自己的公钥发送给对方。**自己的私钥要严格保密，一旦泄露，及时通知对方更换证书保证业务安全。**

B(苏宁银行)私钥解密

A公钥验签

B（苏宁银行）公钥加密

A私钥加签

A私钥解密

B(苏宁银行)公钥验签

A公钥加密

B(苏宁银行)私钥加签

### 五、附件

#### 附件1： 产品名称对应表

|  |  |  |
| --- | --- | --- |
| **产品编码(productCode)** | **值** |  |
| 中诚信 | 100002 |  |
| 众信普惠-车抵贷 | snbcdd |  |
|  |  |  |
|  |  |  |