

Android NFC 充值 SDK 集成说明

版本 V1.0

文档编号		项目名称	
编写		编写日期	2018 年 09 月 13 日
确认签字 (常熟市民卡有限公司)	确认时间:		

文档修订历史记录

日期	版本	说明	作者	审批人
2018-09-13	1.0	建立	贾政	

目 录

API 介绍.....	4
initNfc.....	4
obtain.....	4
getCardNo.....	4
getBalance.....	4
startCreditForLoad.....	5
SDK 使用方法.....	7
导入 sdk.....	7
配置权限.....	7
错误码对照表.....	8

API 介绍

本 SDK 只针对 NFC 方式读写。

由于数据传输不是实时响应，部分采用了 callback 回调实现响应。

SDK 对外的所有接口均由 NfcSdk 类提供。

initNfc

`public void initNfc(Activity context)`

初始化 nfc 模块，必须在 Activity onCreate 里调用

函数参数：

context: 为当前 Activity 的实例

返回：

无

obtain

`public static NfcSdk obtain()`

获取 NfcSdk 实例对象

函数参数

无

返回：

NfcSdk 实例对象

getCardNo

`public String getCardNo(Intent intent)`

获取市民卡卡号

函数参数：

intent: 为 Activity onNewIntent(Intent intent) 回调函数中传入的 intent

返回：

市民卡卡号，异常返回空字符串

getBalance

`public int getBalance(Intent intent)`

获取市民卡余额

函数参数:

intent: 为 Activity `onNewIntent(Intent intent)` 回调函数中传入的 `intent`

返回:

市民卡市民卡余额（单位分），异常返回-1

startCreditForLoad

```
public void startCreditForLoad(Context context, Intent intent, String mobile, int amount, String
serialNumber, String appKey, String appSecret, String IMEI, CreditForLoadListener listener)
```

获取市民卡卡号

函数参数:

@param context Application Context

@param intent 为 Activity `onNewIntent(Intent intent)` 回调函数中传入的 `intent`

@param mobile 手机号

@param amount 充值金额（单位分）

@param serialNumber 订单号

@param appKey 应用 key

@param appSecret 秘钥

@param IMEI 读取手机 IMEI

@param listener 充值结果回调

返回:

无参数返回，充值结果由 `CreditForLoadListener` 回调函数带回 包含 `code` 和 `msg`（见错误码对照表）

/**

* 圈存结果回调

*/

```
public interface CreditForLoadListener {
    void onFailed(int errCode, String errMsg);
    void onSuccess(int code, String msg);
}
```

调用示例

```
package com.dceast.tech.sdk.nfcdemo;
```

```
import android.app.PendingIntent;
```

```
import android.content.Intent;
```

```
import android.nfc.NfcAdapter;
```

```
import android.os.Bundle;
```

```

import android.support.v7.app.AppCompatActivity;

import com.dceast.tech.sdk.nfc.NfcSdk;
import com.dceast.tech.sdk.nfc.listener.CreditForLoadListener;
import com.dceast.tech.sdk.nfc.util.DeviceUtils;
import com.dceast.tech.sdk.nfc.util.Param;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        NfcSdk.obtain().initNfc(this);
    }

    @Override
    protected void onResume() {
        super.onResume();
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, new Intent(this,
            getClass()).addFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP), 0);

        NfcAdapter.getDefaultAdapter(this).enableForegroundDispatch(this, pendingIntent, NfcSdk.obtain().getIntentFiltersArray(), NfcSdk.obtain().getTechListsArray());
    }

    @Override
    protected void onPause() {
        super.onPause();
        NfcSdk.obtain().disableNFC(this);
    }

    @Override
    protected void onNewIntent(Intent intent) {
        super.onNewIntent(intent);
        String IMEI = DeviceUtils.getDeviceId(getApplicationContext());
        NfcSdk.obtain().startCreditForLoad(getApplicationContext(), intent, Param.MOBILE,
10, Param.SERIAL_NUMBER, Param.APPKEY, Param.APP_SECRET, IMEI, new CreditForLoadListener() {
            @Override
            public void onFailed(int i, String s) { //充值失败（包括圈存成功，圈存确认失败的情况）
            }

            @Override

```

```

        public void onSuccess(int i, String s) { //充值成功

        }

    });
}
}

```

SDK 使用方法

导入 sdk

将 nfc-release.aar 导入到调用者工程的 libs 目录下。
在主工程 gradle 配置文件中加入 okhttp 和 gson 库

```

android {
    testImplementation 'com.android.support.test:'
    implementation(name: 'nfc-release', ext: 'aar')
    implementation 'com.google.code.gson:gson:2.8.0'
    implementation 'com.squareup.okhttp3:okhttp:3.4.1'
}

```

配置权限

在调用者工程的 AndroidManifest.xml 中增加权限（NFC，android sdk 最小版本要在 14 及以上）。

权限

```

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.NFC" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

```

错误码对照表

```
public static final int CODE_LOAD_COMPLETE = 0x100;
public static final int CODE_READ_CARD_NO_ERR = 0x101;
public static final int CODE_READ_CARD_BALANCE_ERR = 0x102;
public static final int CODE_READ_CARD_INFO_ERR = 0x103;
public static final int CODE_INITIALIZE_FOR_PURCHASE_ERR = 0x104;
public static final int CODE_INITIALIZE_FOR_LOAD_ERR = 0x105;
public static final int CODE_CREDIT_FOR_LOAD_ERR = 0x106;
public static final int CODE_REQUEST_APPLY_LOAD_ERR = 0x107;
public static final int CODE_REQUEST_CONFIRM_LOAD_ERR = 0x108;

private static HashMap<Integer, String> map = new HashMap<>();

static {
    map.put(CODE_LOAD_COMPLETE, "卡片圈存成功");
    map.put(CODE_READ_CARD_NO_ERR, "读取卡片号码错误");
    map.put(CODE_READ_CARD_BALANCE_ERR, "读取卡片余额错误");
    map.put(CODE_READ_CARD_INFO_ERR, "读取卡片信息错误");
    map.put(CODE_INITIALIZE_FOR_PURCHASE_ERR, "卡片消费初始化错误");
    map.put(CODE_INITIALIZE_FOR_LOAD_ERR, "卡片圈存初始化错误");
    map.put(CODE_CREDIT_FOR_LOAD_ERR, "卡片圈存错误");
    map.put(CODE_REQUEST_APPLY_LOAD_ERR, "圈存申请错误");
    map.put(CODE_REQUEST_CONFIRM_LOAD_ERR, "圈存完成，圈存确认请求错误");
}
```