

# 目录

---

- [项目介绍](#)
- [对于云 POS 厂商](#)
  - [设计总体原则](#)
  - [总体要求](#)
  - [总体框架](#)
- [对于第三方应用开发商](#)
  - [SDK 包目录结构](#)
  - [使用说明](#)
    - [下载](#)
    - [使用](#)
    - [接口调用](#)
    - [权限设置](#)

## 项目介绍

---

智能云 POS SDK（Java 接口）为应用程序操作终端设备提供统一的开发工具包。它由每个终端厂商实现，并开放给第三方应用开发商。终端厂商负责终端的硬件设备及驱动，终端的操作系统。每个终端厂商根据提供的 SDK 接口及自己终端的实际情况实现 SDK。第三方应用需要操作终端设备时，只需调用相应的 SDK 接口就可以了。

JAVA SDK 访问网址 <https://github.com/ZhangUP/CloudposSDK>

## 对于云 POS 厂商

---

→ SDK 实现部分由各个终端厂商负责，终端上每个设备的操作接口都定义在 SDK 中，一般情况下，磁条卡阅读器、接触式 IC 卡阅读器、非接触式 IC 卡阅读器、PIN 输入设备、证书管理与加密运算模块、打印机、蜂鸣器、LED 灯等外设是必须实现的，对于串口、客显设备、钱箱、身份证设备等外设，终端厂商会有选择性的实现。未实现的部分不会 影响已存在的接口。第三方应用开发商开发时，要和终端厂商确认某个具体的外设接口是否可以使用。

## 设计总体原则

- 扩展性 增加额外设备不影响已有 API
- 兼容性 设备具有额外功能时，用户不用关心额外功能

## 总体要求

- 获取终端型号和 OS 版本号
- 设备发现
- 可以判断机器中是否存在某个设备
- 罗列机器中的所有设备
- 接口版本管理
- 设备特性获取
- 获取设备具有的特性信息
- 支持异步操作

## 总体框架



→ 第三方应用开发商开发的应用通过 SDK 接口访问终端设备。SDK 接口部分定义 Java 层 Interface 及部分实现。多数都是 Interface, 少部分如 Exception, ATR, MoneyValue-钱箱, Format-打印机格式等是具体实现。

## SDK 包目录结构

→ SDK 包中除了包含 SDK 接口部分代码外, 还包含文档及示例代码。目录结构如下所示:

→ demo:场景示例代码。

→ doc:API 开发文档, SDK 文档。

→ lib: SDK 接口 jar 包。

## 使用说明

### 下载

→ 从银联指定地址下载最新的 SDK。

### 使用

→ 一般情况下, 只需将下载好的 lib 目录下 SDK 接口 jar 包添加到开发工具的编译路径下, 就可以使用了。

### 接口调用

→ 如果应用中使用到了某个具体的设备, 通过 POSTerminal 来获取对应的设备对象, 如下所示:

```
XXDevice XXDevice =(XXDevice)POSTerminal.getInstance().getDevice("设备名称定义");
```

获取设备对象后, 可以调用相应接口进行操作, 比如打开设备, 关闭设备等。

设备名称定义如下所示：

- 磁条卡阅读器：cloudpos.device.msr
- 安全模块：cloudpos.device.hsm
- 非接触式 IC 卡阅读器：cloudpos.device.rfcardreader
- 身份证读取设备：cloudpos.device.idcardreader
- 打印机：cloudpos.device.printer
- PIN 输入设备：cloudpos.device.pinpad
- 接触式 IC 卡阅读器：cloudpos.device.smartcardreader
- 串口设备：cloudpos.device.serialport
- LED 设备：cloudpos.device.led
- 钱箱设备：cloudpos.device.cashdrawer
- 客显设备：cloudpos.device.secondarydisplay

如果应用中使用了异步操作，那么必须定义自己的 `OperationListener`，在回调函数 `handleResult()` 中对返回结果进行处理。如下所示：

```
OperationListener operationListener = new OperationListener(){  
    public void handleResult(OperationResult result) {  
        // handleResult  
    }  
};
```

不同设备的 `OptionResult` 是不同的。比如磁条卡阅读器的 `OptionResult` 可以获取磁道信息。非接触式 IC 卡阅读器的 `OptionResult` 可以获取卡片。然后对卡片进行后续的操作。

## 权限设置

→ 终端中的每个设备驱动都需要声明权限后才能使用。因此应用使用驱动时，需要在 `AndroidManifest.xml` 中增加相应的权限声明。

- 访问安全模块权限

```
<uses-permission  
    android:name="android.permission.CLOUDPOS_SAFE_MODULE"/>
```

- 访问磁条卡读卡器设备权限

```
<uses-permission android:name="android.permission.CLOUDPOS_MSR" />
```

- 访问接触式 IC 卡读卡设备权限

```
<uses-permission android:name="android.permission.CLOUDPOS_SMARTCARD"  
 />
```

- 访问非接触 IC 卡读卡设备权限

```
<uses-permission  
    android:name="android.permission.CLOUDPOS_CONTACTLESS_CARD" />
```

- 访问打印机设备权限

```
<uses-permission android:name="android.permission.CLOUDPOS_PRINTER" />
```

- 密码键盘访问权限

```
<uses-permission android:name="android.permission.CLOUDPOS_PINPAD" />
```

- 密码键盘计算 Pinblock 权限

```
<uses-permission  
    android:name="android.permission.CLOUDPOS_PIN_GET_PIN_BLOCK" />
```

- 密码键盘计算 Mac 权限

```
<uses-permission android:name="android.permission.CLOUDPOS_PIN_MAC" />
```

- 密码键盘加密数据权限

```
<uses-permission  
    android:name="android.permission.CLOUDPOS_PIN_ENCRYPT_DATA" />
```

- 密码键盘更新用户密钥权限

```
<uses-permission  
    android:name="android.permission.CLOUDPOS_PIN_UPDATE_USER_KEY" />
```