

# ApkPlugComponentSDK 使用文档

1	介绍.....	1
2	使用步骤.....	1
2.1	申请授权.....	1
2.2	配置主应用权限.....	1
2.3	添加库支持.....	2
2.4	SDK 初始化.....	2
2.5	使用组件.....	2
2.6	添加自定义组件.....	3
3	支持组件.....	3
3.1	脉可寻 SDK.....	3
3.1.1	验证功能.....	3
3.1.2	验证功能包括三个函数:.....	3
3.1.3	设置 Sdcard 图片缓存路径.....	4
3.1.4	使用相机拍照并获取名片信息.....	4
3.1.5	获取名片信息.....	4
3.1.6	类.....	4
3.1.7	回调接口.....	4
3.2	极验验证组件.....	5
3.2.1	设置 captcha_id.....	5
3.2.2	验证方法.....	5
3.2.3	验证回调接口.....	5

## 1 介绍

ApkPlugComponentSDK 是为方便开发者使用第三方组件开发的 SDK, SDK 内部集成通用的第三方组件(脉可寻组件, 极验验证组件, IM 环信组件等), 可供开发者直接使用。如果开发者想要使用定制自定义组件, 只需要把制作好的自定义组件(apk 文件)添加到主项目, 并在 json 文件中注册自定义组件信息, 开发者就可以像使用内部集成组件一样使用自定义组件。

Github 地址: [https://github.com/apkplug/apkplug\\_component\\_sdk](https://github.com/apkplug/apkplug_component_sdk)  
已支持组件在 Git 仓库 components 目录下。

## 2 使用步骤

### 2.1 申请授权

在 <http://www.apkplug.com/> 申请授权 ID, 并下载 SDK。

### 2.2 配置主应用权限

主应用需要几个基础的权限配置, 请将以下的几个权限加入到主应用的 AndroidManifest.xml 中

```
<!-- 插件平台需要的权限! -->
```

```
<uses-permission
```

```

        android:name="android.permission.WRITE_EXTERNAL_STORAGE">
    </uses-permission>
    <uses-permission
        android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
    <uses-permission
        android:name="android.permission.INTERNET"/>
    <uses-permission
        android:name="android.permission.READ_PHONE_STATE">
    </uses-permission>

```

另外将一下加入到<application></application>节点中

```
<!-- 插件平台需要的配置! -->
```

```

    <activity
        android:name="org.apkplug.app.apkplugActivity"
        android:theme="@style/android:Theme.Light"
        android:configChanges="orientation|keyboardHidden"
    />

```

最后将我们从 Apkplug 管理后台申请到的 AppAuth 加入到配置文件中

```

<meta-data
    android:name="apkplugauth" android:value="xxxxxxx" >
</meta-data>

```

如下图:

```

<meta-data android:name="apkplug-theme-auth" android:value="Lxr139HYhnzz" />
<meta-data android:name="apkplug-auth" android:value="IsT0Vwk3De2cHOCDBk" />
<meta-data android:name="apkplug-appid" android:value="1a2db8da65fe4c40" />
    <activity
        android:name="org.apkplug.app.apkplugActivity"
        android:theme="@android:style/Theme.Translucent"
        android:configChanges="orientation|keyboardHidden"
    />
</application>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></u
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>

```

### 2.3 添加库支持

拷贝 BundleCloud2.8.5-Release.jar 到主应用 lib 目录。

### 2.4 SDK 初始化

在 Application 类中初始化插件管理器

```

frame=FrameworkFactory.getInstance().start(null, this, null);
final BundleContext context=frame.getSystemBundleContext();
ComponentManager.getInstance().onInit(context);

```

### 2.5 使用组件

通过 ComponentManager 获取所需组件

```

ComponentManager.getInstance().
searchComponent(ComponentFactory.getInstance().getComponent("

```

```
组件名称”),
new ServerCallback<组件接口类型>() {
    @Override
    public void onSuccess(组件接口类型 service) {
        //成功获取到了组件的服务

    }
    @Override
    public void onFailure(int errorNo, String strMsg)
{
}
});
```

2.6 添加自定义组件

把制作好的自定义组件添加到主应用 assets 目录下，并在 assets 目录下的 components.json 中注册自定义组件信息。

3 已支持组件

已支持组件	组件名称	组件接口类型
脉可寻	maketionSdk	IMaketionCardSDK
极验验证	GeeTestSdk	GtSDK
环信 IM	ChatUIDemo	LaunchPlug
Udesk	UdeskDemo	LaunchPlug

3.1 脉可寻 SDK

3.1.1 验证功能

```
public void authenticateAccount
(String pkey,String psign,String name);
public boolean checkAuthentication();
public void clearAuthentication();
```

3.1.2 验证功能包括三个函数：

- authenticateAccount() 提交验证；
- checkAuthentication() 判断验证状态；
- clearAuthenticationh() 清除验证状态。

参数 key: 由脉可寻名片识别服务申请获得;

### 3.1.3 设置 Sdcard 图片缓存路径

```
public void setSavePath(String path);
```

设置 Sdcard 路径后, 拍摄名片所生成的图片, 会保持在此路径下。

### 3.1.4 使用相机拍照并获取名片信息

```
public void takepic(PlugMkxBackUpload callback);
```

### 3.1.5 获取名片信息

```
public void getCardsWithTime(long time, PlugMkxBackCards  
callback);
```

```
public void getCardsWithUUID(String[]
```

```
uuids, PlugMkxBackCards callback);
```

获取名片信息有两种方式: 通过名片的关键字 uuid 获取名片信息; 获取一个时间点之后的所有名片信息。

参数 uuids: 字符串数组, 每个字符串为一个名片的 uuid;

参数 time: 时间点, 1970 年 1 月 1 日开始经过的“秒”(数若此参数为 0 表示获取所有名片信息, 不包含已删除的名片和无法识别的名片);

### 3.1.6 类

PlugMkxCard

成员变量

类型	名称
String	address;
int	audit;
String	carduuid;
String	cname;
long	createtime;
String	duty;
String	email;
String	fax;
String	fields;
int	flag;
String	logo;
String	mobile1;
String	mobile2;
String	name;
String	tell1;
String	tell2;
long	updatetime;
String	website;

### 3.1.7 回调接口

```
public interface PlugMkxBackCards {
```

```
    public void onBack(int code, String info,
```

```
    ArrayList<PlugMkxCard> cards);
```

```

        //参数 code: 网络调用状态;
        //参数 errInfo: 当网络调用失败时, 返回错误参考信息;
        //参数 cards: 返回的名片信息数组。
    }
    public interface PlugMkxBackUpload {
        public void onBack(int code, String errInfo, String
        uuid, int status);
        //参数 code: 网络调用状态;
        //参数 errInfo: 当网络调用失败时, 返回错误参考信息;
        //参数 uuid: 当前上传名片 uuid;
        //参数 status: STATUS_START 表示开始上传,
        STATUS_SUCESS 表示上传成功, STATUS_ERROR 表示上传失败。
    }

```

### 3.2 极验验证组件

#### 3.2.1 设置 captcha\_id

```
void setCaptcha(captcha_id);
```

#### 3.2.2 验证方法

```
GtTest(custom_server_validate_url, plugGtListener);
```

custom\_server\_validate\_url 为服务器 url  
plugGtListener 为验证回调接口

#### 3.2.3 验证回调接口

```
public interface PlugGtListener {
    public void gtResult(String response)
}

```

response 为返回验证信息