

概述与资源

Android SDK 提供给集成 Android 原生客户端开发的开发者使用。

环境需求

条目	资源
开发目标	Android 4.0 以上
开发环境	Android Studio 2.1.3
sdk三方依赖	<code>com.cmhc.sso.sdk</code>

相关开发资料

条目	资源
产品结构流程	交互流程 , 通讯流程
SDK 下载	opv2-android-sdk
接口文档	opv2-android-api-docs
错误码	Error Code 列表

安装

远程依赖

```
implementation 'com.geetest.android:onepass:0.3.4'
```

获取 SDK

- 在 Demo 的 `libs` 包下，将获取的 `.aar` 文件拖拽到工程中的 `libs` 文件夹下。
- 在拖入 `.aar` 到 `libs` 文件夹后, 还要检查 `.aar` 是否被添加到 **Library**，要在项目的 `build.gradle` 下添加如下代码：

```
repositories {
    flatDir {
        dirs 'libs'
    }
}
```

并且要手动将aar包添加依赖：

```
implementation(name: 'geetest_onepassv2_android_vx.y.z', ext: 'aar')
```

3. 添加权限清单

```
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
```

配置接口

开发者集成客户端 SDK 前, 必须先您的服务器上搭建相应的**服务端SDK**, 配置**verifyUrl**, 并配置从**极验后台**获取的 APP_ID。这里以服务端**verifyUrl**配置成功, 客户端开发步骤为例, 如下:

1. 配置初始化接口

```
OnePassHelper.with().init(MainActivity.this);
```

2. 调用校验接口

```
OnePassHelper.with().getToken(phone, appId, onePassListener);
//第一个参数为输入的手机号码
//第二个参数为所需要配置的APP_ID
//第三个参数为所需实现监听回调结果接口
```

集成代码参考下方的 **代码示例**

代码示例

初始化与校验

初始化

在项目的具体页面的 onCreate 方法里面进行初始化。

```
OnePassHelper.with().init(MainActivity.this);
```

调用校验

```
OnePassHelper.with().getToken(phone, appId, onePassListener);
```

接口实现

实现接口进行校验。

```
OnePassListener onePassListener = new OnePassListener() {  
    @Override  
    public void onTokenFail(JSONObject jsonObject) {  
        //过程中出现的错误，具体参考下方错误码  
    }  
  
    @Override  
    public void onTokenSuccess(JSONObject jsonObject) {  
        //sdk输出的参数，通过这些参数请求verifyUrl判断是否是本机号  
    }  
  
    @Override  
    public boolean onAlgorithm() {  
        //是否需要加密手机号，true表示加密，false表示不加密  
    }  
};
```

页面关闭

在页面关闭的时候执行此方法。

```
@Override  
protected void onDestroy() {  
    super.onDestroy();  
    OnePassHelper.with().cancel();  
}
```