e签宝快捷签产品

SDK1.5.2说明

杭州天谷信息科技有限公司

2016年12月

**目录**

[E签宝快捷签Android SDK使用说明 4](#_Toc24592)

[SDK初始化 4](#_Toc16089)

[SDK个性化界面设置(可选项) 5](#_Toc1795)

[创建快捷签印章 5](#_Toc25766)

|  |  |  |
| --- | --- | --- |
| **版本** | **作者** | **修改时间** |
| **V1.5.2** | **赵佰枫** | **2017/03/03** |

# E签宝快捷签Android SDK使用说明

使用说明:.第三方应用需要在工程中引入 E签宝的两个.aar包,分别为QuickSignSDK-release.aar， tSignNetLib-release.aar。

# SDK初始化

原型:public synchronized void init(Context ctx, String project\_id, String project\_secret, EnumServer server) {

**说明：**调用具体接口之前,初始化SDK.

**参数：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可为空** |
| ctx | 应用全局content | Context | 应用全局content | 不可空 |
| projectId | 项目编号 | String | 由第三方开发人员到e签宝开放平台通过创建项目获得 | 不可空 |
| projectSecret | 项目密钥 | String | 由第三方开发人员到e签宝开放平台通过创建项目获得 | 不可空 |
| server | 服务器环境 | EnumServer | e签宝对应服务器地址  第三方应用在开发阶段可选择"模拟环境"，待正式上线时,再由开发人员改为正式环境。 | 不可空 |

调用方式参考如下代码:

//初始化SDK  
 TESeal.getInstance().init(getApplicationContext(), "1111563562", "612c09869b978d3f642242c79439aaaa", EnumServer.simulation)

# SDK个性化界面设置(可选项)

可设置页面标题颜色,提示文字,手绘的颜色.

//对SDK做个性化设置  
Custom custom = new Custom();  
//设置标题的背景颜色  
custom.setTitleBackgroundColor(R.color.*black*);  
//设置标题的文字颜色  
custom.setTitleTextColor(R.color.*red*);  
//设置标题文本  
custom.setTitleText(R.string.*sign\_title*);  
//设置手绘签名前的提示语  
custom.setTipMsg("自定义提示语");  
//设置提示语对齐方向  
custom.setTipMsgGravity(EnumTipMsgGravity.center);  
//设置手绘的签署提供的颜色,只能是红,黑,蓝  
custom.getHandColor().add(EnumHandSignColor.Blue);  
custom.getHandColor().add(EnumHandSignColor.Black);  
custom.getHandColor().add(EnumHandSignColor.Red);  
  
//快捷签弹出式对话框的样式  
DialogCustom dialogCustom = new DialogCustom();  
//取消按钮样式  
dialogCustom.setCancelResourceId(R.drawable.*tsign\_quick\_sign\_dialog\_cancel*);  
//全屏按钮样式  
dialogCustom.setFullScreenResourceId(R.drawable.*tsign\_quick\_sign\_dialog\_full\_screen*);  
//清除按钮样式  
dialogCustom.setClearResourceId(R.drawable.*tsign\_quick\_sign\_dialog\_clear*);  
//确认按钮样式  
dialogCustom.setConfirmResourceId(R.drawable.*tsign\_quick\_sign\_dialog\_confirm*);  
custom.setDialogCustom(dialogCustom);  
  
TESeal.getInstance().setmCustom(custom);

# 创建快捷签印章

原型:public void quickToSign(Activity content, String id, EnumWindow windowOption, final Callback callback) {

**说明:**创建快捷手绘签名的图片,返回图片路径.

**参数:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **参数名称** | **类型** | **参数说明** | **是否可为空** |
| content | activity的content | Activity | activity的content | 不可空 |
| id | 自定义标识 | String | 自定义标识 ,在回调接口中会原样返回 | 可空 |
| windowOption | 窗口选项 | EnumWindow | 全屏或者小窗口 |  |
| callback | 回调 | Callback | 回调的接口. | 不可空 |

**(**callback**)回调说明**: sdk以回调的方法向第三方返回数据.

|  |  |  |
| --- | --- | --- |
| 回调方法 | 方法说明 | 参数说明 |
| didSignBackImage | 手绘成功时回调 | id:自定义标识  sealImagePath: 手绘印章的图片路径 |
| onError | 各种原因出错时回调 | 错误信息 |
| onCancel | 用户取消操作 |  |

**调用举例**

//快捷签 弹出框  
mBtnQuickToSign.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 TESeal.getInstance().quickToSign(Main2Activity.this, "", EnumWindow.Dialog, new Callback() {  
 @Override  
 public void didSignBackImage(String id, String sealImagePath) {  
 tvLog.setText(String.*format*("id=%s\r\nsealImagePath=%s", id, sealImagePath));  
 loadImage(image, sealImagePath);  
 }  
 @Override  
 public void onError(JSONObject json) {  
 tvLog.setText("手绘签名失败:" + JSONUtils.getString(json, "msg", ""));  
 }  
 @Override  
 public void onCancel(JSONObject json) {  
 tvLog.setText("取消手绘签名");  
 }  
 });  
 }  
});