# iAppPDF

Version 4.1.0.120

# 技　术　白　皮　书

江西金格科技股份有限公司

地址：江西省南昌市高新区火炬大街 579 号绿悦科技大厦 15 楼
邮编： 330096
网址： http://www.kinggrid.com
电话： 0791-88108630
服务： 400-6776-800

# 目 录

# 一、 开发背景

随着android的不断普及，支持android的移动设备日趋多样化，如手机、平板电脑、相机、电视等，而android平台上类似PC端的对PDF文件的批注才刚起步。对此我们开发了一个基于Android平台的iAppPDF在线管理中间件应用。软件开发公司可以利用该方案，与自己研制的基于android平台的业务系统相结合，开发出真正符合业务需求的软件。

# 二、 软件特点

1. 实现了在 Android 平台上 PDF 文档和国产化 OFD 文档的阅读。
2. 实现了通过 JAR+SO 库的形式给第三方进行调用，能实现自定义 UI。
3. 实现了在 Android 上对 PDF 文件进行打开、保存、阅读、缩放、手写签名、手写批注、全文批注、插入图片、证件拍照及照片裁剪等功能。

# 三、 技术特点

1. 在原生应用中集成开发，稳定性、兼容性、安全性好。
2. 采用 JAR+SO 库的方式提供，方便进行功能扩展以及 UI 定制。

# 四、 运行环境

**1）硬件平台：**

手机/PAD：CPU 建议双核 1G 以上，并采用了支持 OpenGL ES 的 GPU，ROM 4G 及以上，RAM 1G 及以上，屏幕尺寸：7 英寸电容式多点触摸屏，支持 WiFi，支持后置摄像头，设备必须采用原厂 ROM 固件。

服 务 器： 配置相应级别的 PC 服务器。

**2）软件平台：**

操作系统： Android

应用平台： IIS、Tomcat、WebLogic、WebSphere、Domino 等

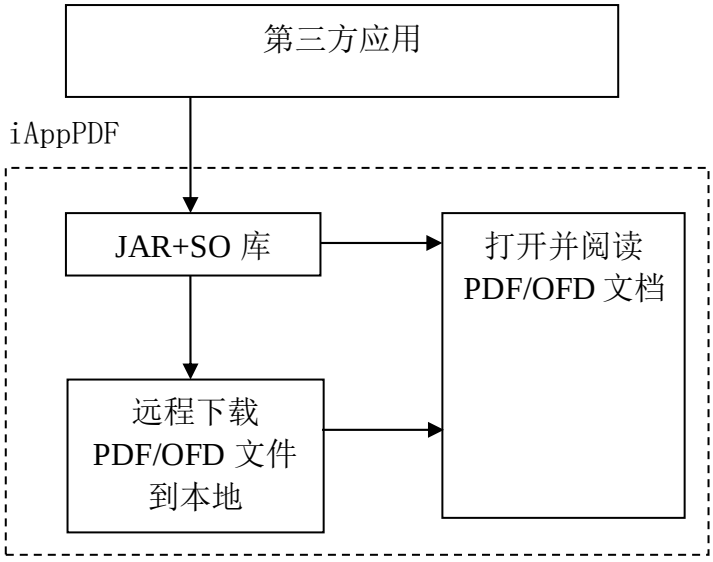
数 据 库： SQLServer2000、MySql、Oracle、DB2、Sybase 等

手机/PAD： Android4.0 及以上

**3）推荐设备：**

| 设备厂商 | 设备型号 |
|---|---|
| 三星 | GT-N5100\5110 |
| 三星 | GT-N8010 |
| E 人 E 本 | T7 |
| E 人 E 本 | T8 |
| E 人 E 本 | T9 |

# 五、 体系架构

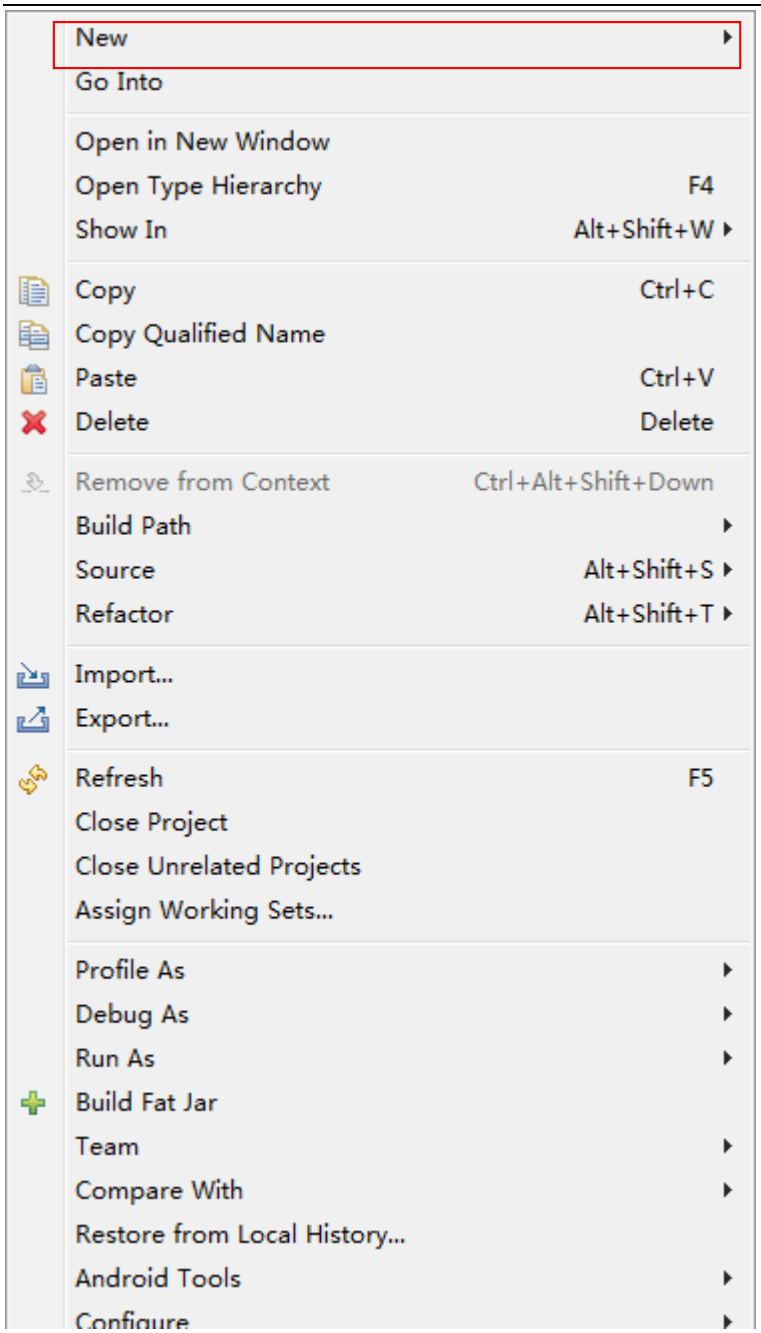


# 六、 集成步骤说明

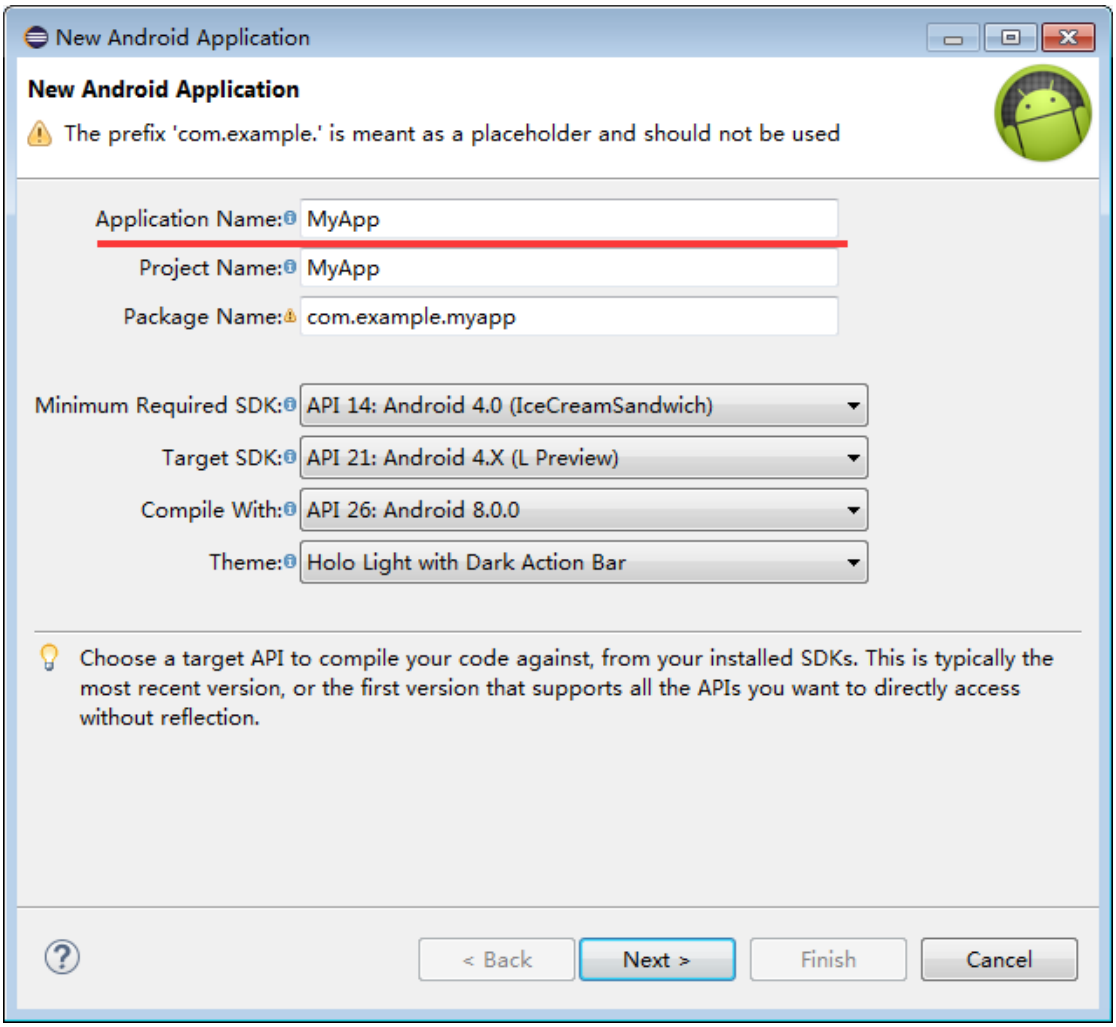
**以 Android 开发编译工具 Eclipse 为例，步骤如下：**

## 第一步：新建项目

打开 Eclipse，在 Package Explorer 的空白区域，点击鼠标右键，弹出菜单，选中之前新建的项目，然后点击鼠标右键，弹出菜单之后，选择【New】选项下的【Android Application Project】，如下图所示：




在弹出的对话框中做如下操作：在【Application Name】框中填入你应用的名字，在【Project Name】框中填入你项目工程的名字，在【Package Name】中填入你项目的包名。在【Minimum Required SDK】下拉列表中选择【API 14: Android 4.0】，这一项表明你的应用最低支持的 Android 版本，建议设置为【API 14】。其余几项可随意。如下图所示：

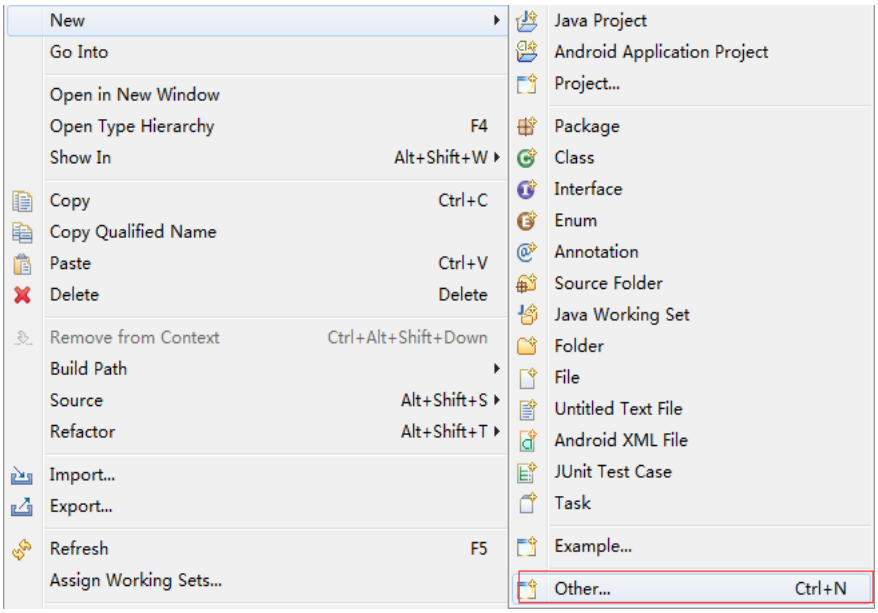


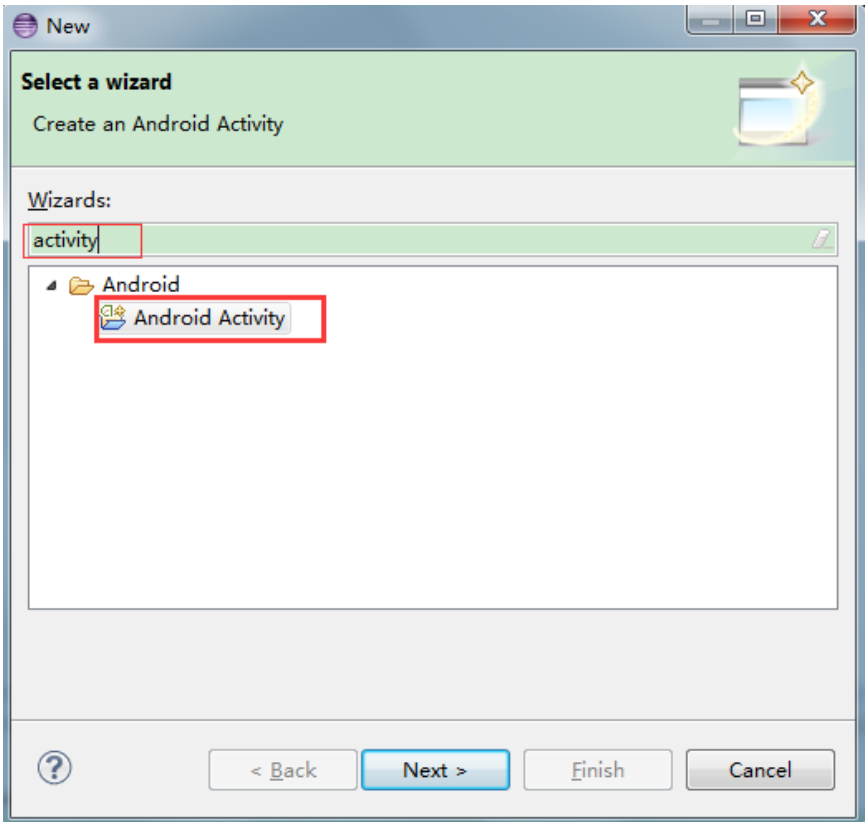

单击【Next】按钮，接下来会弹出一个对话框，接着单击【Next】按钮，一直按【Next】按钮，直到【Next】按钮变灰，【Finish】按钮变亮。然后单击【Finish】按钮。

以上操作就已经完成了创建一个你的 Android 项目工程。

## 第二步：创建 Activity

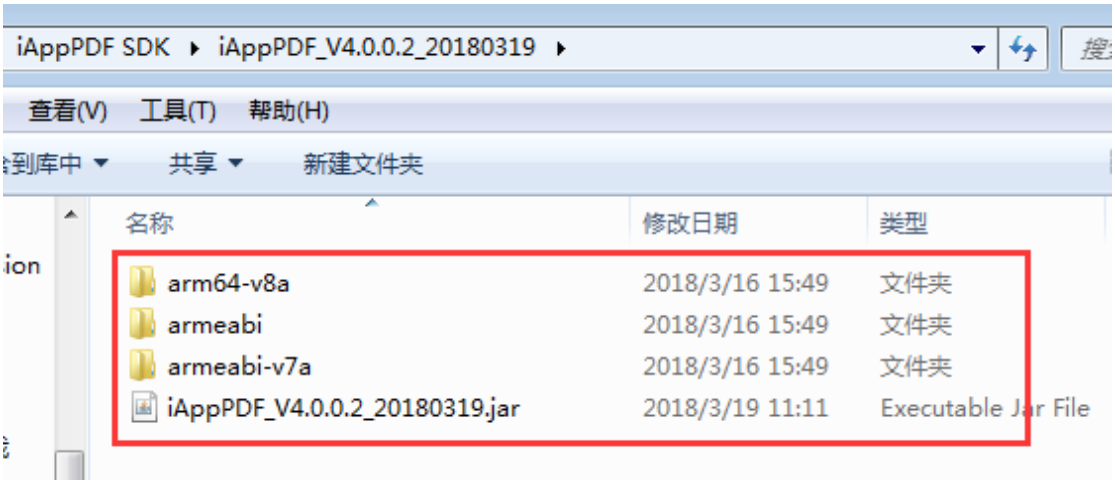
一般情况下在创建工程的时候在 src 目录下会默认生成 MainActivity.java，则可以忽略此步。如果默认没有，可以通过手动创建的方式来创建 MainActivity.java。步骤：右击 src 目录 --> New 选项 --> Other 选项 --> 搜索 activity --> 选择 Android Activity --> 一直选击 Next --> 填写名称（如：Activity Name：MainActivity; Layout Name（布局文件名称）:activtiy_main） --> 完成，此种方式同样适用于新增其他 Activity，如下图：



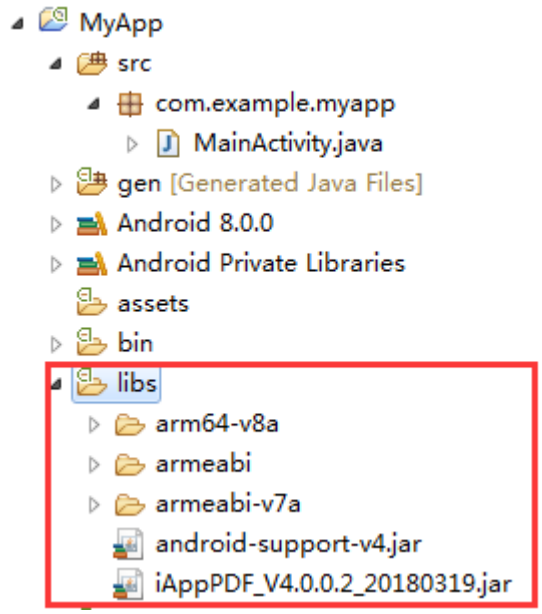




# 第三步：复制 iAppPDF SDK 到项目中

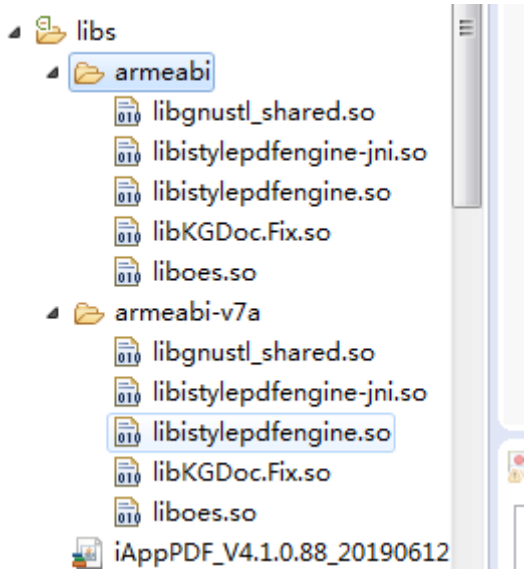
解压产品包，打开【iAppPDF SDK】文件夹下的【iAppPDF_V4.X.X.X_XXXXXXXX】如下图所示：



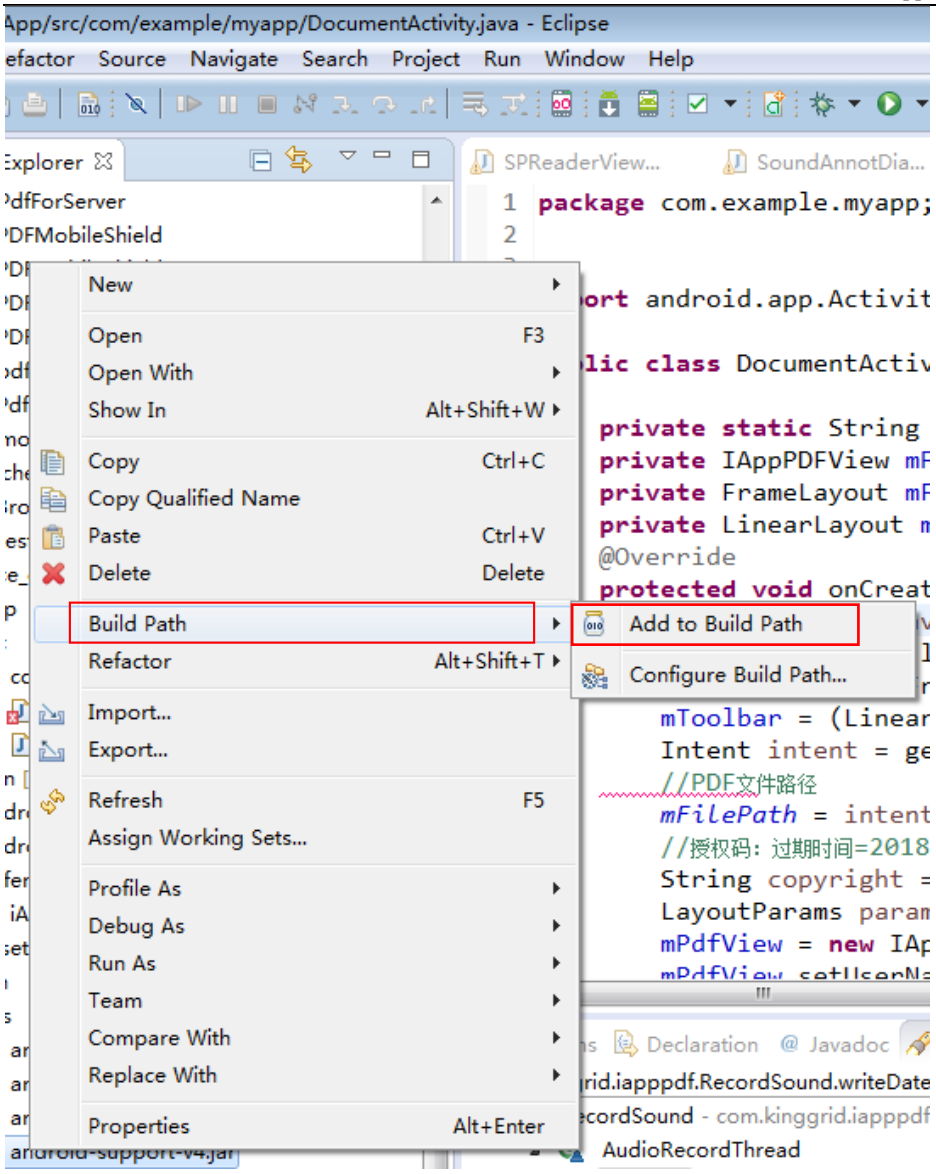

复制红框中的内容。打开 Eclipse，复制到你项目的【libs】文件夹下，如果没有【libs】文件夹，可右击你的项目名称，选择【New】下的【Folder】新建 libs 文件夹。如下图所示：




注：自 iAppPDF_V4.1.XX 版本之后 armeabi 和 armeabi-v7a 包含 PDF 的 so 库（libistylepdfengine.so 和 libistylepdfengine-jni.so）和 OFD 的 so 库（libgnustl_shared.so, libKGDoc.Fix.so, liboes.so）如下图。因为 so 库的体积较大，如果您的项目中只需要集成 PDF 文档的阅读编辑等功能，则可以去除 OFD 的 so 库（libgnustl_shared.so, libKGDoc.Fix.so, liboes.so）。反之，如果您的项目中只需要集成 OFD 文档的阅读编辑等功能，则可以去除 PDF 的 so 库（libistylepdfengine.so 和 libistylepdfengine-jni.so）。如果项目中要同时支持 PDF 文档和 OFD 文档，则需要添加 PDF 的 so 库和 OFD 的 so 库。其他 so 库如无必要，无需添加。




右击【iAppPDF_V4.X.X.X_XXXXXXXX.jar】jar 包，选择【Build path】下的【Add to Build Path】，如下图所示：




# 第四步：完善 PDF 启动界面

创建 PDF 启动的 activity 之后，需要完成 activity 里相关方法的实现，具体实现可参考 iAppPDFDemo 中 DocumentActivity.java。其中必须实现 onCreate()方法，实现方法参考下面代码。可直接将下面代码复制到你的 MainActivity.java 的 onCreate 方法中。DocumentActivity.java 中使用到的 book.xml 布局文件也可从 iAppPDFDemo 中的 res\layout\文件夹下复制到你项目中的相同目录下。

启动 PDF 视图界面的方法实现：

```
/**
     * 在onCreate方法中初始化控件
     */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.book);
        mFrameLayout = (FrameLayout)
ffindViewById(R.id.book_frame);
        mToolbar = (LinearLayout) findViewById(R.id.toolbar);
        Intent intent = getIntent();
        //PDF 文件路径（传入 SD 卡中 PDF 的路径，如:“/sdcard/xxx.pdf”）
        //表示是根目录下的 xxx.pdf 文件
        //如果是从 iAppPDFDemo 中复制过来的 DocumentActivity.java,
        //那么 mFilePath 需要设置一个确切的 PDF 文档路径才可打开文档
        //如: mFilePath = “/sdcard/xxx.pdf”;
        mFilePath = intent.getStringExtra("filepath");
        LayoutParams params = new
LayoutParams(LayoutParams.MATCH_PARENT,
LayoutParams.MATCH_PARENT);
        //PDF 视图控件
        mPdfView = new IAppPDFView(this);
        mPdfView.setUserName("admin");
        mPdfView.setCopyRight(copyright);
        //使用 PDF 视图实例调用打开文档的方法
        mPdfView.openDocument(mFilePath);
        //将 PDF 视图控件添加到布局文件的控件中，完成 PDF 界面的显示
        mFrameLayout.addView(mPdfView);

        }
```

## 第五步：完善 AndroidManifest.xml 文件

找到工程根目录下的 AndroidManifest.xml 文件，每个工程对应一个该文件，编辑此文件：

a） 添加相关权限

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

b）配置 activity 标签

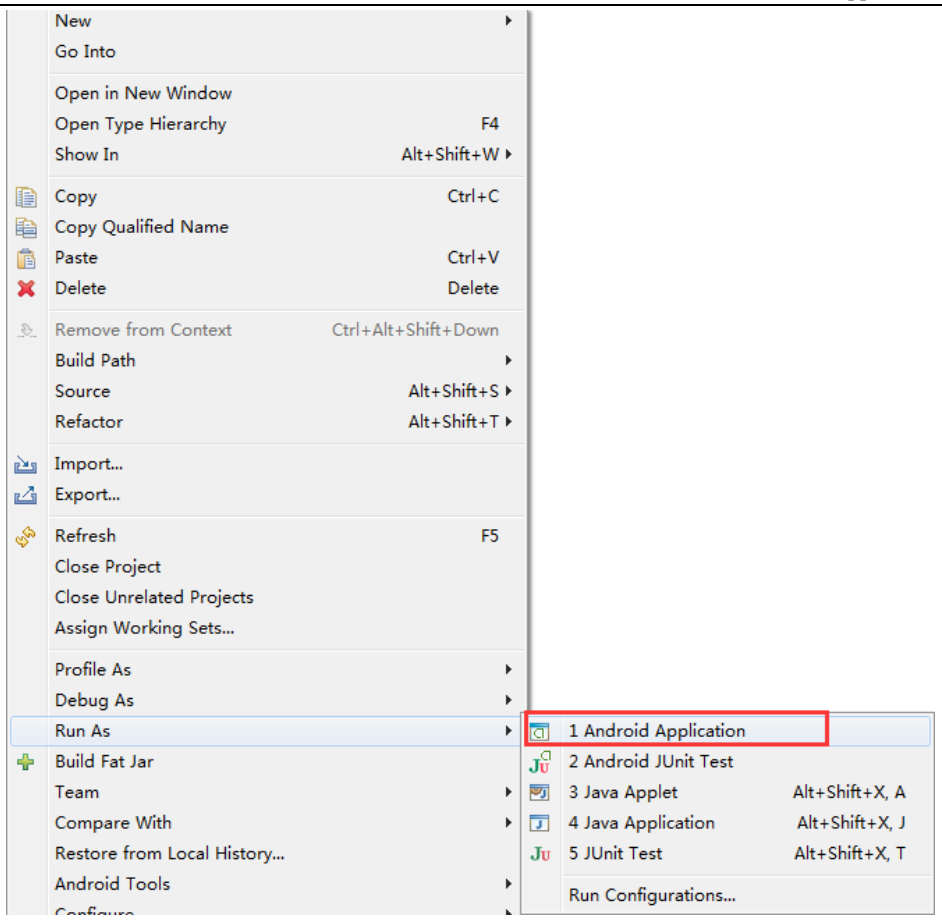
```
<activity
    android:name="com.kinggrid.iappreaderdemo.MainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

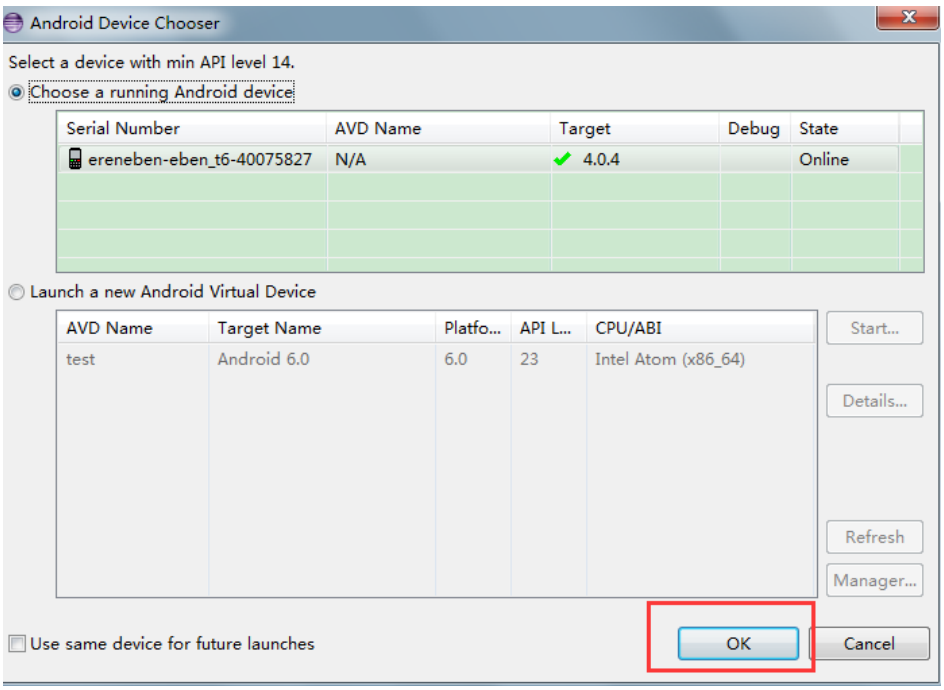
如果你使用的是从 iAppPDFDemo 中复制过来的 DocumentActivity.java，则需要将 MainActivity 改为 DocumentActivity。前面的包名参照你自己项目工程的包名。

# 第六步：部署运行

连接 Andorid 设备，打开 DDMS，可以查看到当前连接的设备。确保当前已有设备连接的情况下，将当前项目部署到设备上。步骤：右击工程 --> Run As 选项 --> Android Application 选项 --> 选择设置 --> 点击 OK，然后在设备上查看。

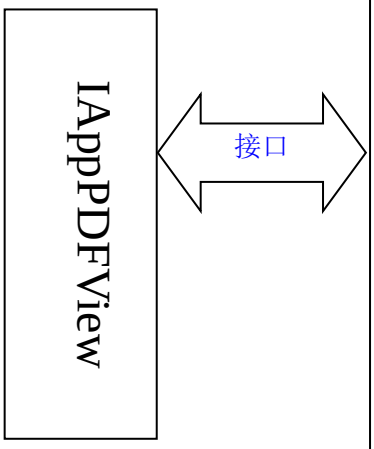
如下图：


New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
Refresh F5
Close Project
Close Unrelated Projects
Assign Working Sets...
Profile As
Debug As
Run As
Build Fat Jar
Team
Compare With
Restore from Local History...
Android Tools
1 Android Application
2 Android JUnit Test
3 Java Applet Alt+Shift+X, A
4 Java Application Alt+Shift+X, J
5 JUnit Test Alt+Shift+X, T
Run Configurations...

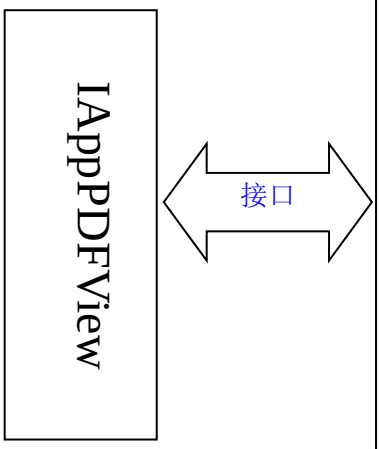

Android Device Chooser
Select a device with min API level 14.
Choose a running Android device
Serial Number
AVD Name
Target
Debug
State
ereneben-eben_t6-40075827
N/A
4.0.4
Online
Launch a new Android Virtual Device
AVD Name
Target Name
Platfo...
API L...
CPU/ABI
test
Android 6.0
6.0
23
Intel Atom (x86_64)
Start...
Details...
Refresh
Manager...
Use same device for future launches
OK
Cancel

# 七、 接口说明

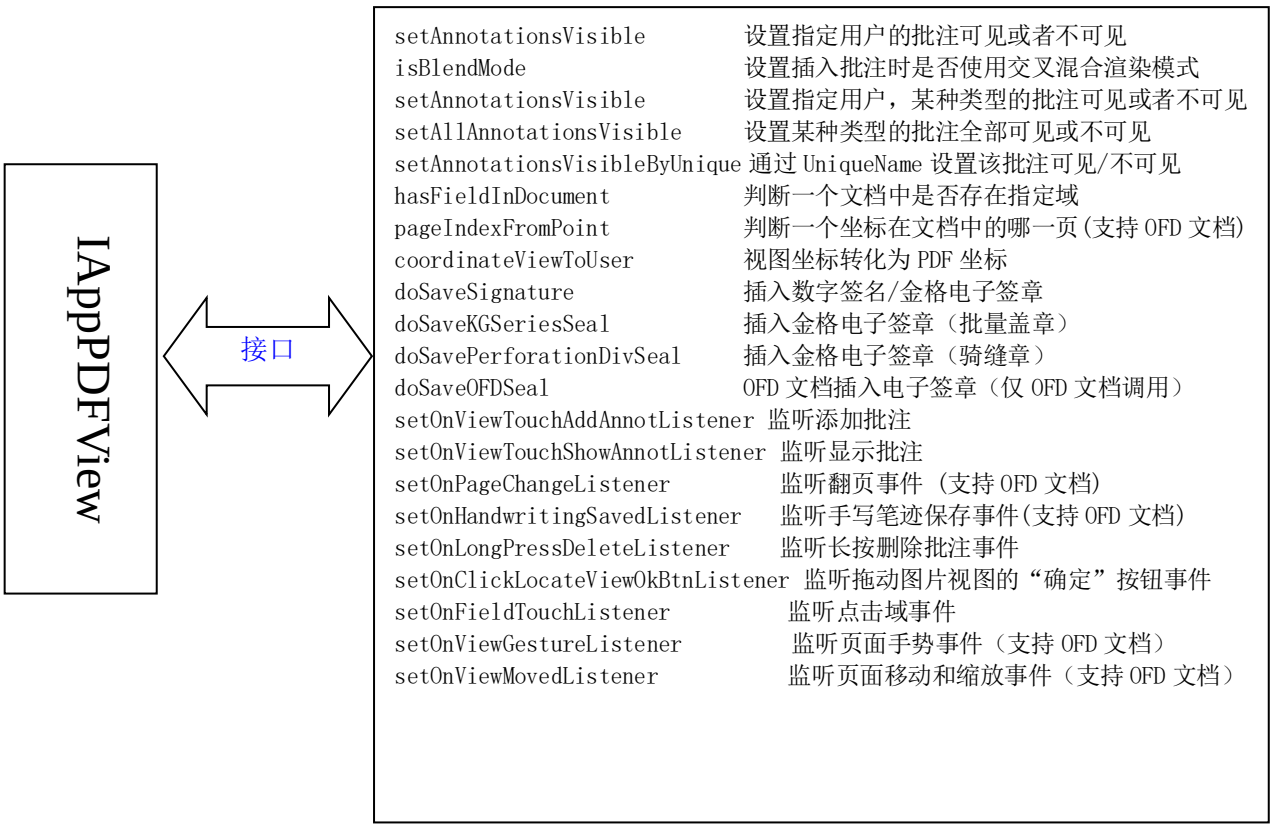
## 7.1 IAppPDFView 对外接口说明




| | |
|---|---|
| setUserName | 设置用户名(支持OFD文档) |
| setCopyRight | 设置授权码(支持OFD文档) |
| setOfflineUrl | 设置离线注册地址(支持OFD文档) |
| setSupportEbenT7Mode | 设置是否是E人E本设备(支持OFD文档) |
| setVectorSign | 设置手写数据是否以矢量方式保存 |
| openDocument | 打开文档(通过文件路径打开)(支持OFD文档) |
| openDocument | 打开文档（通过字节流打开） |
| openAuthenticateDocument | 打开加密文档 |
| openDocumentWithFont | 打开文档并指定加载文档所用字体的路径 |
| saveDocument | 保存文档(支持OFD文档) |
| decryptionDocument | 解密文档 |
| encryptionDocument | 加密文档 |
| refreshDocument | 刷新页面(支持OFD文档) |
| isDocumentModified | 文档是否修改(支持OFD文档) |
| saveAsDocument | 文档另存为(支持OFD文档) |
| closeDocument | 关闭文档(支持OFD文档) |
| getDocType | 获取文档类型（支持OFD文档） |
| getOutlineList | 获取大纲列表(支持OFD文档) |
| gotoOutlineItem | 跳转到目录大纲指定页(支持OFD文档) |
| getCustomtagList | 获取语义树列表（仅支持OFD文档） |
| gotoCustomtagItem | 跳转到语义位置并高亮语义文本（仅支持OFD） |
| searchText | 开始搜索文字(支持OFD文档) |
| searchTextNext | 搜索下一个文字(支持OFD文档) |
| searchTextPrevious | 搜索上一个文字(支持OFD文档) |
| stopSearchText | 停止搜索文字(支持OFD文档) |
| getPageCount | 获取总页数(支持OFD文档) |
| getCurrentPageNo | 获取当前页码(支持OFD文档) |
| jumpToPage | 跳转至指定页(支持OFD文档) |
| setPageZoom | 设置页面放缩值(支持OFD文档) |
| getPageZoom | 获取页面放缩值(支持OFD文档) |
| setMaxPageZoom | 设置页面最大缩放值（支持OFD文档） |
| setZoomState | 设置页面是否允许缩放（支持OFD文档） |
| setDoubleClickZoomValue | 设置双击放大页面的缩放值（支持OFD文档） |
| refreshPage | 刷新指定页(支持OFD文档) |
| getViewTopLimit | 视图可移动到顶部的最大距离(支持OFD文档) |
| getViewBottomLimit | 视图可移动到底部的最大距离(支持OFD文档) |
| getViewLeftLimit | 视图可移动到左边的最大距离(支持OFD文档) |
| getViewRightLimit | 视图可移动到右边的最大距离(支持OFD文档) |
| pdfViewScrollTo | 使视图滚动到指定位置(支持OFD文档) |
| getPDFViewScrollX | 获取视图滚动的x轴位置(支持OFD文档) |
| getPDFViewScrollY | 获取视图滚动的y轴位置(支持OFD文档) |
| addBlankPage | 添加空白页 |
| pagesToBitmaps | 所有页导出成Bitmap(支持OFD文档) |
| pageToBitmap | 将指定页导出成Bitmap(支持OFD文档) |
| openLocateSignature | 打开可拖动图片视图 |
| openLocateSignature | 打开可拖动图片视图，并制定是否显示按钮 |
| addField | 在文档中添加域 |
| insertHandWriteAnnotation | 插入指定手写批注(支持OFD文档) |
| insertTextAnnotation | 插入文字注释 |
| insertSignatureInField | 插入图片到图章域中 |
| insertSignatureInText | 插入图片到文本域中 |
| insertPhotoIntoPDF | 插入图片到PDF文档中 |

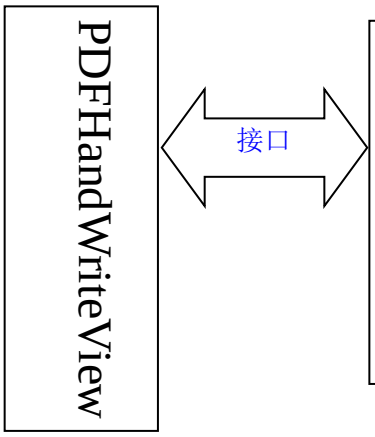



| | |
|---|---|
| insertSoundAnnotation | 插入指定语音批注 |
| insertVectorAnnotation | 插入矢量手写批注 |
| openHandWriteAnnotation | 调用全文手写批注(支持OFD文档) |
| setKgHandwriteViewEraserAnnot | 设置全文签批的橡皮擦模式时可擦除批注 |
| openTextAnnotation | 添加文字批注 |
| openSoundAnnotation | 添加语音批注 |
| openAreaHandwrite | 打开区域手写视图(支持OFD文档) |
| closeAreaHandwrite | 关闭区域手写视图(支持OFD文档) |
| closeAreaHandwrite | 关闭并保存区域手写笔迹(支持 OFD 文档) |
| doSaveSoundAnnot | 保存语音批注 |
| doSaveTextAnnot | 保存文字批注内容 |
| doSaveTextAnnot | 保存文字批注内容并自定义文字图标大小 |
| doSaveFreeTextAnnot | 保存打字机批注(支持 OFD 文档) |
| doSaveHandwriteInfo | 保存手写内容(支持 OFD 文档) |
| doSaveSignByFieldName | 域定位签名 |
| openEBHandwriteView | 打开E人E本手写视图(支持OFD文档) |
| doSaveEBHandwriteInfo | 保存E人E本的手写内容(支持OFD文档) |
| doCloseEBHandwriteInfo | 关闭E人E本的手写视图(支持OFD文档) |
| setEBPenDeleteAnnot | 设置EB手写状态时，笔头可擦除批注 |
| openHandwriting | 打开通用版手笔分离 |
| doSaveHandwriting | 保存通用版手笔分离的手写笔迹 |
| undoHandwriting | 通用版手笔分离笔迹撤销 |
| redoHandwriting | 通用版手笔分离笔迹恢复 |
| isCanUndo | 通用版手笔分离笔迹是否可撤销 |
| isCanRedo | 通用版手笔分离笔迹是否可恢复 |
| isHaveHandwriting | 判断通用版手笔分离是否有笔迹 |
| doClearHandwriting | 清除通用版手笔分离的手写笔迹 |
| doCloseHandwriting | 退出通用版手笔分离 |
| doEraserHandwriting | 通用版手笔分离橡皮擦功能 |
| isEraser | 通用版手笔分离是否在擦除状态 |
| doCloseHandwriteInfo | 关闭全文签批手写视图 |
| doDeleteSoundAnnot | 删除语音批注 |
| doDeleteTextAnnotation | 删除文字批注 |
| doUpdateTextAnnotation | 更新文字批注 |
| doDeleteFreeTextAnnotation | 删除打字机批注 |
| doUpdateFreeTextAnnotation | 更新打字机批注 |
| doSaveLocalDigitallySign | 插入数字签名（使用本地证书做数字签名） |
| getAnnotationNewList | 获取当前文档指定用户，类型的批注列表 |
| getAnnotationList | 获取当前文档中指定类型的批注列表 |
| getAnnotationList | 获取当前文档中指定的多种类型的批注列表 |
| setAllAnnotationOnlyRead | 设置文档所有批注是否锁定 |
| setAnnotationOnlyRead | 设置某一类文档批注是否是否锁定 |
| setCustomType | 设置用户自定义批注类型 |
| deleteAllAnnotations | 删除当前用户的指定类型的所有批注 |
| deleteAllStampAnnot | 删除 PDF 文档中所有的图片/矢量批注 |
| deleteAnnotWithId | 删除指定 annotID 的批注 |
| deleteAnnotWithUniqueName | 删除指定 uniqueName 的批注 |
| updateAnnotRect | 移动批注 |
| setCanMoveAnnot | 设置是否可以移动批注 |
| updateTextAnnotContent | 修改文字批注内容 |
| getFieldByName | 通过域名得到指定的域 |
| setFieldContent | 设置指定域的内容 |
| getFieldContent | 获取指定名称的域的内容 |
| getAllFieldContent | 获取所有编辑域内容 |
| setFormFillEnabled | 设置是否允许表单填充 |
| setHighlightField | 设置域是否高亮 |
| hideLongPressRect | 是否隐藏长按删除批注的矩形框(支持 OFD 文档) |
| hideAnnotToast | 是否隐藏点击批注出现的 toast（支持 OFD 文档） |



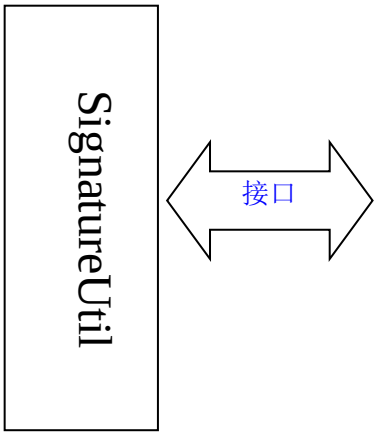

注：以上接口如有“支持 OFD 文档”字样表明该接口支持 PDF 和 OFD 文档调用。如果接口没有标明“支持 OFD 文档”，则表明该接口目前只支持 PDF 文档调用。后续会开放支持 OFD 文档。

## 7.2 PDFHandWriteView 对外接口说明




| | |
|---|---|
| cancelEarseHandwriteInfo | 退出擦除模式 |
| canSave | 是否能够保存，即判断屏幕是否为空 |
| doClearHandwriteInfo | 清除已签笔迹 |
| doEarseHandwriteInfo | 擦除已签笔迹 |
| doRedoHandwriteInfo | 撤销一笔已回退笔迹 |
| doSettingHandwriteInfo | 设置手写信息 |
| doUndoHandwriteInfo | 回退一笔已签笔迹 |
| getEarseState | 判断是否处于擦除模式 |
| resetPenSetting | 重置手写设置为默认设置 |
| setCopyRight | 设置授权码 |
| setPenInfo | 设置手写的笔宽、颜色、笔型等 |

## 7.3 SignatureUtil 对外接口说明




| | |
|---|---|
| getInstance | 获取SignatureUtil实例 |
| sign | 签名 |
| setPFXInfo | 设置本地证书路径和本地证书密码 |
| getPFXPassword | 获取pfx文件的密码 |
| getSubjectName | 获取相对应的证书使用者名称 |
| getCert | 获取证书 |
| setMode | 设置签名方式 |
| getMode | 获取签名方式 |
| setSignType | 设置签名类型（数字签名或者电子签章） |
| getSignType | 获取签名类型（数字签名或者电子签章） |
| getKGHash | 获取文档哈希 |

# 八、 第三方应用调用说明

### 8.1.1 IAppPDFView 接口调用说明

**1. 实例一个 IAppPDFView 对象，将这个实例添加到控件中，完成 PDF/OFD 界面显示**

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    setContentView(R.layout.book);
    mFrameLayout = (FrameLayout) findViewById(R.id.book_frame);
    mToolbar = (LinearLayout) findViewById(R.id.toolbar);
    Intent intent = getIntent();
    //PDF 文件路径
    mFilePath = intent.getStringExtra("filepath");
    LayoutParams params = new LayoutParams(LayoutParams.MATCH_PARENT,
        LayoutParams.MATCH_PARENT);
    mPdfView = new IAppPDFView(this);
    mPdfView.setUserName("admin");
    mPdfView.setCopyRight(copyright);
    mPdfView.openDocument(mFilePath);
    mFrameLayout.addView(mPdfView);
}
```

**2. 设置基本参数**

```
    mPdfView.setUserName("admin");
    mPdfView.setCopyRight(copyright);
```

## 8.1.2 IAppPDFView 接口方法说明

setUserName(String userName)

功能说明： 设置用户名

输入参数： userName String类型 //用户名

输出参数： 无

特别说明： 无

调用示例： mPdfView.setUserName(“admin”);

setCopyRight(String copyRight)

功能说明： 设置授权码

输入参数： copyRight String类型 //授权码，由金格提供

输出参数： 无

特别说明： 无

调用示例： mPdfView.setCopyRight(“”);

setOfflineUrl(String url)

功能说明： 设置离线注册地址

输入参数： url String类型 //离线注册地址

输出参数： 无

特别说明： 在openDocument方法之前调用该方法

调用示例： mPdfView.setOfflineUrl(“192.168.0.4:8080”);

setSupportEbenT7Mode(boolean isSupportEbenT7Mode)

功能说明： 设置是否是E人E本设备

输入参数： isSupportEbenT7Mode boolean类型 //true：是，false：不是

输出参数： 无

特别说明： 无

调用示例： mPdfView.setSupportEbenT7Mode(true);

setVectorSign(boolean isVectorSign)

功能说明： 设置手写数据是否以矢量方式保存

输入参数： isVectorSign boolean类型 //true：是，false：不是

输出参数： 无

特别说明： 无

调用示例： mPdfView.setSupportEbenT7Mode(true);

openDocument(String pdfFilePath)

功能说明： 打开文档

输入参数：pdfFilePath String类型 //PDF/OFD文档路径(“/sdcard/test.pdf”)(“./sdcard/test.ofd”)

输出参数： int类型：0：打开文档成功

特别说明： 无

调用示例： mPdfView.openDocument(“/sdcard/test.pdf”);

mPdfView.openDocument(“/sdcard/test.ofd”);

openDocument(byte[] data)

功能说明： 打开文档

输入参数： data byte数组类型 //PDF字节流

输出参数： int类型：0：打开文档成功

特别说明： 无

调用示例： byte[] data = getBytes(mFilePath);

mPdfView.openDocument(data);

openAuthenticateDocument(String pdfFilePath, String password)

功能说明： 打开加密文档

输入参数： pdfFilePath String类型 //PDF文件路径

password String类型 //PDF文件密码

输出参数： int类型：0：打开文档成功

特别说明： 无

调用示例： mPdfView.openAuthenticateDocument(“/sdcard/test.pdf”, “password”);

openDocumentWithFont(String pdfFilePath, String[] fontPaths)

功能说明： 打开文档，并指定加载文档所用字体的路径

输入参数： pdfFilePath String类型 //PDF文件路径

fontPaths String[]类型 //字体文件路径

输出参数： int类型：0：打开文档成功

特别说明： 无

调用示例： mPdfView.openDocumentWithFont(“/sdcard/test.pdf”, new String[]{“/sdcard/simsun.ttf”});

saveDocument()

功能说明： 保持文档

输入参数： 无

输出参数： int类型：0：保存文档成功

特别说明： 保存文档成功后会发送“com.kinggrid.iapppdf.broadcast.savepdf”的广播

调用示例： mPdfView.saveDocument();

decryptionDocument()

功能说明： 解密文档

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.decryptionDocument();

encryptionDocument(String pwd)

功能说明： 加密文档

输入参数： pwd String类型 //加密文档的密码

输出参数： boolean类型：true：加密文档成功， false：加密文档失败

特别说明： 无

调用示例： mPdfView.encryptionDocument(“123456”);

refreshDocument()

功能说明： 刷新整篇文档

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.refreshDocument();

isDocumentModified()

功能说明： 文档是否被修改

输入参数： 无

输出参数： boolean类型：true：文档已被修改，false：文档未修改

特别说明： 无

调用示例： mPdfView.isDocumentModified();

saveAsDocument(String savePath)

功能说明： 文档另存为

输入参数： savePath String类型 //文档另存的路径

输出参数： int类型：0：另存文档成功

特别说明： 无

调用示例： mPdfView.saveAsDocument("/sdcard/another.pdf");

closeDocument()

功能说明： 关闭文档

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.closeDocument();

getOutlineList()

功能说明： 获取大纲列表

输入参数： 无

输出参数： ArrayList<OutlineLink> //OutlineLink对象集合

特别说明： 无

调用示例： mPdfView.getOutlineList();

gotoOutlineItem(OutlineLink link)

功能说明： 跳转到目录大纲指定页

输入参数： link OutlineLink对象类型 //传入OutlineLink对象

输出参数： 无

特别说明： 无

调用示例： mPdfView.gotoOutlineItem(link);

getCustomtagList()

功能说明： 获取语义树列表（仅支持OFD文档）

输入参数： 无

输出参数： ArrayList<OutlineLink> //OutlineLink对象集合

特别说明： 无

调用示例： mPdfView.getCustomtagList();

gotoCustomtagItem(OutlineLink link)

功能说明： 跳转到语义位置并高亮语义文本（仅支持OFD文档）

输入参数： link OutlineLink对象类型 //传入OutlineLink对象

输出参数： 无

特别说明： 无

调用示例： mPdfView.gotoCustomtagItem(link);

searchText(String content, int highLightColor)

功能说明： 开始搜索文本

输入参数： content String类型 //需要搜索的文字

highLightColor int类型 //文字高亮颜色

输出参数： ArrayList<RectF> //搜索到的文字的矩形区域集合

特别说明： 如果highLightColor参数传0，表示使用默认的高亮颜色

调用示例： mPdfView.searchText(“点击”, Color.parseColor("#ff33cc"));

searchTextNext()

功能说明： 搜索下一个文本

输入参数： isLoopSearch boolean类型 //是否循环搜索

输出参数： int //当前搜索文本的索引

特别说明： 无

调用示例： int searchTextIndex = mPdfView.searchTextNext(false);

searchTextPrevious()

功能说明： 搜索上一个文本

输入参数： isLoopSearch boolean类型 //是否循环搜索

输出参数： int //当前搜索文本的索引

特别说明： 无

调用示例： int searchTextIndex = mPdfView.searchTextPrevious(false);

stopSearchText()

功能说明： 停止搜索文本，并去除文本的高亮

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.stopSearchText();

getPageCount()

功能说明： 获取文档总页数

输入参数： 无

输出参数： int类型

特别说明： 无

调用示例： int count = mPdfView.getPageCount();

getCurrentPageNo()

功能说明： 获取文档当前页的页码

输入参数： 无

输出参数： int类型

特别说明： 无

调用示例： int currPage = mPdfView.getCurrentPageNo();

jumpToPage(int pageIndex)

功能说明： 跳转至指定页

输入参数： pageIndex int类型 //跳转页的页码

输出参数： 无

特别说明： 无

调用示例： mPdfView.jumpToPage(int pageIndex);

setPageZoom(float zoom)

功能说明： 设置页面缩放值

输入参数： zoom float类型 //页码缩放值

输出参数： 无

特别说明： 缩放值最大为3，最小为1

调用示例： mPdfView.setPageZoom(float zoom);

getPageZoom()

功能说明： 获取页面缩放值

输入参数： 无

输出参数： float类型

特别说明： 缩放值最大为3，最小为1

setMaxPageZoom(float zoom)

功能说明： 设置页面最大放缩值

输入参数： zoom float类型 //最大缩放值

输出参数： 无

特别说明： 缩放值默认最大为3，最小为1

调用示例： mPdfView.setMaxPageZoom(6.0f);

setZoomState(boolean isZoom)

功能说明： 设置页面是否允许缩放

输入参数： isZoom boolean类型 //true:允许缩放 false:不允许缩放

输出参数： 无

特别说明： 默认页面允许缩放

调用示例： mPdfView.setZoomState(false);

setDoubleClickZoomValue(float zoomValue)

功能说明： 设置双击页面放大的缩放值（默认双击放大页面的值为3）

输入参数： zoomValue float类型 //缩放值

输出参数： 无

特别说明： 默认页面允许缩放

调用示例： mPdfView.setDoubleClickZoomValue(1.5f);

refreshPage(int pageNo)

功能说明： 刷新指定页

输入参数： pageNo int类型 //刷新页的页码

输出参数： 无

特别说明： 页码从0开始，即0为第一页

调用示例： mPdfView.refreshPage(0);//刷新第一页

getViewTopLimit()

功能说明： 获取视图可移动到顶部的最大距离

输入参数： 无

输出参数： float类型

特别说明： 无

调用示例： float topLimit = mPdfView.getViewTopLimit();

### getViewBottomLimit()

功能说明： 获取视图可移动到底部的最大距离

输入参数： 无

输出参数： float类型

特别说明： 无

调用示例： float bottomLimit = mPdfView.getViewBottomLimit();

### getViewLeftLimit()

功能说明： 获取视图可移动到左边的最大距离

输入参数： 无

输出参数： float类型

特别说明： 无

调用示例： float leftLimit = mPdfView.getViewLeftLimit();

### getViewRightLimit()

功能说明： 获取视图可移动到右边的最大距离

输入参数： 无

输出参数： float类型

特别说明： 无

调用示例： float rightLimit = mPdfView.getViewRightLimit();

### pdfViewScrollTo(int x, int y)

功能说明： 使视图滚动到指定位置

输入参数： x int类型 //x轴坐标

y int类型 //y轴坐标

输出参数： 无

特别说明： 无

调用示例： mPdfView.pdfViewScrollTo(0,100);

getPDFViewScrollX()

功能说明： 获取视图滚动的x轴位置

输入参数： 无

输出参数： int类型

特别说明： 无

调用示例： int x = mPdfView.getPDFViewScrollX();

getPDFViewScrollY()

功能说明： 获取视图滚动的y轴位置

输入参数： 无

输出参数： int类型

特别说明： 无

调用示例： int y = mPdfView.getPDFViewScrollY();

addBlankPage(int insertPos)

功能说明： 添加空白页

输入参数： insertPos int类型 //插入位置

输出参数： 无

特别说明： -1为在末尾页添加空白页。0. 1 .2 ...为第pos页之前添加空白页

调用示例： mPdfView.addBlankPage(0);//在第一页之前插入空白页

pagesToBitmaps()

功能说明： 将文档中所有页面导出成Bitmap

输入参数： 无

输出参数： ArrayList<Bitmap>类型

特别说明： 无

调用示例： ArrayList<Bitmap> bitamps = mPdfView.pagesToBitmaps();

pageToBitmap(int pageNo)

功能说明： 将指定页面导出成Bitmap

输入参数： pageNo int类型 //指定页面

输出参数： Bitmap类型

特别说明： 无

调用示例： .Bitmap bitamps = mPdfView.pageToBitmap(0);//将第一页导出成Bitmap

openLocateSignature(Bitmap signatureBitmap, float scale, final int backgroundColor)

功能说明： 打开可拖动视图，展示图片，拖动位置，插入图片

输入参数： signatureBitmap Bitmap类型 //需要展示的图片

scale float类型 //缩放值（默认为0）

backgroundColor int类型 //背景颜色(保留参数，无效)

输出参数： 无

特别说明： 无

调用示例： mPdfView.openLocateSignature(bmp, 0, Color.parseColor("#fff2cc"));

openLocateSignature(Bitmap signatureBitmap, float scale, final int backgroundColor, boolean isShowBtn)

功能说明： 打开可拖动视图，展示图片，拖动位置，插入图片

输入参数： signatureBitmap Bitmap类型 //需要展示的图片

scale float类型 //缩放值（默认为0）

backgroundColor int类型 //背景颜色(保留参数，无效)

isShowBtn boolean类型 //是否显示“保存/取消”按钮

输出参数： 无

特别说明： 无

调用示例： mPdfView.openLocateSignature(bmp, 0, Color.parseColor("#fff2cc"),true);

addField(FieldType type, int pageNo, float[] rect, String fieldName)

功能说明： 在PDF文档中添加域

输入参数： type FieldType枚举类型 //需要添加的域的类型

pageNo int类型 //域所在的页码（从0开始）

rect float数组类型 //域的矩形区域（坐标系为PDF坐标系）

fieldName String类型 //域的名称

输出参数： 无

特别说明： FieldType枚举类型包括：PUSH_BTN, CHECK_BTN, RADIO_BTN, TEXT, COMBOX, LISTBOX, SIGNATURE, UNKNOWN

调用示例： mPdfView.addField(FieldType.TEXT, 0, new float[]{200,200,300,300}, "testF");

insertHandWriteAnnotation(int pageNo, float x, float y,float w, float h, String imagePath, String userName,String createTime, int type, String annotId, String uniqueName)

功能说明： 插入指定手写批注

输入参数： pageNo int类型 //要插入手写批注的页码

x float类型 //x轴坐标位置

y float类型 //y轴左边位置

w float类型 //批注图片宽度

h float类型 //批注图片高度

imagePath String类型 //批注图片路径

userName String类型 //使用者

createTime String类型 //批注图片创建时间

type int类型 //批注类型

annotId String类型 //批注annotId，可传空

uniqueName String类型 //批注唯一名称，可传空

输出参数： 无

特别说明： 图片格式必须是jpg格式，x, y 代表的是pdf文档以左上角为原点的页面坐标而不是屏幕坐标， 插入批注时createTime的时间格式必须是以下格式：

"2014-05-29 11:10:10"不能为其他格式。

调用示例： mPdfView.insertHandWriteAnnotation(0, 20, 20, 200, 100, "/mnt/sdcard/123.jpg", "admin", "2014-05-29 11:10:10");

**insertTextAnnotation(int pageNo, float x, float y,String content, String userName, String createTime)**

功能说明： 插入文字注释

输入参数： pageNo int类型 //要插入文字注释的页码

x float类型 //x轴坐标位置

y float类型 //y轴左边位置

content String类型 //文字注释内容

userName String类型 //使用者

createTime String类型 //文字注释创建时间

输出参数： 无

特别说明： x, y 代表的是pdf文档以左上角为原点的页面坐标而不是屏幕坐标，插入批注时createTime的时间格式必须是以下格式：

"2014-05-29 11:10:10"不能为其他格式。

调用示例： mPdfView.insertTextAnnotation(0, 20, 20, "测试文字批注", "admin", "2014-05-29 11:10:10");

**insertSignatureInField(String imagePath, String fieldName, int pageNo, float scale)**

功能说明： 在指定名称的域的位置上插入一张图片

输入参数： imagePath String类型 //图片路径

fieldName String类型 //域名称

pageNo int类型 //页码

scale float类型 //放缩值

输出参数： 无

特别说明： pageNo 页码，-1表示搜索整篇文档的域名来插入，其他值表示搜索指定页的域名来插入

调用示例： mPdfView.insertSignatureInField(“/sdcard/myImg.png”, “field1”, 0, 0);

**insertSignatureInText(String imagePath, String content, float scale)**

功能说明： 插入图片到指定的文字处

输入参数： imagePath String类型 //图片路径

content String类型 //文字内容

scale float类型 //放缩值

输出参数： 无

特别说明： 搜索到多处文字时，默认插入到第一处文字的地方

调用示例： mPdfView.insertSignatureInText(“/sdcard/myImg.png”, “盖章”, 0);

**insertPhotoIntoPDF(String imagePath, String fieldPhoto)**

功能说明： 将现场拍照的结果插入到指定域所在的区域内

输入参数： imagePath String类型 //图片路径

fieldPhoto String类型 //域名称

输出参数： 无

特别说明： 无

调用示例： mPdfView.insertPhotoIntoPDF(“/sdcard/myImg.png”, “照片”);

insertSoundAnnotation(String soundFileName, int pageNo,float x, float y, String userName, byte[] soundData, int rate,int channels, int bitspersample, String createTime)

功能说明： 插入指定语音批注

输入参数：

| | | | |
|---|---|---|---|
| | soundFileName | String类型 | //文件名 |
| | pageNo | int类型 | //页码 |
| | x | float类型 | //x轴坐标 |
| | y | float类型 | //y轴坐标 |
| | userNmae | String类型 | //使用者（创建者） |
| | soundData | byte[]数组类型 | //裸音数据 |
| | rate | int类型 | //采样率 |
| | channels | int类型 | //声道数 |
| | biespersample | int类型 | //每个采样所需的bit数 |
| | createTime | String类型 | //创建时间 |

输出参数： 无

特别说明： x, y 代表的是pdf文档以左上角为原点的页面坐标而不是屏幕坐标，插入批注时createTime的时间格式必须是以下格式：

"2014-05-29 11:10:10"不能为其他格式。

调用示例： mPdfView.insertSoundAnnotation("zhangsan20180328.raw", 0, 100, 100, "zhangsan", soundData, 22050, 2, 16, "2018-03-28 15:39:12");

insertVectorAnnotation(int pageNo, float x, float y, float w,float h, String line, String userName, String createTime, int type, String annotId, String uniqueName)

功能说明： 插入指定手写矢量批注

输入参数：

| | | | |
|---|---|---|---|
| | pageNo | int类型 | //页码 |
| | x | float类型 | //x轴坐标 |
| | y | float类型 | //y轴坐标 |

| | | |
|---|---|---|
| w | float类型 | //批注宽度 |
| h | float类型 | //批注高度 |
| line | String类型 | //手写的矢量数据 |
| userName | String类型 | //使用者 |
| createTime | String类型 | //批注图片创建时间 |
| type | int类型 | //批注类型 |
| annotId | String类型 | //批注annotId，可传空 |
| uniqueName | String类型 | //批注唯一名称，可传空 |

输出参数： 无

特别说明： x, y 代表的是pdf文档以左上角为原点的页面坐标而不是屏幕坐标，插入批注时createTime的时间格式必须是以下格式：

"2014-05-29 11:10:10"不能为其他格式。

调用示例： mPdfView.insertVectorAnnotation(0, 100, 100, 200, 200, vector, "zhangsan", "2018-03-28 15:39:12", TYPE_ANNOT_HANDWRITE, null, null);

**openHandWriteAnnotation(View layout, PDFHandWriteView handWriteView)**

功能说明： 全文手写批注

输入参数：

| | | |
|---|---|---|
| layout | View类型 | //装载金格手写控件的视图 |
| handWriteView | PDFHandWriteView对象类型 | //金格手写控件 |

输出参数： 无

特别说明： 无

调用示例： mPdfView.openHandwriteAnnotation(handwriteView_layout,full_handWriteView);

**setKgHandwriteViewEraserAnnot(PDFHandWriteView handWriteView)**

功能说明： 设置全文签批的橡皮擦状态时，可以擦除已保存在文档中的手写批注笔迹

输入参数： handWriteView PDFHandWriteView对象类型 //金格手写控件

输出参数： 无

特别说明： 无

调用示例： mPdfView.setKgHandwriteViewEraserAnnot(full_handWriteView);

openTextAnnotation()

功能说明： 添加文字批注

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.openTextAnnotation();

openSoundAnnotation()

功能说明： 添加语音批注

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.openSoundAnnotation();

openAreaHandwrite (Context context)

功能说明： 打开区域手写视图

输入参数： context Context对象类型 //传入Context上下文对象

输出参数： 无

特别说明： 无

调用示例： mPdfView.openAreaHandwrite();

closeAreaHandwrite ()

功能说明： 关闭区域手写视图

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.closeAreaHandwrite();

**closeAreaHandwrite (PDFHandWriteView pdfHandwriteView, boolean needAddSignature)**

功能说明： 关闭并保存区域手写视图

输入参数： pdfHandwriteView PDFHandWriteView对象类型 //传入手写控件

输出参数： 无

特别说明： 无

调用示例： mPdfView.closeAreaHandwrite(pdfHandwriteView, false);

**doSaveSoundAnnot (Annotation sound,float x, float y)**

功能说明： 保存语音批注

输入参数： sound Annotation对象类型 //批注对象

x float类型 //保存批注的x值

y float类型 //保存批注的y值

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型，x和y都可以从参数获取。

调用示例： mPdfView.doSaveSoundAnnot(sound, x, y);

**doSaveTextAnnot(Annotation text, float x,float y)**

功能说明： 保存文字批注

输入参数： text Annotation对象类型 //批注对象

x float类型 //保存批注的x值

y float类型 //保存批注的y值

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型，x和y都可以从参数获取。

调用示例： mPdfView.doSaveTextAnnot(text, x, y);

doSaveTextAnnot(Annotation text, float x,float y, int expand)

功能说明： 保存文字批注

输入参数： text Annotation对象类型 //批注对象

x float类型 //保存批注的x值

y float类型 //保存批注的y值

expand int类型 //在原始图标大小上增大/缩小该值

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型，x和y都可以从参数获取。

调用示例： mPdfView.doSaveTextAnnot(text, x, y, 5);

doSaveFreeTextAnnot(Annotation text, float x,float y)

功能说明： 保存打字机批注

输入参数： text Annotation对象类型 //批注对象

x float类型 //保存批注的x值

y float类型 //保存批注的y值

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型，x和y都可以从参数获取。

调用示例： mPdfView.doSaveFreeTextAnnot(text, x, y);

doSaveHandwriteInfo(boolean isRefresh, boolean isPng, PDFHandWriteView handWriteView)

功能说明： 保存手写内容

输入参数： isRefresh boolean类型 //是否刷新

isPng boolean类型 //是否保存为png图片

handWriteView PDFHandWriteView对象类型 //金格手写控件

输出参数： 无

特别说明： 无

调用示例： mPdfView.doSaveHandwriteInfo(true, false, handWriteView);

doSaveSignByFieldName(PDFHandWriteView handwriteView, String fieldName)

功能说明： 域定位签名

输入参数： handwriteView PDFHandWriteView对象类型 //金格手写控件

fieldName String类型 //域的名称

输出参数： 无

特别说明： 无

调用示例： mPdfView.doSaveSignByFieldName(handwriteView, “Signature1”);

openEBHandwriteView()

功能说明： 打开E人E本手写视图

输入参数： 无

输出参数： 无

特别说明： 项目需要依赖E人E本Library才可使用该功能

调用示例： mPdfView.openEBHandwriteView();

doSaveEBHandwriteInfo()

功能说明： 保存E人E本手写视图中的手写数据

输入参数： 无

输出参数： 无

特别说明： 项目需要依赖E人E本Library才可使用该功能

调用示例： mPdfView.doSaveEBHandwriteInfo();

doCloseEBHandwriteView()

功能说明： 关闭E人E本手写视图中

输入参数： 无

输出参数： 无

特别说明： 项目需要依赖E人E本Library才可使用该功能

调用示例： mPdfView.doCloseEBHandwriteView();

setEBPenDeleteAnnot()

功能说明： 设置EB手写状态时，笔头可擦除批注

输入参数： 无

输出参数： 无

特别说明： 项目需要依赖E人E本Library才可使用该功能，进入EB手写视图时才能使用该功能

调用示例： mPdfView.setEBPenDeleteAnnot();

openHandwriting()

功能说明： 打开通用版手笔分离

输入参数： 无

输出参数： 无

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.openHandwriting();

doSaveHandwriting()

功能说明： 保存通用版手笔分离的手写笔迹

输入参数： 无

输出参数： 无

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.doSaveHandwriting();

undoHandwriting()

功能说明： 通用版手笔分离笔迹撤销，撤销一笔笔迹

输入参数： 无

输出参数： 无

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.undoHandwriting();

redoHandwriting()

功能说明： 通用版手笔分离笔迹恢复，恢复一笔笔迹

输入参数： 无

输出参数： 无

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.redoHandwriting();

isCanUndo()

功能说明： 通用版手笔分离是否可撤销笔迹

输入参数： 无

输出参数： boolean类型 //true:可以撤销笔迹 false:无笔迹可撤销

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： boolean canUndo = mPdfView.isCanUndo();

isCanRedo()

功能说明： 通用版手笔分离是否可恢复笔迹

输入参数： 无

输出参数： boolean类型 //true:可以恢复笔迹 false:无笔迹可恢复

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： boolean canRedo = mPdfView.isCanRedo();

isHaveHandwriting()

功能说明： 判断通用版手笔分离是否有笔迹

输入参数： 无

输出参数： boolean类型 //true:有手写笔迹 false:没有手写笔迹

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.isHaveHandwriting();

doClearHandwriting()

功能说明： 清除通用版手笔分离笔迹

输入参数： 无

输出参数： 无

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.doClearHandwriting();

doCloseHandwriting()

功能说明： 退出通用版手笔分离

输入参数： 无

输出参数： 无

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.doCloseHandwriting();

doEraserHandwriting(boolean isEraser)

功能说明： 通用版手笔分离的橡皮擦功能

输入参数： isEraser boolean类型 //true:使用橡皮擦，false:不使用橡皮擦

输出参数： 无

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.doEraserHandwriting(true);

isEraser()

功能说明： 通用版手笔分离橡皮擦的状态

输入参数： 无

输出参数： boolean类型 //true:在使用橡皮擦 false:在手写状态

特别说明： 支持任一设备，非E人E本可调用该方法

调用示例： mPdfView.isEraser();

doCloseHandwriteInfo(View layout, PDFHandWriteView handwriteView)

功能说明： 关闭全文签批手写视图

输入参数： layout View类型 //装载金格手写控件的视图

handWriteView PDFHandWriteView对象类型 //金格手写控件

输出参数： 无

特别说明： 无

调用示例： mPdfView.doCloseHandwriteInfo(layout, handwriteView);

doDeleteSoundAnnot (Annotation annotation,String filePath)

功能说明： 删除语音批注

输入参数： annotation Annotation对象类型 //批注对象

filePath 字符串类型 //语音的缓存路径

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： mPdfView.doDeleteSoundAnnot(soundAnnot, filePath);

doDeleteTextAnnotation(Annotation annotation)

功能说明： 删除文字批注

输入参数： annotation Annotation对象类型 //批注对象

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： mPdfView.doDeleteTextAnnotation(textAnnot);

doUpdateTextAnnotation(Annotation annotation)

功能说明： 更新文字批注

输入参数： annotation Annotation对象类型 //批注对象

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： mPdfView.doUpdateTextAnnotation(textAnnot);

doDeleteFreeTextAnnotation(Annotation annotation)

功能说明： 删除打字机批注

输入参数： annotation Annotation对象类型 //批注对象

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： mPdfView.doDeleteFreeTextAnnotation(textAnnot);

doUpdateFreeTextAnnotation(Annotation annotation)

功能说明： 更新打字机批注

输入参数： annotation Annotation对象类型 //批注对象

输出参数： 无

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： mPdfView.doUpdateFreeTextAnnotation(textAnnot);

doSaveLocalDigitallySign(float x, float y, float width, float height, String pfxPath, String pfxPwd, int pageNo, String imagePath)

功能说明： 插入数字签名（使用本地证书做数字签名）

输入参数： x float类型 //数字签名外观（印章图片）x坐标

y float类型 //数字签名外观（印章图片）x坐标

width float类型 //数字签名外观（印章图片）宽度

height float类型 //数字签名外观（印章图片）高度

pfxPath String类型 //数字签名pfx证书路径

pfxPwd String类型 //数字签名pfx证书密码

pageNo int类型 //插入数字签名外观（印章图片）的页码

imagePath String类型 //数字签名外观（印章图片）的路径

输出参数： 无

特别说明： 无

调用示例： mPdfView.doSaveLocalDigitallySign(350, 600, 150, 150, SDCARD_PATH + ″

/pdfsign.pfx", "123456", 0, SDCARD_PATH + "/testSignature.jpg");

**getAnnotationNewList(final String user, final int type)**

功能说明： 获取当前文档中指定用户，指定类型的批注列表

输入参数： user string类型 //用户名

type int类型 //批注类型

输出参数： ArrayList<Annotation>类型

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： mPdfView.getAnnotationNewList(“zhangsan”, TYPE_ANNOT_TEXT);

**getAnnotationList(int type)**

功能说明： 获取当前文档中指定类型的批注列表

输入参数： type int类型 //批注类型

输出参数： ArrayList<Annotation>类型

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： mPdfView.getAnnotationList(TYPE_ANNOT_TEXT);

**getAnnotationList(int[] typeList)**

功能说明： 获取当前文档中指定的多种类型的批注列表

输入参数： typeList int数组类型 //多种批注类型

输出参数： ArrayList<Annotation>类型

特别说明： Annotation为iAppPDFJar定义的对象类型

调用示例： int[] typeList = new int[]{1,2,4}; //同时获取Stamp，Text，Sound类型的批注

mPdfView.getAnnotationList(typeList);

**setAllAnnotationsOnlyRead(boolean isOnlyRead)**

功能说明： 设置文档所有注释是否锁定

输入参数： isOnlyRead boolean类型 //是否锁定 true：锁定，false：不锁定

输出参数： 无

特别说明： 无

调用示例： mPdfView.setAllAnnotationsOnlyRead(true);

setAnnotationsOnlyRead(boolean isOnlyRead, int annotType)

功能说明： 设置某一类文档注释是否锁定

输入参数： isOnlyRead boolean类型 //是否锁定 true：锁定，false：不锁定

annotType int类型 //批注类型

输出参数： 无

特别说明： 无

调用示例： mPdfView.setAnnotationsOnlyRead(true, TYPE_ANNOT_TEXT);

setCustomType(String type)

功能说明： 设置用户自定义批注类型

输入参数： type String类型 //用户自定义类型

输出参数： 无

特别说明： 设置批注标识，用户可以自定义设置该标识，以便区分批注

调用示例： mPdfView.setCustomType("myAnnot");

deleteAllAnnotations(int type)

功能说明： 删除当前用户的指定类型的所有批注

输入参数： type int类型 //批注类型

输出参数： 无

特别说明： 无

调用示例： mPdfView.deleteAllAnnotations(TYPE_ANNOT_TEXT);

deleteAllStampAnnot()

功能说明： 删除整个文档STAMP类型（图片，手写）的批注

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： mPdfView.deleteAllStampAnnot();

deleteAnnotWithId(String annotId)

功能说明： 删除指定annotId的批注

输入参数： annotId String类型 //annotId

输出参数： 无

特别说明： 无

调用示例： mPdfView.deleteAnnotWithId(annotId);

deleteAnnotWithUniqueName(String uniqueName)

功能说明： 删除指定uniqueName的批注

输入参数： uniqueName String类型 //uniqueName

输出参数： 无

特别说明： 无

调用示例： mPdfView.deleteAnnotWithUniqueName(uniqueName);

updateAnnotRect(String annotId, int pageNo, int x, int y)

功能说明： 移动批注

输入参数： annotId String类型 //annotId

pageNo int类型 //页码

x int类型 //x轴坐标

y int类型 //y轴坐标

输出参数： 无

特别说明： x，y 代表的是pdf文档以左上角为原点的页面坐标而不是屏幕坐标

调用示例： mPdfView.updateAnnotRect(annotId, 0, 100, 100);

setCanMoveAnnot(boolean isCanMove)

功能说明： 设置是否能移动批注

输入参数： isCanMove boolean类型 //是否能移动批注 true:可以移动 false:不能移动

输出参数： 无

特别说明： 无

调用示例： mPdfView.setCanMoveAnnot(false);

updateTextAnnotContent(String annotId, String content)

功能说明： 修改文字批注内容

输入参数： annotId String类型 //annotId

content String类型 //批注内容

输出参数： 无

特别说明： 无

调用示例： mPdfView.updateTextAnnotContent(annotId, “新内容”);

getFieldByName(String fieldName)

功能说明： 通过域名得到指定的域

输入参数： fieldName String类型 //域名

输出参数： ArrayList<SPField>类型

特别说明： 无

调用示例： ArrayList<SPField> fields = mPdfView.getFieldByName(“field1”);

setFieldContent(String fieldName, String content)

功能说明： 填充指定域的内容

输入参数： fieldName String类型 //域名

content String类型 //填充内容

输出参数： 无

特别说明： 无

调用示例： mPdfView.setFieldContent(“field1”, “内容”);

getFieldContent(String fieldName)

功能说明： 获取指定名称的域的内容

输入参数： fieldName String类型 //域名

输出参数： String类型

特别说明： 无

调用示例： String content = mPdfView.getFieldContent(“field1”);

getAllTextField()

功能说明： 得到所有的文本域

输入参数： 无

输出参数： ArrayList<SPField>类型

特别说明： 无

调用示例： ArrayList<SPField> fields = mPdfView.getAllTextField();

getAllFieldContent()

功能说明： 获取所有编辑域内容

输入参数： 无

输出参数： ArrayList<String>类型

特别说明： 无

调用示例： ArrayList<String> strs = mPdfView.getAllFieldContent();

setFormFillEnabled(boolean enable)

功能说明： 设置是否允许表单填充

输入参数： enable boolean类型 //是否允许表单填充

输出参数： 无

特别说明： 无

调用示例： mPdfView.setFormFillEnabled(false);

setHighlightField(boolean highlight)

功能说明： 设置是否允许表单填充

输入参数： highlight boolean类型 //域是否高亮

输出参数： 无

特别说明： 无

调用示例： mPdfView.setHighlightField(false);

hideLongPressRect(boolean isHide)

功能说明： 是否隐藏长按删除批注的矩形框

输入参数： isHide boolean类型 //是否隐藏

输出参数： 无

特别说明： 无

调用示例： mPdfView.hideLongPressRect(true);

hideAnnotToast(boolean isHide)

功能说明： 是否隐藏点击批注时出现的toast

输入参数： isHide boolean类型 //是否隐藏

输出参数： 无

特别说明： 无

调用示例： mPdfView.hideAnnotToast(true);

setAnnotationsVisible(boolean visibleFlag, String userName)

功能说明： 设置指定用户的批注可见或者不可见

输入参数： visibleFlag boolean类型 //是否可见

userName String类型 //用户名

输出参数： 无

特别说明： 无

调用示例： mPdfView.setAnnotationsVisible(false, “admin”);

isBlendMode(boolean modeStatus)

功能说明： 设置插入批注时是否使用交叉混合渲染模式

输入参数： modeStatus boolean类型 //是否是交叉混合渲染模式

输出参数： 无

特别说明： 无

调用示例： mPdfView.isBlendMode(true);

setAnnotationsVisible(boolean visibleFlag, String userName, int type)

功能说明： 设置指定用户，某种类型的批注可见或者不可见

输入参数： visibleFlag boolean类型 //是否可见

userName String类型 //用户名

type int类型 //批注类型

输出参数： 无

特别说明： 无

调用示例： mPdfView.setAnnotationsVisible(false, “admin”, KinggridConstant.TYPE_ANNOT_TEXT);

setAllAnnotationsVisible(boolean visibleFlag, int type)

功能说明： 设置某种类型的所有批注可见或不可见

输入参数： visibleFlag boolean类型 //是否可见

type int类型 //批注类型

输出参数： 无

特别说明： 无

调用示例： mPdfView.setAllAnnotationsVisible(false, KinggridConstant.TYPE_ANNOT_TEXT);

setAnnotationsVisibleByUnique(boolean visibleFlag, String uniqueName)

功能说明： 设置uniqueName的批注可见或者不可见

输入参数： visibleFlag boolean类型 //是否可见

uniqueName String类型 //批注的唯一名称

输出参数： 无

特别说明： 无

调用示例： mPdfView.setAnnotationsVisibleByUnique(false, “20180706142536”);

hasFieldInDocument(String fieldName)

功能说明： 判断一个文档中是否存在指定域

输入参数： 无

输出参数： boolean类型

特别说明： 无

调用示例： boolean hasField = mPdfView.hasFieldInDocument(“field1”);

pageIndexFromPoint(float x, float y)

功能说明： 判断一个坐标点处于PDF文档中的哪一页

输入参数： x float类型 //坐标点的x轴

y float类型 //坐标点的y轴

输出参数： int类型

特别说明： 无

调用示例： int pageIndex = mPdfView.pageIndexFromPoint(400, 800);

coordinateViewToUser(int pageIndex, PointF point)

功能说明： 视图坐标（相对坐标）转PDF用户坐标

输入参数： pageIndex int类型 //页面索引（页码）

point PointF类型 //视图坐标点

输出参数： 无

特别说明： 无

调用示例： mPdfView.coordinateViewToUser(0, new PointF(400, 800));

doSaveSignature(final String certData, final String signData,final String hash, final String signPath, final float width, final float height, final float x, final float y, final int pageNo,final SignatureInfo info,final OnInsertKGSealCompleteListener listener)

功能说明： 插入数字签名/电子签章

输入参数： certData String类型 //证书数据（做数字签名时此参数可传空）

signData String类型 //签名数据（做数字签名时此参数可传空）

hash String类型 //PDF文档哈希值（做数字签名时此参数可传空）

signPath String类型 //印章图片路径

width float类型 //印章图片插入到文档中的宽度

height float类型 //印章图片插入到文档中的高度

x float类型 //印章图片插入到文档中的坐标x轴

y float类型 //印章图片插入到文档中的坐标y轴

pageNo int类型 //印章图片插入到文档中的页码

info SignatureInfo对象类型 //签名信息（做数字签名时此参数可传空）

listener OnInsertKGSealCompleteListener //插入数字签名/电子签章完成的回调监听

输出参数： 无

特别说明： 无

调用示例： 插入数字签名示例：

```
SignatureUtil.getInstance().setSignType(SignType.DIGITALLY);//设置插入数字签名模式
SignatureUtil.getInstance().setMode(SignMode.SOF);//设置签名模式为“本地证书”签名
int code = SignatureUtil.getInstance().setPFXInfo(SDCARD_PATH + "/p dfsign.pfx", “123456”);//设置pfx证书文件的路径和密码
if (code == 0) {
    mPdfView.doSaveSignature(null, null, null, imgPath, 150, 150, 3 50, 600, 0, null, null);//插入数字签名
```

```
}
```

插入电子签章示例：

```
SignatureUtil.getInstance().setSignType(SignType.KGSEAL);//设置插入电子签章模式
SignatureUtil.getInstance().setMode(SignMode.SOF);//设置签名模式为“本地证书”签名
int code = SignatureUtil.getInstance().setPFXInfo(SDCARD_PATH + "/pdfsign.pfx",”123456”);//设置pfx证书文件的路径和密码
if (code == 0) {
    hash = SignatureUtil.getInstance().getKGHash(mPdfView.getDocument());//获取文档哈希
    byte[] certData = SignatureUtil.getInstance().getCert();//获取证书值
    byte[] signData = SignatureUtil.getInstance().sign(hash.getBytes());//获取签名值
    certBase64 = Base64.encodeToString(certData,Base64.DEFAULT);
    signBase64 = Base64.encodeToString(signData,Base64.DEFAULT);
    mPdfView.doSaveSignature(certBase64, signBase64, hash, imgPath, width, height, 350, 600, 0, null, null);//插入金格电子签章
}
```

```
doSaveKGSeriesSeal(final String certData, final String signData, final String hash, final String imagePath, final RectF[] rects, final int start, final int end, final SignatureInfo info, final OnInsertKGSealCompleteListener listener)
```

功能说明：　插入金格电子签章（批量盖章）

输入参数：　certData　String类型　//证书数据

signData　String类型　//签名数据

hash　String类型　//PDF文档哈希值

imagePath　String类型　//印章图片路径

rects　RectF对象数组类型　//印章图片位置

start　int类型　//盖章起始页（从1开始）

end　int类型　//盖章结束页

info　SignatureInfo对象类型　//签名信息

listener OnInsertKGSealCompleteListener //插入数字签名/电子签章完成的回调监听

输出参数： 无

特别说明： 无

调用示例：

```
int s = Integer.parseInt(startPage.getText().toString());
int e = Integer.parseInt(endPage.getText().toString());
RectF[] rects = new RectF[e - s + 1];
BitmapFactory.Options options = new BitmapFactory.Options();
options.inJustDecodeBounds = true;
BitmapFactory.decodeFile(imgPath, options);
int width = options.outWidth;
int height = options.outHeight;
for (int i = 0; i < rects.length; i++) {
    rects[i] = new RectF(230, 420, 230 + width, 420 + height);
}
SignatureUtil.getInstance().setSignType(SignType.KGSEAL);//设置为插入金格电子签章模式
SignatureUtil.getInstance().setMode(SignMode.SOF);//设置签名模式为“本地证书”签名
int code = SignatureUtil.getInstance().setPFXInfo(SDCARD_PATH + "/pdfsign.pfx", “123456”);//设置pfx证书文件的路径和密码
if (code == 0) {
    hash = SignatureUtil.getInstance().getKGHash(mPdfView.getDocument());//获取文档哈希
    byte[] certData = SignatureUtil.getInstance().getCert();//获取证书值
    byte[] signData = SignatureUtil.getInstance().sign(hash.getBytes());//获取签名值
    certBase64 = Base64.encodeToString(certData,Base64.DEFAULT);
```

```
signBase64 = Base64.encodeToString(signData,Base64.DEFAULT);
mPdfView.doSaveKGSeriesSeal(certBase64, signBase64, hash, imgPath, re
    cts, s, e, null, insertKGSealListener);//批量插入金格电子签章
}
```

doSavePerforationDivSeal(final Bitmap sealBitmap, final float y, final String certData, final String signData, final String hash, final String signPath, final SignatureInfo info, final OnInsertKGSealCompleteListener listener)

功能说明： 插入金格电子签章（骑缝章）

输入参数： sealBitmap Bitmap对象类型 //印章图片位图

y float类型 //印章图片在文档的y轴

certData String类型 //证书数据

signData String类型 //签名数据

hash String类型 //文档哈希值

signPath String类型 //印章图片路径

info SignatureInfo对象类型 //签名信息

listener OnInsertKGSealCompleteListener //插入数字签名/电子签章完成的回调监听

输出参数： 无

特别说明： 无

调用示例：

```
SignatureUtil.getInstance().setSignType(SignType.KGSEAL);//设置为插入金格电子签章模式
SignatureUtil.getInstance().setMode(SignMode.SOF);//设置签名模式为“本地证书”签名
int code = SignatureUtil.getInstance().setPFXInfo(SDCARD_PATH + "/pdfsign.pfx", “123456”);//设置pfx证书文件的路径和密码
if (code == 0) {
    hash = SignatureUtil.getInstance().getKGHash(mPdfView.getDocument());
```

```
byte[] certData = SignatureUtil.getInstance().getCert();
byte[] signData = SignatureUtil.getInstance().sign(hash.getBytes());
certBase64 = Base64.encodeToString(certData,Base64.DEFAULT);
signBase64 = Base64.encodeToString(signData,Base64.DEFAULT);
FileInputStream fis = null;
try {
    fis = new FileInputStream(imagePath);
} catch (FileNotFoundException e) {
    e.printStackTrace();
}
bitmap = BitmapFactory.decodeStream(fis);
y = mPdfView.getHeight() / 5;
mPdfView.doSavePerforationDivSeal(bitmap, y, certBase64, signBase64, h
    ash,imagePath, null, insertKGSealListener);//插入金格电子签章（骑缝
    章）
}
```

doSaveOFDSeal(byte[] sealData, int x, int y, int pageNo)

功能说明： ofd文档插入电子签章

输入参数： sealData byte[]类型 //印章图片信息（国密0031标准数据）

x int类型 //印章图片在文档的x轴

y int类型 //印章图片在文档的y轴

pageNo int类型 //印章图片所在页码

输出参数： 无

特别说明： 无

调用示例： mPdfView.doSaveOFDSeal(sealData, 400, 400, 0);

setOnViewTouchAddAnnotListener(OnViewTouchAddAnnotListener listener)

功能说明： 监听添加批注：文字、语音、等

输入参数： listener OnViewTouchAddAnnotListener对象类型 //触摸PDF添加批注监听

输出参数： 无

特别说明： 无

调用示例： mPdfView.setOnViewTouchAddAnnotListener(new OnViewTouchAddAnnotListener() {

```
@Override
public void onTouch(float x, float y) {
    if (mPdfView.isSound) {//添加语音批注
        mPdfView.isSound = false;
        annotUtil.addSoundAnnot(x, y);
    }
    if (mPdfView..isAnnotation){//添加文字批注
        mPdfView.isAnnotation = false;
        annotUtil.addTextAnnot(x, y);
    }
}
});
```

setOnViewTouchShowAnnotListener(OnViewTouchShowAnnotListener listener)

功能说明： 监听显示批注：文字、语音等

输入参数： listener OnViewTouchShowAnnotListener对象类型 //触摸PDF显示批注监听

输出参数： 无

特别说明： 无

调用示例： mPdfView.setOnViewTouchShowAnnotListener(new OnViewTouchShowAnnotListener() {

```
@Override
public void onTouchTextAnnot(Annotation annot) {
    annotUtil.showTextAnnot(annot);
```

```
            }
            @Override
            public void onTouchSoundAnnot(Annotation annot) {
                annotUtil.showSoundAnnot(annot);
            }
            @Override
            public void onTouchFreeTextAnnot(Annotation annot){
                //点击的是打字机注释
            }
        });
```

setOnPageChangeListener(OnPageChangeListener listener)

功能说明： 监听翻页事件

输入参数： listener OnPageChangeListener对象类型 //文档翻页监听

输出参数： 无

特别说明： 无

调用示例：

```
mPdfView.setOnPageChangeListener(new OnPageChangeListener() {
            @Override
            public void onPageChange(String arg0) {
                int pageIndex = Integer.parseInt(arg0) + 1;
                DocumentActivity.this.setTitle("(" + pageIndex +"/" + mPdfView.
                    getPageCount() + ")" +mFilePath);
            }
        });
```

setOnHandwritingSavedListener(OnHandwritingSavedListener listener)

功能说明： 监听手写笔迹保存事件

输入参数： listener OnHandwritingSavedListener对象类型 //手写笔迹保存监听

输出参数： 无

特别说明： 无

调用示例： mPdfView.setOnHandwritingSavedListener(new OnHandwritingSavedListener()

{

@Override

public void onHandwritingSaved(JSONArray jsonArray) {

Toast.makeText(mContext, "手写笔迹保存成功", Toast.LENGTH_SH

ORT).show();

}

});

setOnLongPressDeleteListener(OnLongPressDeleteListener listener)

功能说明： 监听长按删除批注事件

输入参数： listener OnLongPressDeleteListener对象类型 //长按删除批注监听

输出参数： 无

特别说明： 无

调用示例： mPdfView.setOnLongPressDeleteListener(new OnLongPressDeleteListener ()

{

@Override

public void onDelete(Annotation annot) {

String userName = annot.getAuthorName();

Toast.makeText(mContext, userName + "的批注已删除", Toast.LENGTH_

SHORT).show();

}

});

setOnClickLocateViewOkBtnListener(OnClickLocateViewOkBtnListener listener)

功能说明： 监听可拖动视图确定按钮事件

输入参数： listener OnClickLocateViewOkBtnListener对象类型 //可拖动视图确定按钮事件监听

输出参数： 无

特别说明： 无

调用示例：
```
mPdfView.setOnClickLocateViewOkBtnListener(new OnClickLocateViewOkBtnListener() {
    @Override
    public void clickOkBtn(int pageIndex, float x, float y, float width, float height) {
        String filePath = "/sdcard/testSignature2.jpg";
        mPdfView.insertHandWriteAnnotation(pageIndex, x, y, width, height,
                filePath, "Susie", "2019-12-25 15:26:52", 1, null, null);
        mPdfView.refreshPage(pageIndex);
    }
});
```

setOnFieldTouchListener(OnFieldTouchListener listener)

功能说明： 监听点击域事件

输入参数： listener OnFieldTouchListener 对象类型 //点击域事件监听

输出参数： 无

特别说明： 无

调用示例：
```
mPdfView.setOnFieldTouchListener(new OnFieldTouchListener() {
    @Override
    public void onFieldTouch(Field field, PointF point) {
        Toast.makeText(mContext, "field onClicked!", Toast.LENGTH_SHORT).show();
    }
});
```

setOnViewGestureListener(OnViewGestureListener listener)

功能说明： 监听页面手势事件（点击和左右滑动事件监听）

输入参数： listener OnViewGestureListener 对象类型 //手势监听

输出参数： 无

特别说明： 无

调用示例：
```
mPdfView.setOnViewGestureListener(new OnViewGestureListener() {
    @Override
    public boolean onSingleTapUp(MotionEvent e, int clickType) {
        Toast.makeText(mContext, "clicked!", Toast.LENGTH_SHORT).show
        ();
        return false;
    }
    @Override
    public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX,
            float velocityY, int moveDirection) {
        if (moveDirection == 1) {
            //左滑
        } else if (moveDirection == 2) {
            //右滑
        }
        return false;
    }
});
```

setOnViewMovedListener(OnViewMovedListener listener)

功能说明： 监听页面移动和缩放事件

输入参数： listener OnViewMovedListener对象类型 //手势监听

输出参数： 无

特别说明： 无

调用示例：mPdfView.setOnViewMovedListener(new OnViewMovedListener() {

```
@Override
public void onViewZoomChanged(float newZoom, float oldZoom) {
    System.out.println("====onViewMoved zoomchanged:" + newZoom
    +", " + oldZoom);
}
@Override
public void onViewMoved(int offsetX, int offsetY) {
    System.out.println("====onViewMoved:" + offsetX +", " + offsetY);
}
});
```

### 8.2.1 PDFHandWriteView 接口调用说明

**1. 第三方应用需为金格手写控件设计布局。**

```
<com.kinggrid.iapppdf.ui.viewer.PDFHandWriteView
        android:id="@+id/v_canvas"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@android:color/transparent" />
```

**2. 初始化金格控件。**

```
PDFHandWriteView sign_handWriteView = (PDFHandWriteView) layout.findViewById(R.id.v_canvas);
//设置许可
sign_handWriteView.setCopyRight(activity, copy_right);
//设置笔宽、笔型、颜色等
sign_handWriteView.setPenInfo(signConfigDialog.getPenMaxSizeFromXML(),
        signConfigDialog.getPenColorFromXML(), signConfigDialog.getPenTypeFromXML());
```

### 8.2.2 PDFHandWriteView 接口方法说明

cancelEarseHandwriteInfo()

功能说明： 退出擦除模式。

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： sign_handWriteView.cancelEarseHandwriteInfo();

canSave()

功能说明： 是否能够保存，即是否存在已签笔迹。

输入参数： 无

输出参数： boolean 是否能够保存，当没有已签笔迹即屏幕为空时为false,反之为true

特别说明： 无

调用示例： if (sign_handWriteView.canSave()){...}

doClearHandwriteInfo()

功能说明： 清除已签笔迹。

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： sign_handWriteView.doClearHandwriteInfo()

doEarseHandwriteInfo()

功能说明： 擦除手写笔迹。

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： sign_handWriteView.doEarseHandwriteInfo()

doRedoHandwriteInfo()

功能说明： 撤销一个回退笔迹。

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： sign_handWriteView.doRedoHandwriteInfo()

doSettingHandwriteInfo()

功能说明： 设置手写信息。

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： sign_handWriteView.doSettingHandwriteInfo()

doUndoHandwriteInfo()

功能说明： 回退一个已签笔迹。

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： sign_handWriteView.doUndoHandwriteInfo()

getEarseState()

功能说明： 判断是否处于擦除模式

输入参数： 无

输出参数： 返回true则处于擦除模式，否则处于手写模式

特别说明： 无

调用示例： getEarseState();

resetPenSetting()

功能说明： 重置手写设置为默认

输入参数： 无

输出参数： 无

特别说明： 无

调用示例： resetPenSetting();

setCopyRight(Activity activity,String copyRight)

功能说明： 设置授权码。

输入参数： activity 对象类型 //Activity

copyRight 字符串类型 //授权码

输出参数： 无

特别说明： 无

调用示例： setCopyRight(this,copyRight);

setPenInfo(float penWidth, final int penColor, final int penType)

功能说明： 设置手写笔的笔宽、颜色、笔型等。

输入参数： penWidth float类型 //要设置的笔宽

penColor int类型 //要设置的颜色

penType int类型 //要设置的笔型

输出参数： 无

特别说明： 无

调用示例： sign_handWriteView.setPenInfo(signConfigDialog.getPenMaxSizeFromXML(),
signConfigDialog.getPenColorFromXML(),
signConfigDialog.getPenTypeFromXML());

注：

已定义的笔形说明如下：

int TYPE_BALLPEN 钢笔

int TYPE_BRUSHPEN 毛笔

int TYPE_PENCIL 铅笔

int TYPE_WATERPEN 水彩笔

### 8.3.1 SignatureUtil 接口调用说明

**SignatureUtil 是用于对 PDF 文档做数字签名或金格电子签章所提供的一个签名工具类，使用 SignatureUtil 类设置一些签名相关参数。调用示例如下：**

```
SignatureUtil.getInstance().setSignType(SignType.KGSEAL);//设置插入的签名为金格电子签章
SignatureUtil.getInstance().setMode(SignMode.SOF);//设置签名模式为本地证书签名
//设置 pfx 证书文件的路径和证书密码
int code = SignatureUtil.getInstance().setPFXInfo(SDCARD_PATH + "/pdfsign.pfx",”123456”);
String hash = SignatureUtil.getInstance().getKGHash(mPdfView.getDocument());//获得文档哈希
byte[] certData = SignatureUtil.getInstance().getCert();//获得证书数据
byte[] signData = SignatureUtil.getInstance().sign(hash.getBytes());//获取签名数据
```

### 8.3.2 SignatureUtil 接口方法说明

`getInstance()`

功能说明： 获取SignatureUtil实例

输入参数： 无

输出参数： SignatureUtil对象

特别说明： 无

调用示例： SignatureUtil.getInstance();

`sign(byte[] origData)`

功能说明： 对原文数据进行签名

输入参数： origData byte[]类型 //原文数据

输出参数： byte[]类型 //返回签名后的数据（签名数据）

特别说明： 无

调用示例： SignatureUtil.getInstance().sign(data);

setPFXInfo(String pfxPath, String pfxPwd)

功能说明： 设置本地证书路径和本地证书密码

输入参数： pfxPath String类型 //pfx证书文件路径

pfxPwd String类型 //pfx证书文件密码

输出参数： 无

特别说明： 如果是本地证书签名模式，则需要设置本地证书路径和本地证书密码

调用示例： SignatureUtil.getInstance().setPFXInfo("/sdcard/pfxsign.pfx", "123456");

getPFXPassword()

功能说明： 获取pfx证书密码

输入参数： 无

输出参数： String类型 //pfx证书密码

特别说明： 无

调用示例： String pwd = SignatureUtil.getInstance().getPFXPassword();

getSubjectName()

功能说明： 获取证书使用者名称

输入参数： 无

输出参数： String类型 //证书使用者名称

特别说明： 无

调用示例： String name = SignatureUtil.getInstance().getSubjectName();

getCert()

功能说明： 获取证书数据

输入参数： 无

输出参数： byte[]类型 //证书数据

特别说明： 无

调用示例： String name = SignatureUtil.getInstance().getCert();

setMode(SignMode mode)

功能说明： 设置签名方式

输入参数： mode SignMode枚举类型 //签名方式

输出参数： 无

特别说明： SignMode 枚举值为：

SignMode.NONE //无

SignMode.BLE //蓝牙KEY

SignMode.CA //CA机构

SignMode.TF //TF卡

SignMode.SOF //本地证书

调用示例： SignatureUtil.getInstance().setMode(SignMode.SOF);

getMode()

功能说明： 获取签名方式

输入参数： 无

输出参数： SignMode枚举类型

特别说明： SignMode 枚举值为：

SignMode.NONE //无

SignMode.BLE //蓝牙KEY

SignMode.CA //CA机构

SignMode.TF //TF卡

SignMode.SOF //本地证书

调用示例： SignMode mode = SignatureUtil.getInstance().getMode();

setSignType(SignType type)

功能说明： 设置签名类型（数字签名还是电子签章）

输入参数： type SignType枚举类型 //签名类型

输出参数： 无

特别说明： SignType 枚举值为：

SignType.DIGITALLY //数字签名

SignType.KGSEAL //金格电子签章

调用示例： SignatureUtil.getInstance().setSignType(SignType.DIGITALLY);

getSignType()

功能说明： 获取签名类型（数字签名还是电子签章）

输入参数： 无

输出参数： SignType枚举类型

特别说明： SignType 枚举值为：

SignType.DIGITALLY //数字签名

SignType.KGSEAL //金格电子签章

调用示例： SignType type = SignatureUtil.getInstance().setSignType(SignType.DIGITALLY);

getKGHash()

功能说明： 获取PDF文档哈希值

输入参数： 无

输出参数： String类型 //PDF文档的哈希值

特别说明： 无

调用示例： String hash = SignatureUtil.getInstance().getKGHash();

# 九、 文档声明

本文档内容改动及版本更新将不再另行通知。本文档的范例中使用的人名、公司名和数据如果没有特别指明，均属虚构。对于本文档、及本文档涉及的技术和产品，江西金格科技股份有限公司拥有其专利、商标、著作权或其它知识产权，除非得到江西金格科技股份有限公司的书面许可，本文档不授予这些专利、商标、著作权或其它知识产权的许可。

- kinggrid、iWebOffice、iSignature、SurRead Office、可信云、信签和DBPacket是江西金格科技股份有限公司的商标。
- 其它标牌和产品名称是其各自公司的商标或注册商标。

（完）