WPS Android 版 API 使用手册 V6.3

文档版本：1.0

更新日期：2015-01-26

一、WPS Office API 简介

通过第三方程序启动 WPS，然后进行文档的编辑、修改等操作，甚至可以通过第三方程序控制 wps 的一些基本操作，如打开文档、保存、另存、复制、剪切、粘贴、插入图片（文字）等功能。而根据操作的类型，调用打开方式分为第三方启动方式以及 AIDL 方式两种。

二、第三方启动方式

1.方式简介

第三方启动方式是指通过 StartActivity 方式，启动 WPS，启动的时候可以传入一些参数，以便控制文档的打开效果。主要可以实现控制上次打开进度、上次缩放进度、 第三方操作身份验证等功能。

2.参数介绍

（1）打开模式参数列表

模式	说明	备注
” ReadOnly”	只读模式	
“Normal”	正常模式	
” ReadMode”	打开直接进入阅读器模式	仅 Word、TXT 文档支持
” SaveOnly”	保存模式(打开文件,另存,关闭)	仅 Word、TXT 文档支持

（2）文档打开参数列表

参数名	参数说明	类型	默认值
OpenMode	打开文件的模式。	String	Normal
SendSaveBroad	文件保存时是否发送广播。	boolean	false
SendCloseBroad	文件关闭时是否发送广播。	boolean	false
ThirdPackage	第三方的包名，关闭的广播会包含该项。	String	
ClearTrace	关闭文件时是否删除使用记录(即删除照片墙上的痕迹，就是文档的预览图)。	boolean	false
ClearFile	关闭文件时是否删除打开的文件。	boolean	false
ViewProgress	文件上次查看的进度。	float	0.0%
AutoJump	是否自动跳转到上次查看的进度。	boolean	false
SavePath	文件保存路径。	String	
ViewScale	文件上次查看的视图的缩放。	float	1.0
ViewScrollX	文件上次查看的视图的 X 坐标。	int	0
ViewScrollY	文件上次查看的视图的 Y 坐标。	int	0

UserName	批注的作者。	String	
HomeKeyDown	监听 home 键并发广播	boolean	false
BackKeyDown	监听 back 键并发广播	boolean	false
EnterReviseMode	以修订模式打开文档	boolean	false
CacheFileInvisible	Wps 生成的缓存文件外部是否可见	boolean	True
AutoPlay	【演示文档】打开演示文档进入自动播放模式	boolean	True
AutoPlayInternal	【演示文档】自动播放时间间隔设置	int	0
WaterMaskText	设置水印文字内容 bundle.putString("WaterMaskText", "水印测试 style");	string	
WaterMaskColor	设置水印颜色 bundle.putInt("WaterMaskColor", Color.argb(51, 238, 238, 238));	int	51, 238, 238, 238

PS：以上参数均放在 Intent 中，如有需要使用或者需要参加，请使用 Intent 添加或获取参数的方式使用。例如：

```
Intent intent = new Intent();
Bundle bundle = new Bundle();
bundle.putString(Define.OPEN_MODE, OpenMode);          //打开模式
```

WPS Office 包名类名相关参数

参数名	值
className	cn.wps.moffice.documentmanager.PreStartActivity2
packageName	com.kingsoft.moffice_pro

（3）广播事件

3.1 Back 按钮广播

BackKeyDown 的值为 true 的时候，会发送该条广播

参数：action = “com.kingsoft.writer.back.key.down”

3.2 Home 按钮广播

HomeKeyDown 的值为 true 的时候，会发送该条广播

参数：action = “com.kingsoft.writer.home.key.down”

3.3 保存文件

文件保存时，SendSaveBroad 为 true 的时候，会发送一个" cn.wps.moffice.broadcast.AfterSaved"的广播。

广播含有以下信息，具体见下表。

参数名	参数说明	类型	默认值
“CurrentPath”	当前文档路径	String	
“ThirdPartyPackage”	传入的第三方的包名。	String	
“SaveAs”	本次保存事件是否是另存	Boolean	

3.4 关闭文件

文件关闭时，SendCloseBroad 值为 true 时，会发送一个" cn.wps.moffice.broadcast.AfterClosed "的广播。

广播含有以下信息，具体见下表。

参数名	参数说明	类型	默认值
“CurrentPath”	关闭文件的路径	String	
“ThirdPartyPackage”	传入的第三方的包名。	String	
“ViewProgress”	文件查看的进度	float	0.0%
“ViewScale”	文件上次查看的视图的缩放	float	1.0
“ViewScrollX”	文件上次查看的视图的 X 坐标	int	0
“ViewScrollY”	文件上次查看的视图的 Y 坐标	int	0

3.示例代码

PS：使用 StartActivity 方式，传入指定的参数，即可控制 WPS 打开的时候表现方式。以打开指定路径的文档为例：

```
boolean openFile(String path)
{
  Intent intent = new Intent();
  Bundle bundle = new Bundle();
  bundle.putString(OPEN_MODE, READ_ONLY);
   //打开模式
  bundle.putBoolean(SEND_CLOSE_BROAD, true);
   //关闭时是否发送广播
  bundle.putString(THIRD_PACKAGE, selfPackageName);
   //第三方应用的包名，用于对改应用合法性的验证
  bundle.putBoolean(CLEAR_TRACE, true);
   //清除打开记录
  //bundle.putBoolean(CLEAR_FILE, true);
   //关闭后删除打开文件
  intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
 intent.setAction(android.content.Intent.ACTION_VIEW);
  intent.setClassName(packageName, className);

  File file = new File(path);
  if (file == null || !file.exists())
  {                  return false;
  }

  Uri uri = Uri.fromFile(file);
  intent.setData(uri);
  intent.putExtras(bundle);
  try
  {
       startActivity(intent);
  }
  catch (ActivityNotFoundException e)
  {
       e.printStackTrace();
       return false;
  }
  return true;
}
```

注：在支持http模式打开文档的版本下，可以使用如下代码打开http文件：

```
Intent intent = new Intent();
intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
intent.setAction(android.content.Intent.ACTION_VIEW);
Uri uri =
Uri.parse("http://10.20.138.164:8080/TomcatServlet/upload/a.doc");
intent.setData(uri);
try
{
startActivity(intent);
}
catch (ActivityNotFoundException e)
{
e.printStackTrace();
}
```

三、AIDL方式

1.方式简介

AIDL 方式启动 WPS 是指，通过绑定 Service，调用 WPS 对外暴露的接口，直接操作 WPS 的功能。如打开、保存、另存、打开手绘等。

上述两种操作方式各有利弊，使用第三方启动方式，简单方便，但是仅仅能控制一些打开文档的显示方式；使用 AIDL 方式打开可以操控的功能比较多，但是实现较为复杂。下面我们介绍一下具体的操作实现。

2.AIDL 操作实现

下面代码展示如何绑定 WPS 的 Service，通过获得绑定的 Service 对象，调用对应的接口，即可操作 WPS 的相关功能。

```
public static final String OFFICE_SERVICE_ACTION = "cn.wps.moffice.service.OfficeService";
public static final String PRO_OFFICE_SERVICE_ACTION =
"cn.wps.moffice.service.ProOfficeService";

void doBindService()
    {
        Intent intentOfficeService = new
        Intent(PRO_OFFICE_SERVICE_ACTION);
// PRO_OFFICE_SERVICE_ACTION是绑定service的IntentFilter，之前版本提供的
OFFICE_SERVICE_ACTION也可以使用，只不过为了避免和个人版的冲突，添加了一个新参数
        intentOfficeService.putExtra("DisplayView", true);
        bindService(intentOfficeService, mConnection,
BIND_AUTO_CREATE);
    }
    void doUnbindService()
    {
        if (mIsBound)
        {
            unbindService(mConnection);
            mIsBound = false;
        }
    }
//获取连接实例
private ServiceConnection connection = new ServiceConnection() {
    @Override
    public void onServiceConnected(ComponentName name, IBinder service)
    {
        mService = OfficeService.Stub.asInterface(service);
    }
    @Override
    public void onServiceDisconnected(ComponentName name)
    {
```

```
            mService = null;
        }
        };
```

注意：第三方程序如果需要使用 AIDL 方式操作 wps，需要将 moffice-service.jar 导入到自己的工程中，并在 eclipse 中选择 Java Build Path 的 Order and Export 中选中该 jar 包，这样编出的第三方应用程序才能正常操作控制 wps。

3.AIDL 接口介绍

【Word 组件】

（1）获得和打开文档接口

简介：该接口是获得和打开文档的操作，通过绑定 WPS 内部的 Service 即可获得 OfficeService 对象，然后可以调用这些接口打开指定文档

OfficeService 对象及方法摘要	
Documents	getDocuments();获取文档对象集合
Document	openDocument(String docPath, String password, Intent intent)：打开文字文档获得文档对象； 参数说明： docPath:要打开的文档路径 Password:WPS 加密文档的密码，如果文档没有使用 WPS 加密，传入”“； Intent：包装其他数据，传入不同的打开参数，实现不同的效果，文档具体打开参数见附录 1。
Document	newDocument(String newDocPath, in Intent intent); //功能：新建一个 doc 文档，newDocPath 是文件路径，intent 是 AIDL 操作文档的一些设置，与 openDocument 的 intent 相同作用

打开文档使用示例：

示例 1：

```
Intent intent = new Intent();
Bundle bundle = new Bundle();
bundle.putBoolean(Define.BACK_KEY_DOWN, true);//监听Back键事件,对外发送广播
intent.putExtras(bundle);
mDoc = mService.openDocument(path, "", intent);//mService为绑定文档后返回的对象
```

注：请将打开文档功能在子线程中进行，避免出现死锁现象

（2）Document 对象调用接口

【简介】该接口是 Document 对象调用的接口，要使用这些接口必须先通过 OfficeService 对象的 openDocument 方法获得文档对象。

【详细参数及方法介绍】

Document 对象及方法摘要	
void	**close()** ；关闭文档
String	getPath();获取文档的完整路径
String	getName();获取文档名
Page	**getPage（int index）；**获取文档第 n 页(索引从 0 开始)，**绑定 OfficeService 时设定显示 wps 界面时候不可调用该 方法**
int	getPageCount();获取文档的总页数，**绑定 OfficeService 时设定不显示 wps 界面时候可以调用该 方法**
Selection	getSelection（ ）；获取文档当前选区；**绑定 OfficeService 时设定显示 wps 界面时候可以调用该方法**
boolean	print(PrintSetting setting);打印输出文档，**绑定 OfficeService 时设定不显示 wps 界面时候，调用该方法实现打印。**
boolean	save(boolean noPrompt)；保存文档(noPrompt: 该参数是用来设置是否自动保存的 ,但是暂未实现，请将该参数设为 true。)
boolean	saveAs(String filePath, SaveFormat saveFormat, String password, String writePassword)； 文档另存为其他格式时候调用； fileName：另存文档的全路径； SaveFormat saveFormat,：另存文档格式 password,：绑定 OfficeService 设定显示 wps 界面时候无效 传入”“ writePassword：绑定 OfficeService 设定显示 wps 界面时候无效 传入”“
void	hiddenToolBar(）：隐藏 Edit Toolbar
void	hiddenMenuBar（ ）：隐藏 MenuBar
boolean	isModified()：文档是否修改
void	showHandWriteComment()：打开手绘批注模式
void	closeHandWriteComment()：关闭手绘批注模式
Shapes	Shapes getShapes();获取 Shapes，插入文件时可用到
boolean	fairCopy(String path, String password); //清稿功能 ：接受所有修订并另存到指定路径，原文档保持当前状态
void	enterReviseMode(); //进入修订模式，在普通状态下会进入修订模式
void	exitReviseMode(); //退出修订模式，在修订状态下会生效
void	acceptAllRevision(); //接受所有修订，该方法在修订状态下会有效果
void	signature(float left, float top, float width, float height, String sealFilePath, String

	password)；
	//功能：对文档进行电子签章
	//sealFilePath 图片地址，
	// password 保护密码
	//该方法暂时不可用，开发中

void	undo(); //撤销功能
void	redo(); //还原功能
void	clearAllComments(); //删除文档中的所有批注

int	getCurrentPageNum(int posAtScreen); //获得当前显示页面的页码
	//posAtScreen:代表以屏幕的哪个位置作为标准，取值及含义如下
	static final int HEAD_PAGE_NUMBER = 0; //获取屏幕顶部对应页的页码
	static final int MIDDLE_PAGE_NUMBER = 1; //获取屏幕中间对应页的页码
	static final int BOTTOM_PAGE_NUMBER = 2; //获取屏幕底部对应页的页码

boolean	saveCurrentPageToImage(String path, in PictureFormat picFormat, int index, int item, int bkColor, int width, int height);
	//对当前页面进行截图保存
	//path 为保存路径，picFormat 为要保存的图片格式，index 为页码，暂时不用，填写 0 即可，item 为是否截取修订界面，bkColor 为截图背景，width 和 height 分别为截图的长宽

Int	getLength(); //获取文档的长度
int	getScrollY(); //获得文档已经滚过的 Y 坐标长度，比如向下拖动了 100 个单位坐标，该方法返回值即为 100
int	getScrollX() //获得文档已经滚过的 X 坐标长度
void	pageUp(); //上一页
void	pageDown(); //下一页
void	addDocumentVariable(String name, String value); //添加文档自定义属性，name 是名称，value 是自定义属性的值
String	getDocumentVariable(String name); //获取文档自定义属性的值，如果 name 对应的 value 存在则返回，不存在返回 null

void	protect(String password, int protectionType, boolean styleLockEnforced);
	//password 文档保护的密码，protectionType文档保护类型，styleLockEnforced是否强制保护，直接设置为true即可
	protectionType 的取值类型在jar包的ProtectionType，如下面四种
	public final static short *TRACKEDCHANGES* = 0;
	public final static short *COMMENTS* = 1;
	public final static short *FORMS* = 2;
	public final static short *READONLY* = 3;
	public final static short *NONE* = 7;

void	unprotect(String password); //文档解保护，password为密码
boolean	isProtectOn(); //文档是否处于保护状态，只读、修订、批注保护模式都属于保护状态
int	getProtectionType(); //获取文档保护类型
int	getCommentNumber(); //获取文档中批注个数
boolean	isInRevisionMode(); //是否处于修订模式
String	getUser(); //获得当前修订批注手绘的作者名
boolean	isInHandWriterMode(); //是否是手绘模式
void	toggleInkFinger(); //菜单栏，笔->使用手指功能， 墨迹相关接口
void	toggleToPen(); //菜单栏，笔->笔功能， 墨迹相关接口
void	toggleToHighLightPen(); //菜单栏，笔->荧光笔功能， 墨迹相关接口
void	toggleToEraser(); //菜单栏，笔->橡皮擦功能， 墨迹相关接口
void	setInkColor(int color); //菜单栏，笔->颜色功能， 墨迹相关接口
void	setInkThick(float strokeWidth); //菜单栏，笔->粗细功能， 墨迹相关接口
int	getInkColor(); //墨迹颜色， 墨迹相关接口
float	getInkPenThick(); //笔的粗细， 墨迹相关接口
float	getInkHighLightThick(); //荧光笔的粗细， 墨迹相关接口

【代码示例】另存文档

```
SaveFormat saveFormat= SaveFormat.DOC;        //    另存的文档格式
mDoc.saveAs(path, saveFormat, null, null);//另存文档
```

注：mDoc是示例1中打开文档得到的文档对象，该对象可以调用上表中的方法

（3）选区的操作接口

【简介】通过调用 Document 的 getSelection 方法获得对象，对 wps 选区进行操作

【详细参数及方法介绍】

Selection 对象及方法摘要	
void	copy(); 复制选区
void	cut(); 剪切选区
void	Paste();粘贴内容到选区
void	typeText(String text) 将文本插入所选选区
String	getText();返回选区中的所有字符
void	insertParagraph();用新段落替换选区内容
InlineShapes	getInlineShapes(); 获取选区内嵌入式对象
float	getLeft(); //获取选取当前的位置： Left是相对页面
float	getTop(); //获取选取当前的位置： Left是相对页面
int	getStart()； //获取当前selection的startCp
int	getEnd(); //获取当前selection的EndCp
void	setSelection(int start, int end, boolean scrollToVisible); //设置选区从start到end，scrollToVisible为是否滚动到选区
boolean	addComment(String text); //在当前选区添加一个批注，text是批注的内容，返回值表示添加批注是否成功
Font	getFont(); //获取当前选区的字体对象，以便可以设置字体，具体见下面Font对象的使用

【代码示例】复制选区

```
mDoc.getSelection().copy();
```

注：在复制文字的时候，需要在wps中选中一段文字，这样才能复制文字。

（4）嵌入式对象的操作接口

【简介】该接口是嵌入式对象的操作接口，首先需要使用 Selection 对象的 getInlineShapes()接口获得对象，才可以插入图片。

【详细参数及方法介绍】

InlineShapes 对象及方法摘要	
void	**addPicture(String** filePath); 插入 filepath 指向的图片；

【代码示例】

```
mDoc.getSelection().getInlineShapes().addPicture(picPath);
//picPath 为你要插入图片的路径
```

（5）对象的操作接口

【简介】该接口是 shapes 对象的操作接口，首先需要使用 Document 对象的 getShapes()接口获得对象，才可以插入图片。

【详细参数及方法介绍】

Shapes 对象及方法摘要	
void	addPicture(String filePath, boolean linkToFile, boolean saveWithDocument,float left, float top, float width, float height, int cp, in WrapType wrapType); filePath: 图片路径 inkToFile:如果为 True，则将图片链接到创建它的文件；如果为 False，则使图片成为该文件的独立副本。默认值为 False。 saveWithDocument:如果为 True，则将链接的图片与文档一起保存。默认值为 False。 left ：图片左边缘 top：图片上边缘 width：图片宽度 height：图片高度 cp：图片插入点（如果是空文档，值应设为0） WrapType：绕排类型，只能取值：Square, Tight, TopOfText, BottomOfText
Void	addTextBox(**int** x, **int** y, **int** width, **int** height, **int** lineColor, **boolean** isShowLine, **int** fillColor, **boolean** isTransparent, **int** fontColor, **float** fontSize, String fontName, String content) 添加文本框 X，文本框左上角的x坐标 Y，文本框左上角的y坐标 Width，文本框宽 Height，文本框高 lineColor，文本框线条颜色 isShowLine，边框是否透明 fillColor，填充颜色 isTransparent，是否透明 fontcolor，文本框中字体颜色 fontSize，文本框中字体大小 fontName,文本框中字体名称 content，文本框中的内容 示例代码： *mDoc*.getShapes().addTextBox(100,100,50, 50, 0x00ff0000, **false**, 0x0000ff00, **false**, 0x0000ff, (**float**) 20.5, "宋体", "通过！");

【代码示例】（插入图片）

mDoc.getShapes().addPicture(filePath, linkToFile, saveWithDocument, mLeaf, mTop, width, height, mCp, WrapType.valueOf(wrapType));

其中部分参数如下：

float mLeaf = mDoc.getSelection().getLeft();

float mTop = mDoc.getSelection().getTop();

int mCp = mDoc.getSelection().getStart(); //获取选区的开始位置，始终在选区的开始插入图片，getEnd 可以获取选区的结束为止，其他参数不变

Font 对象及方法摘要	
void	setBold(boolean bBold); //设置当前选区字体加粗，bBold，是否加粗
void	setItalic(boolean bItalic); //设置当前选区字体倾斜，bItalic，是否倾斜
void	setTextColor(int color); //设置字体颜色，color，为要设置的颜色
void	setName(String name); //设置选区字体，name为要设置的字体名称
void	setSize(float size); //设置字体大小，size为字体的大小
void	setStrikeThrough(); //设置选区文字单行删除线
void	setDoubleStrikeThrough(); //设置选区文字双行删除线
void	setNoneStrikeThrough(); //设置选区文字无删除线

【代码示例】（设置选区字体粗体）

mDoc.getSelection().getFont().setBold(true);

//设置当前选区字体加粗

Page 对象及方法摘要	
float	getWidth(); //获取页面宽度
float	getHeight(); //获取页面高度
boolean	saveToImage(String path, in PictureFormat format, int quality, float width, float height, int dpi, int item);

	//把当前页保存为图片,参数含义见下面示例：

Bitmap	//把当前页保存为 bitmap
	saveToBitmap(float width, float height, int dpi, int item);

【代码示例】（把页码导出为图片）

```
final int QUALITY = 100;
final int DPI = 64;
final int PRINT_ITEM = PrintOutItem.wdPrintContent;
PictureFormat picFormat = PictureFormat.PNG;//图片格式
loat width = mDoc.getPage(0).getWidth();
float height = mDoc.getPage(0).getHeight();

//方法参数：第一个为图片保存路径吗，第二个为图片保存格式，第三个为图片质量，第四个为图片宽度，
//第五个为图片高度，第六个是 DPI，第七个是图片保存选项
//可以直接参考这段代码的默认参数使用即可实现截图
mDoc.getPage(0).saveToImage(picPath, picFormat, QUALITY, width, height, DPI, PRINT_ITEM);
```

【Excel 组件】

（1）获得和打开文档接口

简介：该接口是获得和打开文档的操作，通过绑定 WPS 内部的 Service 即可获得 OfficeService 对象，然后可以调用这些接口打开指定文档

OfficeService 对象及方法摘要	
Workbooks	getWorkbooks(); //获取表格对象集合；暂时不支持带界面的操作
Workbook	openWorkbook(String path, String password); //打开一份excel文档； 暂时不支持带界面的操作
Workbook	newWorkbook(String newFilePath, in Intent intent); //新建一个excel文档

（2）**Workbooks** 对象调用接口

【简介】该接口是 Workbooks 对象调用的接口

【详细参数及方法介绍】

Workbooks 对象及方法摘要	

Workbook	openBookEx(String path, String password, in Intent intent);打开表格文档，获取表格文档对象；
	// 打开一个工作簿，会启动spreadsheet程序
	// path:文档路径
	// password:保留参数
	// intent:调用spreadsheet程序可提供其他参数,可选参数
	// 返回值:返回一个工作簿。null表示打开工作簿失败

（3）**Workbook** 对象调用接口

【简介】该接口是 Workbook 对象调用的接口，要使用这些接口必须先通过 OfficeService 对象的 openBookEx 方法获得文档对象。

【详细参数及方法介绍】

Workbook 对象及方法摘要	
String	getName();//获取工作簿的名字
Worksheet	getActiveSheet();// 获取工作簿中活动的工作表
Worksheet 对象及方法摘要	
String	getName() ;// 获取工作表的名字

boolean	saveToImage(String path, in PictureFormat format, int quality, float scale);
	// 将工作表内容导出为图片到指定目录中
	// path:导出文件夹路径
	// format:导出图片的格式类型，见PictureFormat定义
	// quality:打印质量，从1到100,数值越大打印质量越高
	// scale:打印缩放比，默认为1

	// 返回值：true表示导出图片成功；false表示失败

boolean	saveCurrentPage(String path, in PictureFormat format);
	// 将当前页导出为图片。当前页定义为左上角单元格所处打印分页的页面，当左上角单元格不在打印范围内，不会保存当前页。
	// path：导出路径
	// format:导出图片的格式类型，见PictureFormat定义
	// 返回值：true表示导出成功，false表示导出失败

int	getCurrentPageIndex();
	// 获取左上角单元格所在分页中的页码,页码从0开始计数。
	// 返回值：-1表示左上角单元格不在打印范围内，此时返回值不能用于打印。页码大于等于0时为有效值。

int	getPageCount(); // 获取当前工作表打印的总页数

【PPT 组件】

（1）获得和打开文档接口

简介：该接口是获得和打开文档的操作，通过绑定 WPS 内部的 Service 即可获得 OfficeService 对象，然后可以调用这些接口打开指定文档

OfficeService 对象及方法摘要	
Presentations	getPresentations();获取演示文档对象集合
Presentation	openPresentation(String filePath, String password, in Intent intent); 参数说明： filePath， 文件路径 password， 加密密码，暂不支持，请传入“”。 Intent，同 Document
Presentation	newPresentation(String newFilePath, in Intent intent); //新建一个演示文档

（2）**Presentations** 对象调用接口

【简介】该接口是 Presentations 对象调用的接口

【详细参数及方法介绍】

Presentations 对象及方法摘要	
Presentation	openPresentation(String filePath, String password, in Intent intent); //打开一个文档
void	closePresentations(); //关闭集合中的所有文档

（3）**Presentation** 对象调用接口

【简介】该接口是 Presentation 对象调用的接口，要使用这些接口必须先通过 OfficeService 对象的 openPresentation 方法获得文档对象。

【详细参数及方法介绍】

Document 对象及方法摘要	
void	transitionPre(); //切换到上一页
void	transitionNext(); //切换到下一页
void	close(); //关闭文档
int	getCurrentPageIndex(); //获取当前页码(索引从 0 开始)
int	getPageCount(); //获取 slide 的个数
String	exportCurPageToImage(String path, in PictureFormat picFormat); //保存当前页为图片，path 为保存路径文件夹，返回输出图片的全路径，null 表示失败
List<String>	exportImage(String path, in PictureFormat picFormat, PrintProgress progress); //将所有页导出到path所在路径文件夹中，并返回输出图片的路径

【PDF 组件】

（1）获得和打开文档接口

简介：该接口是获得和打开文档的操作，通过绑定 WPS 内部的 Service 即可获得 OfficeService 对象，然后可以调用这些接口打开指定文档

OfficeService 对象及方法摘要	
PDFReaders	getPDFReaders(); 获取当前打开的 PDF 文档对象集合
PDFReader	openPDFReader(String filePath, String password, Intent intent) 参数说明： filePath， 文件路径 password， 加密密码，暂不支持，请传入“”。 Intent，同 Document

2）PDFReader 对象调用接口

【简介】该接口是 PDFReader 对象调用的接口，要使用这些接口必须先通过 OfficeService 对象的 openPDFReader 方法获得文档对象。

【详细参数及方法介绍】

Document 对象及方法摘要	
void	close(); // 关闭文档
String	getPath(); // 获取文档的完整路径
void	transitionPre(); // 滚动至上一页
void	transitionNext(); // 滚动至下一页