

Android OfficeSDK 集成使用指南

橄榄办公

2018. 07. 09

目录

一、简介.....	1
1. 产品简介.....	1
2. 支持格式.....	2
二、开发环境设定.....	2
1. 开发所需操作系统:	2
2. 开发所需环境:	2
3. 开发所需工具:	2
三、使用示例工程.....	2
1. 导入示例工程.....	2
2. 编译执行示例工程.....	3
四、SDK 架构与工作原理	4
1. Office 文档格式简介	4
2. 实现概览.....	5
2.1. 文档视图:	5
2.2. 文档控制器:	5
五、使用授权与自定义水印.....	6
1. 获取授权文件.....	6
2. 导入授权文件.....	6
3. 用户设置自定义水印.....	6
六、联系方式.....	9
1、技术咨询及商务合作联系方式.....	9
2、产品官方网站.....	9

一、简介

1. 产品简介

Android Office SDK (以下简称 SDK)是一款 Android 平台的 Office 文档查看产品。通过集成 SDK, 开发者可以在应用中轻易实现查看 Office 文档功能。



SDK 包括以下子产品:

- OfficeSDK-Word For Android
- OfficeSDK-XLS For Android
- OfficeSDK-PPT For Android
- OfficeSDK-PDF For Android

2. 支持格式

SKD 目前支持查看以下 Office 文档格式:

- 电子文档: .doc 或 .docx 及 .txt
- 电子表格: .xls 或 .xlsx
- 演示文稿: .ppt 或 .pptx
- PDF 文档: .pdf

二、开发环境设定

1. 开发所需操作系统:

Windows 7(32/64 位) 或 Windows8(32/64 位)

Mac OS X 10.5.8 或以上

Linux 2.6 或以上

2. 开发所需环境:

JDK 6 或以上及 Android Framework 4.0.3 或以上

3. 开发所需工具:

Eclipse 3.4 或以上

Android Studio 1.5 或以上。

三、使用示例工程

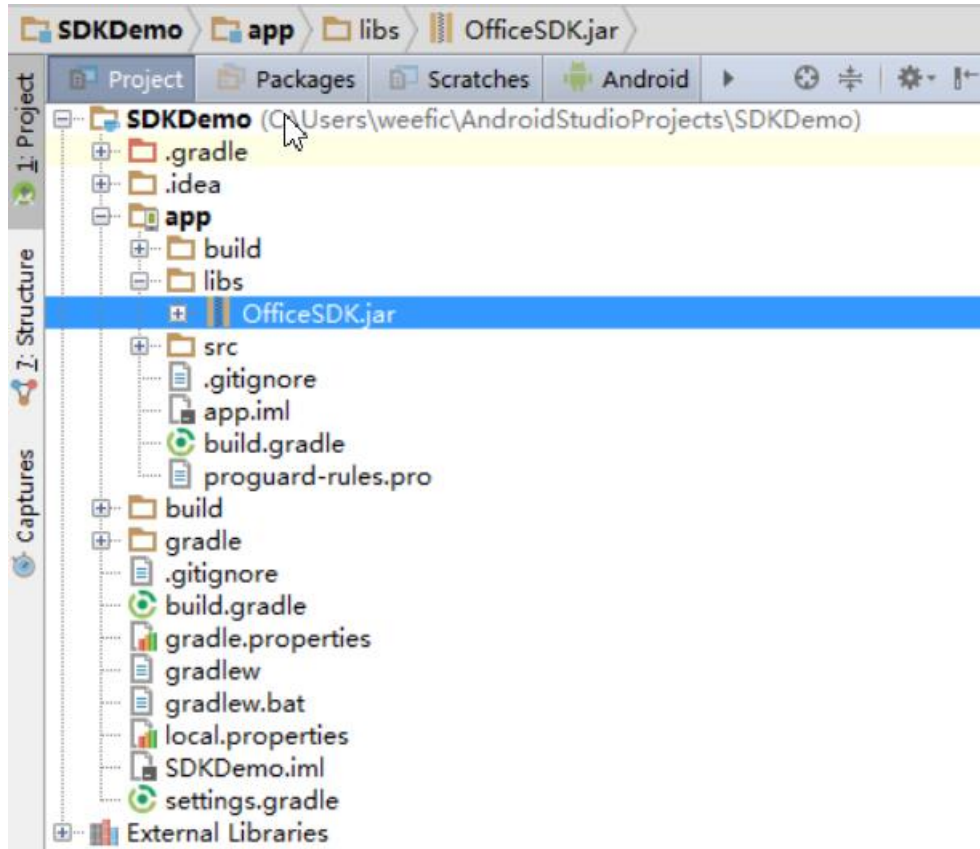
在进行开发前, 我们先向您展示我们提供的示例工程来增加您对 SDK 的认识。

1. 导入示例工程

我们需要使用开发工具及移动设备来运行示例工程, 根据 Android 官方指引, 我们向您

推荐使用 Android Studio 作为开发使用 SDK 的工具。而较早前因官方已经不再推荐使用 Eclipse，故此文档中不再列出 Eclipse 示例教程，若有需要，请联络技术支持。

在完成上述开发环境设定后，打开 Android Studio 并导入示例工程。该示例工程在 Android Studio 3.0 下创建，在其它版本的 AndroidStudio 中可能有兼容性问题。导入工程后，如下图所示：



2. 编译执行示例工程

示例工程中的 SDK 库文件可能不是最新的，您需要联络我们的技术支持人员以获取最新的 SDK。请将最上图中 OfficeSDK.jar 替换为最新的 SDK 库文件。此时，整个示例工程已可以在模拟器或移动设备中运行。

示例工程的主界面是一个文件浏览器，该文件浏览器将访问您的移动设备的 SDCARD 目录，并列出所有支持的 Office 文档及文件夹。您可以选择在移动设备上的文件进行查看。

注意，由于是示例工程，查看 Office 文档时会显示出我司的水印信息。如下图所示，分别是文件浏览器、查看电子文档、查看电子表格、查看演示文稿：（注：图中按钮作为 SDK 外控件，用于自定义控制 SDK）



1. Introduction

Hi, thanks for using CoolOffice Premium by ZySoft.

CoolOffice Premium is an Office application for viewing and editing Word Documents, Excel Spreadsheets and PowerPoint slides. It is developed for mobile users and can be used on smartphones and tablets. Now you can easily take your work and ideas away with you whether at home, at a meeting, at Starbucks or on a bus.

2. Birth and Growth

CoolOffice was born in 2010 with a concept of letting your Office free from PC and breaking away from the obsolete working style. It first came to the world as CoolOffice Basic for Android which was designed for viewing Word 2003 Documents and Excel 2003 Spreadsheets. Then it was upgraded to CoolOffice Standard with the extra added features.



四、SDK 架构与工作原理

1. Office 文档格式简介

微软 Office 文档的存储格式主要分为两种：

Office 97-2003 采用的二进制复合文档格式（简称 o3 格式）

Office 2007 以后的 OpenXML 格式（简称 o7 格式）

2. 实现概览

SDK 在设计及实现上遵循 MVC 模式，我们向开发者提供了文档视图及文档控制器以便开发者能够简单地使用 SDK。

2.1. 文档视图：

文档视图用于将文档展示给用户，在 SDK 中以 `DocumentView` 作为表示。

`DocumentView` 相当于 Android 中的自定义控件，其将监听用户的输入及操作，并将计算结果呈现给用户。您可以使用以下代码创建一个 `DocumentView`：

```
DocumentViewFactory.newWordView(context);
DocumentViewFactory.newSpreadsheetView(context);
DocumentViewFactory.newPresentationView(context);
```

值得注意的是，目前 SDK 未能支持使用 XML 布局创建文档视图。开发者需要通过判断档格式来创建对应格式的文档视图，在视图创建完成后，将文档视图加入到要显示的容器 `View` 中即可。

2.2. 文档控制器：

文档控制器用于开发者控制文档视图。

在 SDK 中以 `DocumentViewController` 作为表示。在创建 `DocumentView` 后，通过以下代码可以获得文档控制器：`DocumentView view = DocumentViewController controller = view.getController();`

```
获得控制器后，此时可以对整个文档视图进行控制。开发者可以通过以下代码打开 Office 文档：
boolean encrypted = controller.checkEncrypted(...); // 传入文件
if(encrypted) ...; /* 处理加密文档 */
else controller.loadFile(..., ...); /* 打开文档 */
```

执行上述代码后，Office 文档将会把文档呈现到文档视图中，此时基本的打开文档操作已经完成。

如需要更多的额外操作，也可以在通过文档控制器来完成，如所缩放文档、监听事件等等。更多的操作请查看 [Android OliveOffice SDK API docs](#)。

五、使用授权与自定义水印

声明：授权文件是我司保护产品的重要文件，为保证您的应用安全，我们倡导使用正版的 SDK 进行产品开发。

1. 获取授权文件

请联络我司的商务部门，您可以从商务部门获取价格、服务支持等等信息。在您完成付款后，授权文件将会交付给开发者。

联系邮箱：635512001@qq.com

联系 QQ：635512001

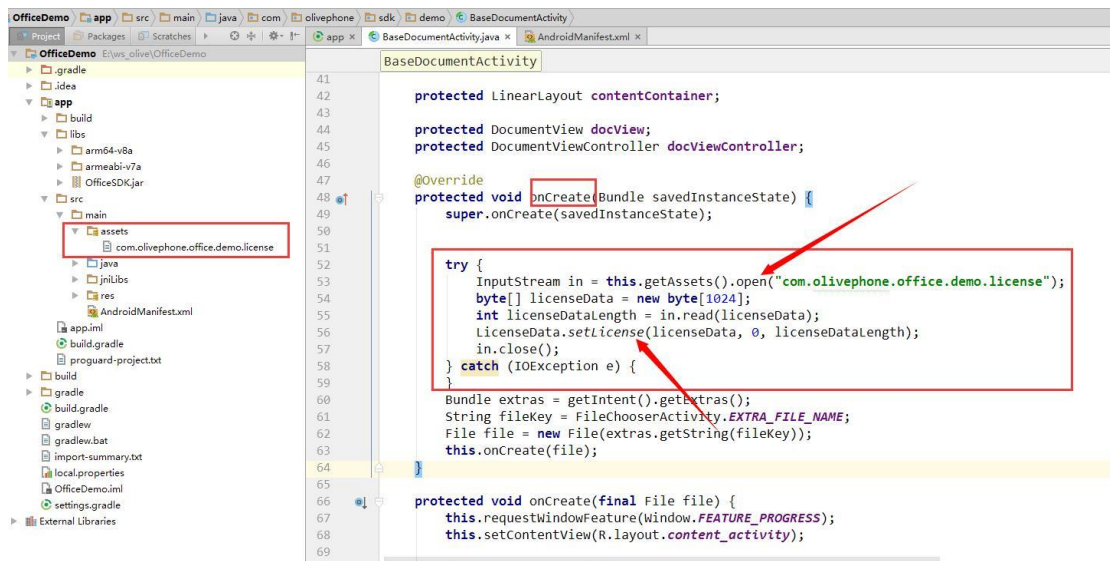
联系电话：15071327475

2. 导入授权文件

开发者可以通过以下代码将授权信息导入至 SDK 中：

```
byte[] licenseData = LicenseData.setLicense(licenseData);
```

该代码可以在应用启动时执行或者在打开 Office 文档时执行，仅首次执行有效。



3. 用户设置自定义水印

使用自定义水印，需要在自己的 Activity 中导入如下文件：

```
import com.olivephone.office.CustomWatermarkHelper;
```

3.1 使用方法

```
CustomWatermarkHelper.setCustomWatermark(String text, int color, float textSize, int lineSpace)
```

```
CustomWatermarkHelper.setCustomWatermark(customWaterMark,Color.RED,36,100);
```

```

48= @Override
49 protected void onCreate(Bundle savedInstanceState) {
50     super.onCreate(savedInstanceState);
51
52
53     try {
54         InputStream in = this.getAssets().open("com.olivephone.demo.lic");
55         byte[] licenseData = new byte[1024];
56         int licenseDataLength = in.read(licenseData);
57         LicenseData.setLicense(licenseData, 0, licenseDataLength);
58         in.close();
59
60         String customWaterMark = "内部文件，请勿外传";
61         /**
62          * CustomWatermarkHelper.setCustomWatermark(String text, int color, float textSize, int lineSpace)
63          * 设置用户自定义水印信息
64          * @param text 水印文本信息
65          * @param color 水印文本颜色 默认为android.graphics.Color.RED
66          * @param textSize 水印文本大小 默认大小36
67          * @param lineSpace 水印行间距 默认高度80
68          */
69         CustomWatermarkHelper.setCustomWatermark(customWaterMark,Color.RED,36,100);
70     } catch (IOException e) {
71         e.printStackTrace();
72     }
73

```

3.2 自定义水印方法参数说明

方法名	参数	参数类型	参数说明
setCustomWatermark	text	String	自定义水印文本内容
	color	int	水印文本颜色 默认为 android.graphics.Color.RED
	textSize	float	水印文本大小。默认大小 36.0
	lineSpace	int	水印文本行间距。默认高度 80 像素。

3.3 自定义水印使用效果图

This is a trial version.
www.olivephone.com
sales@olivephone.com

运营风险控制管理办法

第一章 总则

第一条 为保障基金运营和投资业务的安全运作和管理，加强公司及所管理基金的内部风险管理，规范基金运营和投资行为，提高风险防范能力，有效防范与控制基金运营和投资项目运作风险，根据《公司章程》等法律法规的相关规定，特制定本办法。

第二条 本办法所称风险控制，是指对基金运营和项目投资中的各种投资风险进行识别、评估，并在基金运营、项目管理、项目退出中实施动态风险监控，提出解决方案。

第三条 风险控制原则

- (1) 全面性原则：风险控制制度应覆盖基金运营和投资业务的各项工作和各级人员，并渗透到决策、执行、监督、反馈等各个环节；
- (2) 审慎性原则：内部风险控制的核心是有效防范各种风险，公司部门组织的构成、内部管理制度的建立要以防范风险、审慎经营为出发点；
- (3) 独立性原则：风险控制工作应保持高度的独立性和权威性，并贯彻到基金运营和业务的各具体环节；
- (4) 有效性原则：风险控制制度应当符合国家法律法规和监管部门的规章，具有高度的权威性，成为所有员工严格遵守的行动指南；执行风险管理制度不能存在任何例外，任何员工不得拥有超越制度或违反规章的权力；
- (5) 适时性原则：应随着国家法律法规、政策制度的变化，公司经营战略、经营方针、风险管理理念等内部环境的改变，以及公司业务的发展，及时对风险控制制度进行相应修改和完善；



六、联系方式

1、技术咨询及商务合作联系方式

电话：15071327475

QQ：635512001

邮箱：635512001@qq.com

2、产品官方网站

橄榄办公 OfficeSDK： <http://sdk.olivephone.com/>

橄榄办公官网： <http://www.olivephone.com/>

OliveOffice Premium SDK

For Android API

➤ DocumentView

com.olivephone.sdk.DocumentView

显示 Word/Excel/PowerPoint 文档内容和处理用户滚动/缩放操作的 View。

1. Summary

Method	Summary
getController()	获取用于操作该 View 的控制器
asView()	将该 View 转换为 android.view.View
destroy()	清除当前文档, 释放资源

2. Details

➤ getController

DocumentViewController getController()

获取用于操作该 View 的控制器

Parameters:

无

Return:

Word、Excel 或者 PowerPoint 的 View 控制器

➤ asView

View asView()

将该 View 转换为 android.view.View

Parameters:

无

Return:

android.view.View 对象

➤ destroy

void destroy()

清除当前文档, 释放资源

Parameters:

无

Return:

无

➤ DocumentViewController

com.olivephone.sdk.DocumentViewController

DocumentView 的控制器，提供文档加载/滚动/缩放等交互操作。

1. Summary

Method	Summary
checkEncrypted(File file)	检查文件是否加密并初始化文档
loadFile(String password, LoadListener listener)	开始加载文档
stopLoading()	停止文档加载过程，并清空文档
setPinchZoomEnabled(boolean enabled)	设置是否启用两指手势缩放
zoomIn()	放大
zoomOut()	缩小
goToBottom()	跳转至文档结束位置
goToTop()	跳转至文档开始位置
getScrollX()	获取当前的水平滚动位置
getScrollY()	获取当前的垂直滚动位置
getMaxScrollX()	获取水平方向可滚动的最大值
getMaxScrollY()	获取垂直方向可滚动的最大值
scrollTo(int x, int y, boolean animate)	滚动到指定位置

2. Details

➤ checkEncrypted

boolean checkEncrypted(File file)
检查文件是否加密并初始化文档

Parameters:

file: 需要打开的文件

Return:

如果文件加密则为 true，否则为 false

➤ loadFile

void loadFile(String password, LoadListener listener)
开始加载文档

Parameters:

password: 打开文件的密码，若文件未加密则为 null
listener: 文档加载监听器

Return:

无

➤ stopLoading

void stopLoading()
停止文档加载过程，并清空文档

Parameters:

无

Return:

无

➤ **setPinchZoomEnabled**

void setPinchZoomEnabled(boolean enabled)

设置是否启用两指手势缩放

Parameters:

Enabled: 是否启用

Return:

无

➤ **zoomIn**

void zoomIn()

放大

Parameters:

无

Return:

无

➤ **zoomOut**

void zoomOut()

缩小

Parameters:

无

Return:

无

➤ **goToBottom**

public void goToBottom()

跳转至文档结束位置

Parameters:

无

Return:

无

➤ **goToTop**

public void goToTop()

跳转至文档开始位置

Parameters:

无

Return:

无

➤ **getScrollX**

int getScrollX()
获取当前的水平滚动位置

Parameters:

无

Return:

当前的水平滚动位置

➤ **getScrollY**

int getScrollY()
获取当前的垂直滚动位置

Parameters:

无

Return:

当前的垂直滚动位置

➤ **DocumentViewFactory**

com.olivephone.sdk.DocumentViewFactory

创建指定文档格式 DocumentView 的工厂类。

1. **Summary**

Method	Summary
newWordView(Context context)	创建一个查看 Word 文档的 DocumentView
newExcelView(Context context)	创建一个查看 Excel 文档的 DocumentView
newPPTView(Context context)	创建一个查看 PowerPoint 文档的 DocumentView

2. **Details**

➤ **newWordView**

public static DocumentView newWordView(Context context)
创建一个查看 Word 文档的 DocumentView

Parameters:

context: android.content.Context 对象

Return:

查看 Word 文档的 DocumentView

➤ **newExcelView**

public static DocumentView newExcelView(Context context)
创建一个查看 Excel 文档的 DocumentView

Parameters:

context: android.content.Context 对象

Return:

查看 Excel 文档的 DocumentView

➤ **newPPTView**

public static DocumentView newPPTView(Context context)

创建一个查看 PowerPoint 文档的 DocumentView

Parameters:

context: android.content.Context 对象

Return:

查看 PowerPoint 文档的 DocumentView

➤ LoadListener

com.olivephone.sdk.LoadListener

用于文档加载的事件监听器。

1. Summary

Method	Summary
onProgressChanged(int newProgress)	文件加载进度发生变化时调用
onWrongPassword()	密码有误时调用
onError(String description, Throwable throwable)	加载文档发生错误时调用

2. Details

➤ onProgressChanged

void onProgressChanged(int newProgress)

文件加载进度发生变化时调用

Parameters:

newProgress: 新的进度值

Return:

无

➤ onWrongPassword

void onWrongPassword()

密码有误时调用

Parameters:

无

Return:

无

➤ onError

void onError(String description, Throwable throwable)

加载文档发生错误时调用

Parameters:

description: 错误描述

throwable: 相关异常

Return:

无

➤ **WordViewController**

com.olivephone.sdk.WordViewController

针对 Word 文档的控制器。

1. Summary

Method	Summary
listBookmarks()	获取文档中书签名称的列表
goToBookmark(String bookmark)	根据给定的书签名称跳转到指定位置

2. Details

➤ **listBookmarks**

```
public List<String> listBookmarks()
```

构造函数

Parameters:

无

Return:

书签名称列表，如果文档中没有书签则返回 null

➤ **goToBookmark**

```
public void goToBookmark(String bookmark)
```

根据给定的书签名称跳转到指定位置

Parameters:

bookmark: 书签名称

Return:

无

➤ **WorkbookViewController**

com.olivephone.sdk.WorkBookViewController

针对 Excel 工作薄的控制器。

1. Summary

Method	Summary
setPrintCut(int typeAreaWidth, int typeAreaHeight, Rect margins)	设定用于切割工作表的参数
changeSheet(int index)	设置指定的 sheet 到 TableView 中
getSheetsName()	获取所有表单的名字
scrollBy(int x, int y)	移动视图

2. Details

➤ **setPrintCut**

```
public void setPrintCut(int typeAreaWidth, int typeAreaHeight, Rect margins)
```

设定用于切割工作表的参数。在对工作表进行分页转换时将依据此处传入的版心宽度/

高度参数进行切, 并在输出图片时添加边距

Parameters:

typeAreaWidth: 版心宽度

typeAreaHeight: 版心高度

margins: 边距, 该参数的 left/top/right/bottom 字段分别对应左/上/右/下边距

Return:

无

➤ **changeSheet**

public void changeSheet(int index)

设置指定的 sheet 到 TableView 中

Parameters:

index: sheet 索引

Return:

无

➤ **getSheetsName**

public ArrayList<String> getSheetsName()

获取所有表单的名字

Parameters:

无

Return:

所有表单的名字列表

➤ **scrollBy**

public void scrollBy(int x, int y)

移动视图

Parameters:

x: 水平坐标

y: 垂直坐标

Return:

无

➤ **PresentationController**

com.olivephone.sdk.PresentationController

针对 PowerPoint 幻灯片的控制器。

3. **Summary**

Method	Summary
setVerticallyFlipPage()	设定为垂直方向翻页
setHorizontallyFlipPage()	设定为水平方向翻页
setAutoDirectionFlipPage()	设定为自动方向翻页
getPageCount()	获取总页数

prevPage()	跳转至上一页
nextPage()	跳转至下一页
gotoPage(int pageNumber)	跳转到指定页

4. Details

➤ **setVerticallyFlipPage**

public void setVerticallyFlipPage()

设定为垂直方向翻页

Parameters:

无

Return:

无

➤ **setHorizontallyFlipPage**

public void setHorizontallyFlipPage()

设定为水平方向翻页

Parameters:

index: sheet 索引

Return:

无

➤ **setAutoDirectionFlipPage**

public void setAutoDirectionFlipPage()

设定为自动方向翻页

Parameters:

无

Return:

无

➤ **getPageCount**

public int getPageCount()

获取总页数

Parameters:

无

Return:

总页数

➤ **prevPage**

public boolean prevPage()

跳转至上一页

Parameters:

无

Return:

跳转至上一页成功返回 true, 失败返回 false

➤ **nextPage**

```
public boolean nextPage()
```

跳转至下一页

Parameters:

无

Return:

跳转至下一页成功返回 `true`，失败返回 `false`

➤ **gotoPage**

```
public boolean gotoPage(int pageNumber)
```

跳转到指定页

Parameters:

`pageNumber`: 页码，编号从 1 开始

Return:

跳转到指定页成功返回 `true`，失败返回 `false`