

zm.blog

Knowledge is power.

## Android加载PDF文件的使用

📅 2018-07-31 | 📁 [Android](#) | 👁 30

### Android PdfViewer

AndroidPdfViewer 1.x可在AndroidPdfViewerV1 repo上获得，可以独立开发。版本1.x使用不同的引擎在画布上绘制文档，因此如果您不喜欢2.x版本，请尝试1.x。

图书馆在Android上显示的PDF文档，用 animations， gestures， zoom 和 double tap 支持。它基于PdfiumAndroid来解码PDF文件。适用于API 11（Android 3.0）及更高版本。在Apache License 2.0下获得许可。

#### 3.1.0-beta.1有什么新功能？

- 合并拉取请求 # 557用于捕捉页面（逐页滚动）
- 合并拉出请求 # 618用于夜间模式
- 合并拉取请求 # 566 OnLongTapListener
- 将PdfiumAndroid更新为1.9.0，而 c++\_shared 不是使用 gnustdl\_static
- 更新Gradle插件
- 将编译SDK和支持库更新到26
- 将最低SDK更改为14

#### 3.0 API的变化

- 换成 Contants.PRELOAD\_COUNT 了 PRELOAD\_OFFSET
- 删除 PDFView#fitToWidth()（没有参数的变体）
- 删除 Configurator#invalidPageColor(int) 方法，因为无法呈现无效页面
- 从 OnRenderListener#onInitiallyRendered(int) 方法中删除了页面大小参数，因为文档可能具有不同的页面大小
- 删除 PDFView#setSwipeVertical() 方法

### 安装

添加到*build.gradle*:

```
compile 'com.github.barteksc:android-pdf-viewer:3.1.0-beta.1'
```

或者如果你想使用更稳定的版本:

```
compile 'com.github.barteksc:android-pdf-viewer:2.8.2'
```

库在jcenter存储库中可用, 可能它很快就会在Maven Central中。

## ProGuard的

如果您使用的是ProGuard, 请将以下规则添加到proguard配置文件中:

```
-keep class com.shockwave.**
```

## 在布局中包含PDFView

```
<com.github.barteksc.pdfviewer.PDFView  
    android:id="@+id/pdfView"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"/>
```

## 加载PDF文件

所有可用选项都带有默认值:

```
pdfView . fromUri (Uri)  
要么  
pdfView . fromFile (文件)  
要么  
pdfView . fromBytes (byte [])  
要么  
pdfView . fromStream (InputStream) //将流写入bytearray - 本机代码不能使用Java Streams  
要么  
pdfView . fromSource (DocumentSource)  
要么  
pdfView . fromAsset (String)  
    .PAGES (0, 2, 1, 3, 3, 3) //所有页面默认显示  
    .enableSwipe (真) //允许阻止改变使用滑动页  
    .swipeHorizontal (假)  
    .enableDoubletap (true)  
    .defaultPage (0)  
    //允许在当前页面上绘制一些东西, 通常在屏幕中间可见  
    .onDraw (onDrawListener)
```

```
// 允许在所有页面上绘制内容，分别为每个页面绘制。仅针对可见页面调用
.onDrawAll (onDrawListener)
.onLoad (onLoadCompleteListener) // 在加载文档并开始渲染后调用
.onPageChange (onPageChangeListener)
.onPageScroll (onPageScrollListener)
.onError (onErrorListener)
.onPageError (onPageErrorListener)
.onRender (onRenderListener) // 在第一次呈现文档后调用
// 单击时调用，如果处理则返回true，false以切换滚动句柄可见性
.onTap (onTapListener)
.onLongPress (onLongPressListener)
.enableAnnotationRendering (false) // 呈现注释（例如注释，颜色或表单）
.password (null)
.scrollHandle (null)
.enableAntialiasing (true) // 在低分辨率屏幕上改进渲染一点
// 在dp中页面之间的间距。要定义间距颜色，请设置视图背景
.spacing (0)
.autoSpacing (false) // 在屏幕上添加动态间距以适应每个页面
.linkHandler (DefaultLinkHandler)
.pageFitPolicy (FitPolicy . WIDTH)
.pageSnap (true) // 将页面捕捉到屏幕边界
.pageFling (false) // 仅对像ViewPager
这样的单页进行一次更改 .nightMode (false) // 切换夜间模式
.load () ;
```

- `pages` 是可选的，它允许您根据需要过滤和排序PDF页面

## 滚动手柄

Scroll handle是从1.x分支替换`ScrollBar`。

从版本2.1.0将`PDFView`放在`RelativeLayout`中以使用`ScrollHandle`不是必需的，您可以使用任何布局。

要使用滚动手柄，只需使用方法注册它 `Configurator#scrollHandle()`。此方法接受`ScrollHandle`接口的实现。

`AndroidPdfViewer` 附带默认实现，您可以使用它 `.scrollHandle(new DefaultScrollHandle(this))`。`DefaultScrollHandle`位于右侧（垂直滚动时）或底部（水平滚动时）。通过使用带有第二个参数（`new DefaultScrollHandle(this, true)`）的构造函数，可以将句柄放在左侧或顶部。

您还可以创建自定义滚动句柄，只需实现`ScrollHandle`界面。所有方法都记录为接口源上的Javadoc注释。

## 文件来源

2.3.0版引入了文档源，它们只是PDF文档的提供者。每个提供程序都实现**DocumentSource**接口。预定义的提供程序可以在**com.github.barteksc.pdfviewer.source**包中找到，可以用作创建自定义提供程序的示例。

预定义的提供程序可以与速记方法一起使用：

```
pdfView.fromUri(Uri)
pdfView.fromFile(File)
pdfView.fromBytes(byte[])
pdfView.fromStream(InputStream)
pdfView.fromAsset(String)
```

自定义提供程序可与 `pdfView.fromSource(DocumentSource)` 方法一起使用。

## 链接

3.0.0版引入了对PDF文档中链接的支持。默认情况下，使用**DefaultLinkHandler** 并单击引用同一文档中的页面的链接会导致跳转到目标页面并单击以某个URI为目标的链接导致在默认应用程序中打开它。

您还可以创建自定义链接处理程序，只需实现 **LinkHandler** 接口并使用 `Configurator#linkHandler(LinkHandler)` 方法进行设置。查看[DefaultLinkHandler](#) 源以实现自定义行为。

## 页面符合政策

从版本3.0.0开始，库支持以3种模式将页面装入屏幕：

- 宽度 – 最宽页面的宽度等于屏幕宽度
- 高度 – 最高页面的高度等于屏幕高度
- BOTH – 基于最宽和最高的页面，每个页面都缩放为在屏幕上完全可见

除了选定的策略之外，每个页面都会缩放到相对于其他页面的大小。

可以使用适合的策略进行设置 `Configurator#pageFitPolicy(FitPolicy)`。默认策略是**WIDTH**。

## 其他选项

### 位图质量

默认情况下，生成的位图使用格式**压缩 RGB\_565** 以减少内存消耗。**ARGB\_8888** 可以使用 `pdfView.useBestQuality(true)` 方法强制渲染。

## 双击缩放

有三种缩放级别：分钟（默认1），中间（默认1.75）和最大（默认3）。在第一次双击时，视图缩放到中等水平，在第二个到最大水平，第三个返回到最低水平。如果您处于中级和最高级别之间，则双击会导致缩放到最大值，依此类推。

可以使用以下方法更改缩放级别：

```
void setMinZoom (float zoom) ;  
void setMidZoom (float zoom) ;  
void setMaxZoom (float zoom) ;
```

## 可能的问题

### 为什么导致apk太大了？

Android PdfViewer依赖于PdfumAndroid，它是许多架构的本地库集（大约16 MB）。Apk必须包含所有这些库，以便在市场上的每个设备上运行。幸运的是，Google Play允许我们上传多个apks，例如每个架构一个。有一篇关于自动将您的应用程序拆分为多个apks的文章，[可在此处获得](#)。最重要的部分是使用*APK Splits*改进多个*APK*创建和版本代码处理，但整篇文章值得一读。您只需要在您的应用程序中执行此操作，无需分支PdfumAndroid等。

### 为什么我无法从URL打开PDF？

下载文件是一个长时间运行的过程，必须知道Activity生命周期，必须支持一些配置，数据清理和缓存，因此创建这样的模块可能最终会成为新的库。

### 如何在配置更改后显示上次打开的页面？

您必须存储当前页码然后进行设置 `pdfView.defaultPage(page)`，请参阅示例应用程序

### 如何将文档放入屏幕宽度（例如，方向更改）？

`FitPolicy.WIDTH` 如果要在具有不同页面大小的文档中放置所需页面，请使用策略或添加以下代码段：

```
配置器。onRender (new OnRenderListener () {  
    @  
    Override public void onInitiallyRendered (int pages, float pageWidth, float pageHeight) {  
        pdfView . fitToWidth (PageIndex的) ;  
    }  
}) ;
```

## 如何像ViewPager一样滚动浏览单个页面？

您可以使用以下设置的组合来获得类似于ViewPager的滚动和拖动行为：

```
.swipeHorizontal (true)
.pageSnap (true)
.autoSpacing (true)
.pageFling (true)
```

Demo地址：<https://github.com/xiaofeng0325/AndroidPDFView>

参考：<https://github.com/barteksc/AndroidPdfViewer>

坚持原创技术分享，您的支持将鼓励我继续创作！

打赏

[# blog](#) [# markdown](#) [# Android](#)

◀ android日志框架Logger的详细使用（目前最新版2.1.1）

AnimationDrawable使用简介 ▶

© 2018  ZhangMiao

由 [Hexo](#) 强力驱动 | 主题 — [NexT.Pisces](#) v5.1.4