

1. 简介

联想图像提供了PdfRender库，用于将Pdf文档转换为图片格式。本文档介绍其基本使用方法。

2. 接口说明

2.1 初始化

用于创建PdfRender实例。

```
class PDFRenderer(pdfFilePath: String, imageFolderPath:String? = null)
```

参数说明

- pdfFilePath，带转换的pdf文件路径。
- imageFolderPath，转化结果存储目录。如不指定则默认使用程序外部cache目录，详见context.externalCacheDir.absolutePath。

2.2 加密判断

若指定的文档已加密则返回true。

```
fun isEncrypted(): Boolean
```

2.3 解密文档

使用指定密码尝试解密文档，若成功则返回true。

```
fun unlockWithPassword(password: String): Boolean
```

2.4 文档页数

返回指定文档的页数。

```
fun numberOfPages(): Int
```

2.5 提取文档

提取全部pdf文档为指定ImageType, 并输出至指定的imageFolderPath。

```
fun allPages(  
    inType: ImageType,  
    withDPI: Float,  
    context: Context,  
    pageComplete: PageCompleteCallback,  
    jobComplete: JobCompleteCallback  
)
```

参数说明

- `inType`, 输出图片类别, 可为ImageType.BMP或ImageType.PNG
- `withDPI`, 像素密度, 默认为600。
- `context` , 上下文。
- `pageComplete`, 页转化完成回调。
- `jobComplete` , 转化任务完成回调。

回调接口定义如下:

```
interface PageCompleteCallback {  
    fun pngPath(path: String?, forPage: Int): Void  
}  
  
interface JobCompleteCallback {  
    fun complete(result: Boolean): Void  
}
```

此外, 可使用page方法提取指定pdf页面到图片。

```
fun page(atIndex: Int, inType: ImageType, withDPI: Float, context:  
Context, complete: PageCompleteCallback)
```

可用cancel方法取消进行中的allPages任务。

```
fun cancel()
```

2.6 退出

退出转化并释放持有的资源。

```
fun destory()
```

3 示例代码

1. 创建PDFRenderer实例

```
PDFRenderer pdfRenderer = new PDFRenderer(yourPdfFilePath,  
yourImageOutputFolder)
```

2. 解密文档（可选）

```
if (pdfRenderer.isEncrypted()) {  
    if (!pdfRenderer.unlockWithPassword("your_password")) {  
        return  
    }  
}
```

3. 提取文档

```
pdfRenderer.allPages(  
    PDFRenderer.ImageType.PNG, 600, getContext(),  
    new PageCompleteCallback() {  
        @Override  
        public Void pngPath(String path, int forPage) {  
            Log.d(tag, format("got image file %s for pdf page %d",  
path, forPage));  
            return null;  
        }  
    },  
    new JobCompleteCallback() {  
        @Override  
        public Void complete(boolean result) {  
            Log.d(tag, "job done");  
            return null;  
        }  
    }  
);
```

5. 注意事项

1. 不支持多线程转换多个pdf文档。
2. 本库只是对开源项目 [Pdfium](#) 的简单二次封装，使用者需做好相关版权声明。