可能是国内封装 Libreoffice 最好的一个项目

作者：liumapp
原文链接：https://ld246.com/article/1558083988473
来源网站：链滴

基于libreoffice实现的文档转换项目，无框架依赖，即插即用

项目源代码：github/workable-converter

1. 技术栈

- LibreOffice:v6.2.3
- jodconverter:4.2.2
- PDFBox:2.0.12
- cglib动态代理 + 懒汉工厂模式 + 策略模式 + 装饰器模式
- qtools-property管理配置文件(application.yml、bootstrap.yml、workable-converter.yml三种名的配置文件任意包含一种即可)

2. 功能

- 支持doc、docx、html、ppt、png、pdf等等类型的文件 **互相转换**
- 支持按照文件路径、字节输入输出流、Base64等不同姿势转换
- 不依赖第三方框架，即插即用，支持application.yml、bootstrap.yml、workable-converter.yml种配置（自己项目中具体配置一个即可）

3. 使用

3.1 安装配置LibreOffice6.2.3

CentOS请直接参考这篇文章：CentOS7安装LibreOffice6.2.3

windows跟Mac同样可以在上述文章中拿到下载链接

安装完成后，请记住您的LibreOffice的Home目录，后面需要用到

默认目录：

- CentOS: /opt/libreoffice6.2/
- Mac: /Applications/LibreOffice.app/Contents/
- Windows: C:\Program Files\Libr
Office\

3.2 获取依赖

- Maven

```
<dependency>
  <groupId>com.liumapp.workable.converter</groupId>
  <artifactId>workable-converter</artifactId>
  <version>v1.1.0</version>
</dependency>
```

- Gradle

```
compile group: 'com.liumapp.workable.converter', name: 'workable-converter', version: 'v1.1.0
```

3.3 编辑配置文件

在项目的resources目录下，创建一个yml配置文件，需要确保文件名称为application.yml、bootstra
.yml或workable-converter.yml三种命名任意一个即可

添加以下配置：

```
com:
  liumapp:
    workable-converter:
      libreofficePath: "/Applications/LibreOffice.app/Contents"
```

libreofficePath的值为LibreOffice:6.2.3的安装目录

完整的配置项列表如下

<table>
<tr><th>参数名</th><th>解释</th><th>默认值</th></tr>
<tr><td>libreofficePath</td><td>LibreOffice安装目录</td><td>(String) 无默认值，该项必填
/td></tr>
<tr><td>libreofficePort</td><td>LibreOffice监听端口</td><td>(int) 2002</td></tr>
<tr><td>tmpPath</td><td>临时存储目录</td><td>(String) "./data/"</td></tr>

</table>

3.4 执行转换

3.4.1 按照文件路径转换

以doc转PDF为例

```
WorkableConverter converter = new WorkableConverter();//实例化的同时，初始化配置项，配
项的校验通过Decorator装饰

ConvertPattern pattern = ConvertPatternManager.getInstance();
pattern.fileToFile("./data/test.doc", "./data/pdf/result1.pdf"); //test.doc为待转换文件路径，resul
1.pdf为转换结果存储路径
pattern.setSrcFilePrefix(DefaultDocumentFormatRegistry.DOC);
pattern.setDestFilePrefix(DefaultDocumentFormatRegistry.PDF);

converter.setConverterType(CommonConverterManager.getInstance());//策略模式，后续实现
新的转换策略后，在此处更换，图片转换将考虑使用新的策略来完成
boolean result = converter.convert(pattern.getParameter();
```

如果要用html转PDF，将上述代码的

```
pattern.setSrcFilePrefix(DefaultDocumentFormatRegistry.DOC);
pattern.setDestFilePrefix(DefaultDocumentFormatRegistry.PDF);
```

改为

```
pattern.setSrcFilePrefix(DefaultDocumentFormatRegistry.HTML);
pattern.setDestFilePrefix(DefaultDocumentFormatRegistry.PDF);
```

其他类型的同理

3.4.2 按照输入输出流转换

以doc转pdf为例

```
// you can also choice not use proxy
WorkableConverter converter = new WorkableConverter();
ConvertPattern pattern = ConvertPatternManager.getInstance();
pattern.streamToStream(new FileInputStream("./data/test.doc"), new FileOutputStream("./dat
/pdf/result1_2.pdf"));
// attention !!! convert by stream must set prefix.
pattern.setSrcFilePrefix(DefaultDocumentFormatRegistry.DOC);
pattern.setDestFilePrefix(DefaultDocumentFormatRegistry.PDF);
converter.setConverterType(CommonConverterManager.getInstance());
boolean result = converter.convert(pattern.getParameter();
```

跟上例基本相同，唯一的变化是通过pattern.streamToStream()来设置输入输出流，转换源文件数据输入流中读取，转换结果会直接写入输出流中，

同时要切换转换格式，跟上例一样设置不同的prefix即可

3.4.3 按照文件Base64转换

仍以doc转pdf为例

```
WorkableConverter converter = new WorkableConverter();
ConvertPattern pattern = ConvertPatternManager.getInstance();
pattern.base64ToBase64(Base64FileTool.FileToBase64(new File("./data/test.doc")));
// attention !!! convert by base64 must set prefix.
pattern.setSrcFilePrefix(DefaultDocumentFormatRegistry.DOC);
pattern.setDestFilePrefix(DefaultDocumentFormatRegistry.PDF);
converter.setConverterType(CommonConverterManager.getInstance());
boolean result = converter.convert(pattern.getParameter();
String destBase64 = pattern.getBase64Result();
```

输入base64执行转换，首先通过pattern.base64ToBase64()来设置转换源的base64值

转换结果result仍然是一个boolean类型，通过pattern.getBase64Result来获取转换结果的base64值

要切换转换格式，跟上例一样设置不同的prefix即可

3.5 图片处理

目前对于图片的处理，只支持将PDF转PNG图片(如果1份pdf文件有20页，那么将会转换为20张png片)，该功能的实现基于PDFBox:2.0.12

3.5.1 按照文件路径处理

pattern.fileToFiles()第一个参数为待转换的pdf文件路径，第二个参数为转换后的图片存储路径

```
WorkableConverter converter = new WorkableConverter();
ConvertPattern pattern = ConvertPatternManager.getInstance();
pattern.fileToFiles("./data/test5.pdf", "./data/");
pattern.setSrcFilePrefix(DefaultDocumentFormatRegistry.PDF);
pattern.setDestFilePrefix(DefaultDocumentFormatRegistry.PNG);
converter.setConverterType(PdfBoxConverterManager.getInstance()); // pdf box converter ma
ager only support pdf to png
assertEquals(true, converter.convert(pattern.getParameter()));
assertEquals(true, FileTool.isFileExists("./data/test5_0.png"));
assertEquals(true, FileTool.isFileExists("./data/test5_1.png"));
assertEquals(true, FileTool.isFileExists("./data/test5_2.png"));
assertEquals(true, FileTool.isFileExists("./data/test5_3.png"));
```

3.5.2 按照文件Base64处理

pattern.base64ToBase64()的参数为待转换pdf文件的base64值

转换结束后，通过List<String> resultBase64 = pattern.getBase64Results()获取转换后的图片base4值的集合

```
WorkableConverter converter = new WorkableConverter();
ConvertPattern pattern = ConvertPatternManager.getInstance();
pattern.base64ToBase64(Base64FileTool.FileToBase64(new File("./data/test5.pdf")));
```

```
pattern.setSrcFilePrefix(DefaultDocumentFormatRegistry.PDF);
pattern.setDestFilePrefix(DefaultDocumentFormatRegistry.PNG);
converter.setConverterType(PdfBoxConverterManager.getInstance()); // pdf box converter ma
ager only support pdf to png
boolean result = converter.convert(pattern.getParameter());
List<String> resultBase64 = pattern.getBase64Results();
assertEquals(true, result);
assertEquals(4, resultBase64.size());
```

4. 待办事项

- 已经测试通过的有doc、docx、html 按照不同姿势转PDF，其他类型的并没有编写测试单元，后考虑增加
- 目前只支持yml配置，后续考虑添加其他类型的配置支持（xml、properties等）
- 目前Markdown格式很流行，考虑实现markdown格式的字符串转PDF(markdown -> html -> pdf)

5. 注意事项

- 因为需要LibreOffice的支持，所以不建议在Docker等容器内运行(LibreOffice暂无Docker稳定发版的镜像)
- 转换乱码、转换耗时过长，请检查服务器是否安装有中文字体
- 项目启动后，在执行第一次转换任务时，因为涉及到与LibreOffice建立连接等操作，所以会耗时较，第二次任务及以后稳定在0.5秒以内（具体时间因机器配置会有所差异）

6. 参考链接

- https://www.libreoffice.org/download/download/?type=rpm-x86_64&version=6.2.3&lang=h-CN
- https://api.libreoffice.org/
- https://github.com/sbraconnier/jodconverter/