

基于 libreoffice 实现的文档转换项目，无框架依赖，即插即用

- [1. 技术栈](#)
- [2. 功能](#)
- [3. 使用](#)
 - [3.1 安装配置 LibreOffice6.2.3](#)
 - [3.2 获取依赖](#)
 - [3.3 编辑配置文件](#)
 - [3.4 执行转换](#)
 - [3.4.1 按照文件路径转换](#)
 - [3.4.2 按照输入输出流转换](#)
 - [3.4.3 按照文件 Base64 转换](#)
- [4. 待办事项](#)
- [5. 参考链接](#)

1. 技术栈

- LibreOffice:v6.2.3
- jodconverter:4.2.2
- 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 三种命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 三种配置（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\LibreOffice\

3.2 获取依赖

- Maven

```
<dependency>
  <groupId>com.liumapp.workable.converter</groupId>
  <artifactId>workable-converter</artifactId>
  <version>v1.0.0</version>
</dependency>
```

、

- Gradle

```
compile group: 'com.liumapp.workable.converter', name: 'workable-converter', version:
'v1.0.0'
```

、

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、bootstrap.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

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

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，配置项的校验通过 Decorator 装饰

```
ConvertPattern pattern = ConvertPatternManager.getInstance();
pattern.setConvertByFilePathRequire("./data/test.doc", "./data/pdf/result1.pdf");
//test.doc 为待转换文件路径，result1.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，后续实现了新的转换策略后，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.setConvertByStream(new FileInputStream("./data/test.doc"), new
FileOutputStream("./data/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.setConvertByStream() 来设置输入输出流，转换源文件数据从输入流中读取，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.setConvertByBase64(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.setConvertByBase64()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

4. 待办事项

- 文档跟图片之间的转换还不够智能，比如：一份 20 多页的 doc 文档转图片，只支持单页转换，不能批量按页转换，后续迭代改进
- 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，后续考虑增加
- 目前只支持 yaml 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. 参考链接

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