

# OpenOffice学习笔记

---

## 概述

- aspose-words
- poi
- OpenOffice
- spire.doc
- docx4j
- 其它方案
  - lText
  - document4j

## docx4j实现

- 使用
- 结论

## poi实现

- 使用
- 结论

## openoffice

### 概述

### 安装与部署

#### Windows安装

- 第一步：下载
- 第二步：安装
- 第三步：启动GUI程序
- 第四步：启动服务
- 第五步：检查是否启动成功
- 第六步：安装中文语言包

#### Linux安装

- 第一步：下载
- 第二步：创建文件夹
- 第三步：上传文件
- 第四步：解压文件
- 第五步：安装rpm文件
- 第六步：启动

#### Docker安装

- 第一步：搜索镜像
- 第二步：拉取镜像
- 第三步：查看镜像是否拉取成功
- 第四步：运行镜像
- 第五步：查看容器是否运行

## SpringBoot整合openoffice

### 概述

### docx转pdf

- 第一步：创建Springboot程序
- 第二步：配置maven
- 第三步：编写application.yml配置
- 第四步：启动服务
- 第五步：编写单元测试
- 第六步：编写文档转换服务
- 第七步：编写服务实现类
- 第八步：编写接口

第九步：编写简单前端页面

第十步：启动项目测试

## java调用openoffice

概述

docx转pdf

第一步：新建maven项目

第二步：配置maven

第三步：编写DocumentConverterUtil

第四步：编写openoffice.properties

第五步：编写测试类

第六步：运行查看结果

---

# 概述

公司有一个需求，需要处理word文件，再将word文件转换成pdf文件响应给用户，服务器是Linux服务器，代码是跑在Linux上的，有些转pdf的技术只能跑在Windows平台上。

关于Word转PDF网上能找到的方案大概有六七种，其中5种如下：

- aspose-words
- docx4j
- openoffice
- poi
- spire.doc

---

## aspose-words

Aspose公司旗下的最全的一套office文档管理方案，公司设在澳大利亚。

公司差不多是专做各种文件格式处理插件的，产品系列挺多

官网：<https://www.aspose.com/>

不需要依赖任何组件，不依赖操作系统。

收费，不考虑

```
1 <dependency>
2   <groupId>com.aspose</groupId>
3   <artifactId>aspose-words</artifactId>
4   <version>22.11</version>
5   <classifier>jdk17</classifier>
6 </dependency>
```

## poi

---

大名鼎鼎的apache的开源组件，应用非常广泛

官网：<https://poi.apache.org/>

组件拆分较细，引用一些类库，但都问题不大，不依赖操作系统，经常被用于excel文档处理

## OpenOffice

---

Apache旗下又一开源组件，前身是1998年一家德国公司StarDivision所研发出来的一个办公室软件，称之为StarOffice。1999年8月被sun公司收购。2010年团队成员分家，分出来的一批成立了新团队做一个LibreOffice。2011年6月Oracle将其捐赠给Apache基金会

OpenOffice本身就是一套Office软件，该方案需要使用jodconverter组件配合OpenOffice完成转换，推荐使用此方案

jodconverter：<https://sourceforge.net/projects/jodconverter/files/>

JODConverter，是一个Java的OpenDocument文件转换器，可以进行许多文件格式的转换。它依赖于OpenOffice.org或者LibreOffice提供的服务来进行转换，它能将Microsoft Office文档（Word，Excel，PowerPoint）转换为PDF格式。

## spire.doc

---

收费，用公司名和邮箱可以申请一个月的试用license，不考虑

# docx4j

---

澳大利亚一公司赞助的开源组件

有一个开源版，还有一个Docx4j Enterprise Edition，开源版转换效果不行，表格严重错位，与原版格式严重失真

不依赖其它组件，不依赖操作系统

官网：<https://www.docx4java.org/>

## 其它方案

---

### IText

IText是直接操作PDF的，不是docx转pdf，而且java API非常难用

```
1 <dependency>
2   <groupId>com.lowagie</groupId>
3   <artifactId>itext</artifactId>
4   <version>4.2.2</version>
5 </dependency>
```

### document4j

document4j是依赖于office软件，Linux无法使用

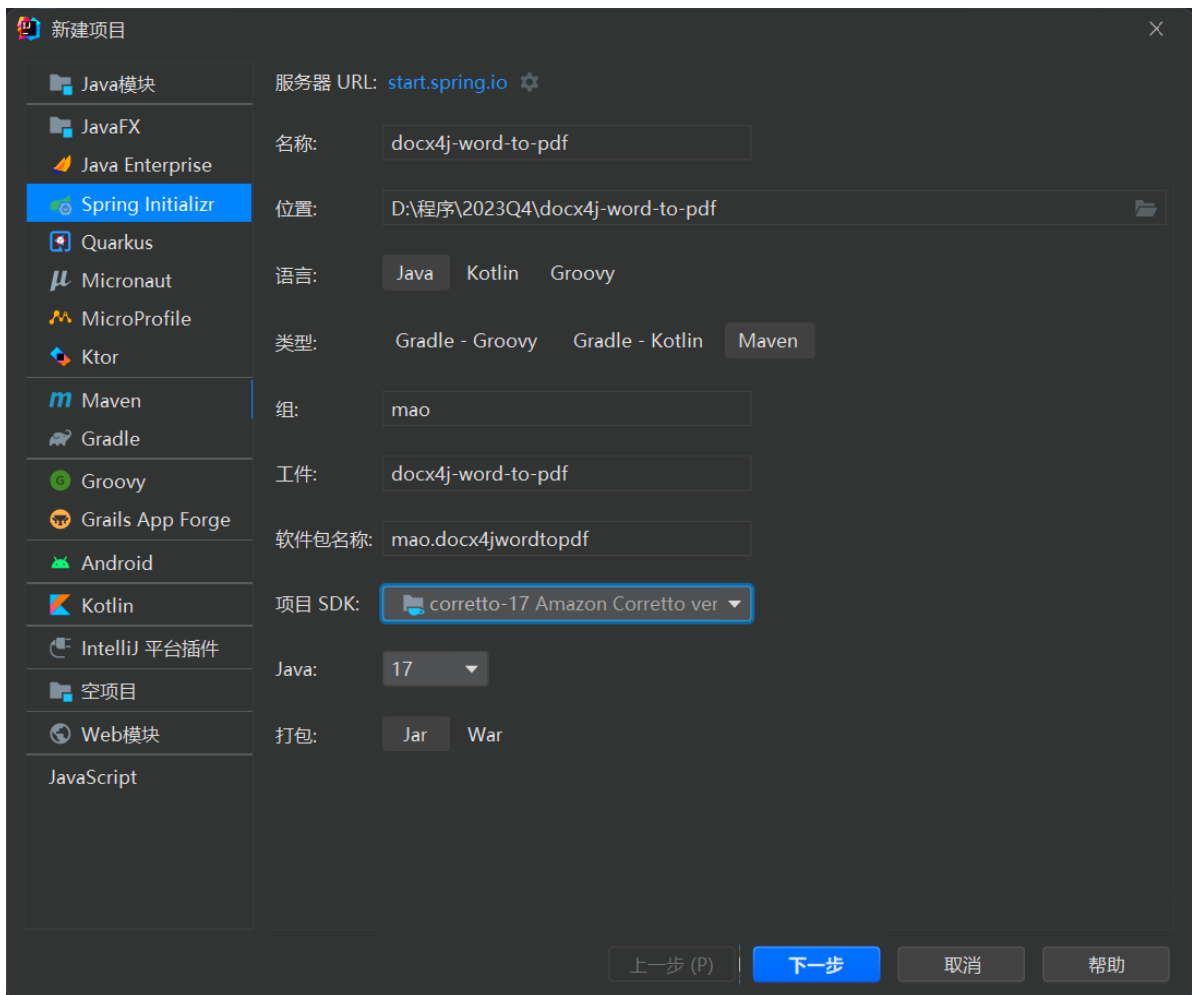
## docx4j实现

---

### 使用

---

创建项目：



pom文件依赖如下:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
5     https://maven.apache.org/xsd/maven-4.0.0.xsd">
6     <modelVersion>4.0.0</modelVersion>
7     <parent>
8         <groupId>org.springframework.boot</groupId>
9         <artifactId>spring-boot-starter-parent</artifactId>
10        <version>2.7.1</version>
11        <relativePath/> <!-- lookup parent from repository -->
12    </parent>
13    <groupId>mao</groupId>
14    <artifactId>docx4j-word-to-pdf</artifactId>
15    <version>0.0.1-SNAPSHOT</version>
16    <name>docx4j-word-to-pdf</name>
17    <description>docx4j-word-to-pdf</description>
18
19    <properties>
20        <java.version>17</java.version>
21    </properties>
22
23    <dependencies>
```

```

22     <dependency>
23         <groupId>org.springframework.boot</groupId>
24         <artifactId>spring-boot-starter</artifactId>
25     </dependency>
26
27     <dependency>
28         <groupId>org.springframework.boot</groupId>
29         <artifactId>spring-boot-starter-test</artifactId>
30         <scope>test</scope>
31     </dependency>
32
33     <dependency>
34         <groupId>org.docx4j</groupId>
35         <artifactId>docx4j-JAXB-Internal</artifactId>
36         <version>8.3.1</version>
37     </dependency>
38     <dependency>
39         <groupId>org.docx4j</groupId>
40         <artifactId>docx4j-JAXB-ReferenceImpl</artifactId>
41         <version>8.3.1</version>
42     </dependency>
43     <dependency>
44         <groupId>org.docx4j</groupId>
45         <artifactId>docx4j-export-fo</artifactId>
46         <version>8.3.1</version>
47     </dependency>
48
49 </dependencies>
50
51 <build>
52     <plugins>
53         <plugin>
54             <groupId>org.springframework.boot</groupId>
55             <artifactId>spring-boot-maven-plugin</artifactId>
56         </plugin>
57     </plugins>
58 </build>
59
60 </project>

```

编写工具类:

```

1  package mao.docx4jwordtopdf.utils;
2
3  import org.apache.commons.compress.utils.IOUtils;
4  import org.docx4j.Docx4J;
5  import org.docx4j.fonts.IdentityPlusMapper;
6  import org.docx4j.fonts.Mapper;
7  import org.docx4j.fonts.PhysicalFont;
8  import org.docx4j.fonts.PhysicalFonts;
9  import org.docx4j.openpackaging.packages.WordprocessingMLPackage;
10 import org.slf4j.Logger;

```

```

11 import org.slf4j.LoggerFactory;
12
13 import javax.servlet.http.HttpServletResponse;
14 import java.io.File;
15 import java.io.FileOutputStream;
16 import java.util.UUID;
17
18 /**
19  * Project name(项目名称): docx4j-word-to-pdf
20  * Package(包名): mao.docx4jwordtopdf.utils
21  * Class(类名): DocxToPdfUtils
22  * Author(作者): mao
23  * Author QQ: 1296193245
24  * GitHub: https://github.com/maomao124/
25  * Date(创建日期): 2023/11/25
26  * Time(创建时间): 18:06
27  * Version(版本): 1.0
28  * Description(描述): docx转pdf
29  */
30
31 public class DocxToPdfUtils
32 {
33     private static final Logger log =
34     LoggerFactory.getLogger(DocxToPdfUtils.class);
35
36     /**
37      * docx转pdf
38      *
39      * @param docxPath docx文件路径
40      * @param pdfPath pdf文件路径
41      * @throws Exception 异常
42      */
43     public static void convertDocxToPdf(String docxPath, String pdfPath)
44     throws Exception
45     {
46         FileOutputStream fileOutputStream = null;
47         try
48         {
49             File file = new File(docxPath);
50             fileOutputStream = new FileOutputStream(new File(pdfPath));
51             wordprocessingMLPackage mlpPackage =
52             WordprocessingMLPackage.load(file);
53             setFontMapper(mlpPackage);
54             Docx4J.toPDF(mlpPackage, new FileOutputStream(new
55             File(pdfPath)));
56         }
57         catch (Exception e)
58         {
59             e.printStackTrace();
60             log.error("docx文档转换为PDF失败");
61         }
62     }
63 }

```

```

62         IOUtils.closeQuietly(fileOutputStream);
63     }
64 }
65
66 /**
67  * 加载字体文件（解决linux环境下无中文字体问题）
68  *
69  * @param mlPackage {@link WordprocessingMLPackage}
70  * @throws Exception 异常
71  */
72 private static void setFontMapper(WordprocessingMLPackage mlPackage)
73 throws Exception
74 {
75     Mapper fontMapper = new IdentityPlusMapper();
76     //加载字体文件（解决linux环境下无中文字体问题）
77     if (PhysicalFonts.get("SimSun") == null)
78     {
79         System.out.println("加载本地SimSun字体库");
80         //PhysicalFonts.addPhysicalFonts("SimSun",
81         WordUtils.class.getResource("/fonts/SIMSUN.TTC"));
82     }
83     fontMapper.put("隶书", PhysicalFonts.get("LiSu"));
84     fontMapper.put("宋体", PhysicalFonts.get("SimSun"));
85     fontMapper.put("微软雅黑", PhysicalFonts.get("Microsoft Yahei"));
86     fontMapper.put("黑体", PhysicalFonts.get("SimHei"));
87     fontMapper.put("楷体", PhysicalFonts.get("KaiTi"));
88     fontMapper.put("新宋体", PhysicalFonts.get("NSimSun"));
89     fontMapper.put("华文行楷", PhysicalFonts.get("STXingkai"));
90     fontMapper.put("华文仿宋", PhysicalFonts.get("STFangsong"));
91     fontMapper.put("仿宋", PhysicalFonts.get("FangSong"));
92     fontMapper.put("幼圆", PhysicalFonts.get("YouYuan"));
93     fontMapper.put("华文宋体", PhysicalFonts.get("STSong"));
94     fontMapper.put("华文中宋", PhysicalFonts.get("STZhongsong"));
95     fontMapper.put("等线", PhysicalFonts.get("SimSun"));
96     fontMapper.put("等线 Light", PhysicalFonts.get("SimSun"));
97     fontMapper.put("华文琥珀", PhysicalFonts.get("STHupo"));
98     fontMapper.put("华文隶书", PhysicalFonts.get("STLiti"));
99     fontMapper.put("华文新魏", PhysicalFonts.get("STXinwei"));
100    fontMapper.put("华文彩云", PhysicalFonts.get("STCaiyun"));
101    fontMapper.put("方正姚体", PhysicalFonts.get("FZYaoti"));
102    fontMapper.put("方正舒体", PhysicalFonts.get("FZShuTi"));
103    fontMapper.put("华文细黑", PhysicalFonts.get("STXihei"));
104    fontMapper.put("宋体扩展", PhysicalFonts.get("simsun-extB"));
105    fontMapper.put("仿宋_GB2312",
106    PhysicalFonts.get("FangSong_GB2312"));
107    fontMapper.put("新細明體", PhysicalFonts.get("SimSun"));
108    //解决宋体（正文）和宋体（标题）的乱码问题
109    PhysicalFonts.put("PMingLiU", PhysicalFonts.get("SimSun"));
110    PhysicalFonts.put("新細明體", PhysicalFonts.get("SimSun"));
111    //宋体&新宋体
112    PhysicalFont simsunFont = PhysicalFonts.get("SimSun");
113    fontMapper.put("SimSun", simsunFont);
114    //设置字体
115    mlPackage.setFontMapper(fontMapper);
116 }

```



编写单元测试：

```

1  package mao.docx4jwordtopdf.utils;
2
3  import org.junit.jupiter.api.Test;
4
5  import static org.junit.jupiter.api.Assertions.*;
6
7  /**
8   * Project name(项目名称): docx4j-word-to-pdf
9   * Package(包名): mao.docx4jwordtopdf.utils
10  * Class(测试类名): DocxToPdfUtilsTest
11  * Author(作者): mao
12  * Author QQ: 1296193245
13  * GitHub: https://github.com/maomao124/
14  * Date(创建日期): 2023/11/25
15  * Time(创建时间): 18:14
16  * Version(版本): 1.0
17  * Description(描述): 测试类
18  */
19
20  class DocxToPdfUtilsTest
21  {
22
23      @Test
24      void convertDocxToPdf() throws Exception
25      {
26          DocxToPdfUtils.convertDocxToPdf("./test.docx", "./test.pdf");
27      }
28
29      @Test
30      void convertDocxToPdf2() throws Exception
31      {
32          //相对复杂的docx
33          DocxToPdfUtils.convertDocxToPdf("./out.docx", "./out.pdf");
34      }
35  }

```

test.docx内容如下：

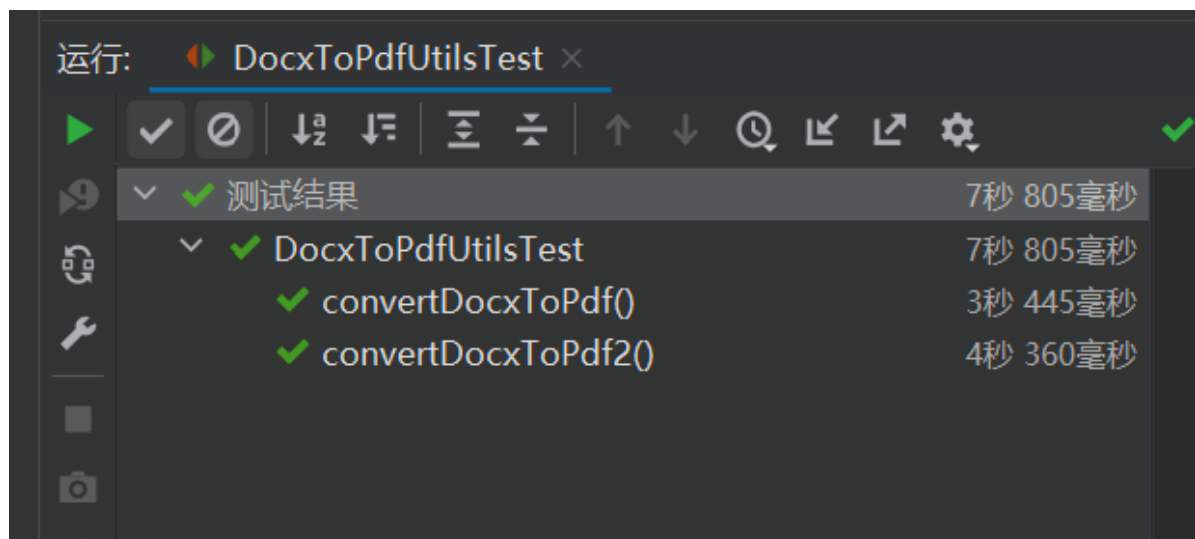


12←

131314←

1

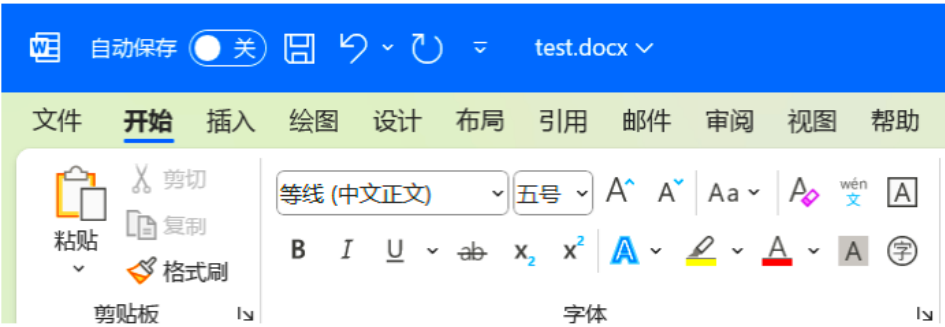
10 / 71



test.pdf输出结果如下:

Wrewrwet

Sfsfsgsdg



Werwewe

中文

居中  
12  
5644  
131314

1	2	3
4	5	6
7	8	9

## 结论

test.pdf目前没什么大问题，字体和格式有点问题，但是out.pdf就乱了，因为out.docx是公司的合同文档，这里就不放出结果了，而且输出太慢了，CPU占用较高，会拖垮服务器的，不考虑

# poi实现

## 使用

创建springboot程序，pom文件如下：

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
   http://maven.apache.org/xsd/maven-4.0.0.xsd">
4      <modelVersion>4.0.0</modelVersion>
5      <parent>
6          <groupId>org.springframework.boot</groupId>
7          <artifactId>spring-boot-starter-parent</artifactId>
8          <version>2.7.1</version>
9          <relativePath/> <!-- lookup parent from repository -->
10     </parent>
11     <groupId>mao</groupId>
12     <artifactId>poi-word-to-pdf</artifactId>
13     <version>0.0.1-SNAPSHOT</version>
14     <name>poi-word-to-pdf</name>
15     <description>poi-word-to-pdf</description>
16     <properties>
17         <java.version>17</java.version>
18     </properties>
19     <dependencies>
20
21         <dependency>
22             <groupId>org.springframework.boot</groupId>
23             <artifactId>spring-boot-starter</artifactId>
24         </dependency>
25
26         <dependency>
27             <groupId>org.springframework.boot</groupId>
28             <artifactId>spring-boot-starter-test</artifactId>
29             <scope>test</scope>
30         </dependency>
31
32         <dependency>
33             <groupId>org.apache.poi</groupId>
34             <artifactId>poi-ooxml</artifactId>
35             <version>4.1.2</version>
36         </dependency>
37         <!-- word转PDF操作依赖 -->
38         <dependency>
39             <groupId>fr.opensagres.xdocreport</groupId>
```

```

40         <artifactId>fr.opensagres.poi.xwpf.converter.pdf-
gae</artifactId>
41         <version>2.0.2</version>
42     </dependency>
43
44 </dependencies>
45
46 <build>
47     <plugins>
48         <plugin>
49             <groupId>org.springframework.boot</groupId>
50             <artifactId>spring-boot-maven-plugin</artifactId>
51         </plugin>
52     </plugins>
53 </build>
54
55 </project>
56

```

编写转换工具类DocxToPdfUtils:

```

1  package mao.poiwordtopdf.utils;
2
3
4  import fr.opensagres.poi.xwpf.converter.pdf.PdfConverter;
5  import fr.opensagres.poi.xwpf.converter.pdf.PdfOptions;
6  import org.apache.poi.xwpf.usermodel.XWPFDocument;
7  import java.io.*;
8
9
10 import java.io.*;
11
12 /**
13  * Project name(项目名称): poi-word-to-pdf
14  * Package(包名): mao.poiwordtopdf.utils
15  * Class(类名): DocxToPdfUtils
16  * Author(作者): mao
17  * Author QQ: 1296193245
18  * Github: https://github.com/maomao124/
19  * Date(创建日期): 2023/11/25
20  * Time(创建时间): 19:26
21  * Version(版本): 1.0
22  * Description(描述): 无
23  */
24
25 public class DocxToPdfUtils
26 {
27     /**
28      * word转pdf
29      *
30      * @param wordPath word文件路径
31      * @param pdfPath pdf文件路径

```

```

32     * @throws IOException ioexception
33     */
34     public static void wordToPDF(String wordPath, String pdfPath) throws
IOException
35     {
36         XWPFDocument xwpfDocument = null;
37         InputStream inputStream = null;
38         OutputStream outputStream = null;
39         try
40         {
41             inputStream = new FileInputStream(wordPath);
42             xwpfDocument = new XWPFDocument(inputStream);
43             PdfOptions pdfOptions = PdfOptions.create();
44             outputStream = new FileOutputStream(pdfPath);
45             PdfConverter.getInstance().convert(xwpfDocument, outputStream,
pdfOptions);
46         }
47         finally
48         {
49             if (xwpfDocument != null)
50             {
51                 xwpfDocument.close();
52             }
53             if (inputStream != null)
54             {
55                 inputStream.close();
56             }
57             if (outputStream != null)
58             {
59                 outputStream.close();
60             }
61         }
62     }
63 }
64

```

编写测试类：

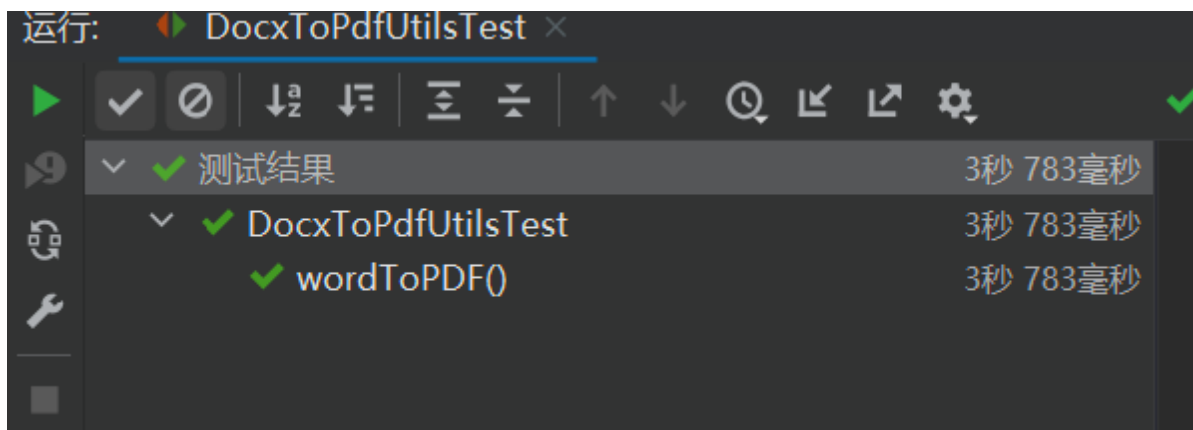
```

1  package mao.poiwordtopdf.utils;
2
3  import org.junit.jupiter.api.Test;
4  import org.springframework.boot.test.context.SpringBootTest;
5
6  import java.io.IOException;
7
8  import static org.junit.jupiter.api.Assertions.*;
9
10 /**
11  * Project name(项目名称): poi-word-to-pdf
12  * Package(包名): mao.poiwordtopdf.utils
13  * Class(测试类名): DocxToPdfUtilsTest
14  * Author(作者): mao

```

```
15  * Author QQ: 1296193245
16  * GitHub: https://github.com/maomao124/
17  * Date(创建日期): 2023/11/25
18  * Time(创建时间): 19:36
19  * Version(版本): 1.0
20  * Description(描述): 测试类
21  */
22
23 @SpringBootTest
24 class DocxToPdfUtilsTest
25 {
26
27     @Test
28     void wordToPDF() throws IOException
29     {
30         DocxToPdfUtils.wordToPDF("./test.docx", "test.pdf");
31     }
32 }
```

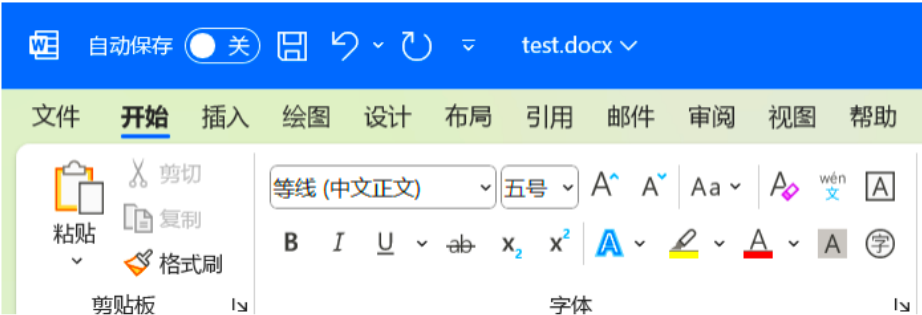
还是那个docx文件，测试结果如下：





Wrewrwet

Sfsfsgsdg



Werwewe

中文

居中

12  
5644  
131314

1	2	3
4	5	6
7	8	9

## 结论

效果不好，性能不好，不考虑

# openoffice

## 概述

OpenOffice.org 是一套跨平台的办公室软件套件，能在 Windows、Linux、MacOS X (X11)、和 Solaris 等操作系统上执行。它与各个主要的办公室软件套件兼容。OpenOffice.org 是开源软件，任何人都可以免费下载、使用、及推广它。

OpenOffice.org 的主要模块有 Writer (文本文档)/Calc (电子表格)/Impress (演示文稿)/Math (公式计算)/Draw (画图)/Base (数据库)

OpenOffice.org 不仅是六大组件的组合，而且与同类产品不同的是，本套件不是独立软件模块形式创建的，从一开始，它就被设计成一个完整的办公软件包

















推荐使用jodconverter调用OpenOffice来实现docx转PDF，OpenOffice不仅支持word，还支持html、ppt、excel等转PDF

## 安装与部署


### Windows安装

#### 第一步：下载


进入下载界面：<https://sourceforge.net/projects/openofficeorg.mirror/files/>

Summary	Source Code	Files	Blog	Reviews
<div> <b>Download Latest Version</b> Apache_OpenOffice_4.1.14_Win_x86_install_en-US.exe (142.6 MB)</div> <div><a href="#">Get Updates</a></div> <div></div>				
Home				
Name	Modified	Size	Downloads / Week	
 <a href="#">4.1.12</a>	2023-02-23		1,488	
 <a href="#">4.1.13</a>	2023-02-23		2,116	
 <a href="#">4.1.14</a>	2023-02-23		533,186	
 <a href="#">4.1.11</a>	2021-10-05		1,837	
 <a href="#">4.1.10</a>	2021-04-29		19,237	
 <a href="#">4.1.9</a>	2021-01-26		1,237	
 <a href="#">4.1.8</a>	2020-11-05		172	



选择一个版本

Download Latest Version  
Apache\_OpenOffice\_4.1.14\_Win\_x86\_install\_en-US.exe (142.6 MB)


[Get Updates](#)



Home / 4.1.14

Name	Modified	Size	Downloads / Week
<a href="#">Parent folder</a>			
<a href="#">binaries</a>	2023-02-23		533,120 
<a href="#">source</a>	2023-02-23		7 <input type="checkbox"/>
<a href="#">README.md</a>	2023-02-23	143 Bytes	59 <input type="checkbox"/> 
Totals: 3 Items		143 Bytes	533,186

选择二进制包

<a href="#">Parent folder</a>	2023-02-23	533,120 
<a href="#">ru</a>	2023-02-23	12,309 <input type="checkbox"/>
<a href="#">SDK</a>	2023-02-23	7 <input type="checkbox"/>
<a href="#">sk</a>	2023-02-23	940 <input type="checkbox"/>
<a href="#">sl</a>	2023-02-23	248 <input type="checkbox"/>
<a href="#">sr</a>	2023-02-23	321 <input type="checkbox"/>
<a href="#">sv</a>	2023-02-23	1,946 <input type="checkbox"/>
<a href="#">ta</a>	2023-02-23	33 <input type="checkbox"/>
<a href="#">th</a>	2023-02-23	175 <input type="checkbox"/>
<a href="#">tr</a>	2023-02-23	1,971 <input type="checkbox"/>
<a href="#">vi</a>	2023-02-23	178 <input type="checkbox"/>
<a href="#">zh-CN</a>	2023-02-23	2,166 <input type="checkbox"/>
<a href="#">zh-TW</a>	2023-02-23	3,074 <input type="checkbox"/>
Totals: 42 Items		533,120
For more information please have a look at our <a href="#">Release Notes</a> .		
Source: README.md, updated 2023-02-23		

选择中文简体(zh-cn)


Home / 4.1.14 / Binaries / zh-CN

Name	Modified	Size	Downloads / Week	
Parent folder				
<a href="#">Apache_OpenOffice_4.1.14_Win_x86_install_zh-CN.exe</a>	2023-02-23	131.1 MB	1,617	
<a href="#">Apache_OpenOffice_4.1.14_Win_x86_langpack_zh-CN.exe</a>	2023-02-23	13.3 MB	99	
<a href="#">Apache_OpenOffice_4.1.14_MacOS_x86-64_install_zh-CN.dmg</a>	2023-02-13	165.4 MB	85	
<a href="#">Apache_OpenOffice_4.1.14_MacOS_x86-64_langpack_zh-CN.dmg</a>	2023-02-13	13.1 MB	12	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86_install-rpm_zh-CN.tar.gz</a>	2023-02-13	148.5 MB	12	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86_langpack-deb_zh-CN.tar.gz</a>	2023-02-13	13.9 MB	2	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86_langpack-rpm_zh-CN.tar.gz</a>	2023-02-13	13.8 MB	3	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86_install-deb_zh-CN.tar.gz</a>	2023-02-13	146.8 MB	4	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86-64_langpack-deb_zh-CN.tar.gz</a>	2023-02-13	13.9 MB	13	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86-64_langpack-rpm_zh-CN.tar.gz</a>	2023-02-13	13.8 MB	20	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86-64_install-rpm_zh-CN.tar.gz</a>	2023-02-13	156.6 MB	208	
<a href="#">Apache_OpenOffice_4.1.14_Linux_x86-64_install-deb_zh-CN.tar.gz</a>	2023-02-13	155.0 MB	91	
Totals: 12 Items		985.1 MB	2,166	

下载第一个和第二个

- [Apache OpenOffice 4.1.14 Win x86 install zh-CN.exe](#)
- [Apache OpenOffice 4.1.14 Win x86 langpack zh-CN.exe](#)

Home / Browse Open Source / Text Editors / Word Processors / Apache OpenOffice



# Apache OpenOffice

The free and Open Source productivity suite

Your download will start shortly... 1

[Get Updates](#)
[Share This](#)
[Problems Downloading?](#)

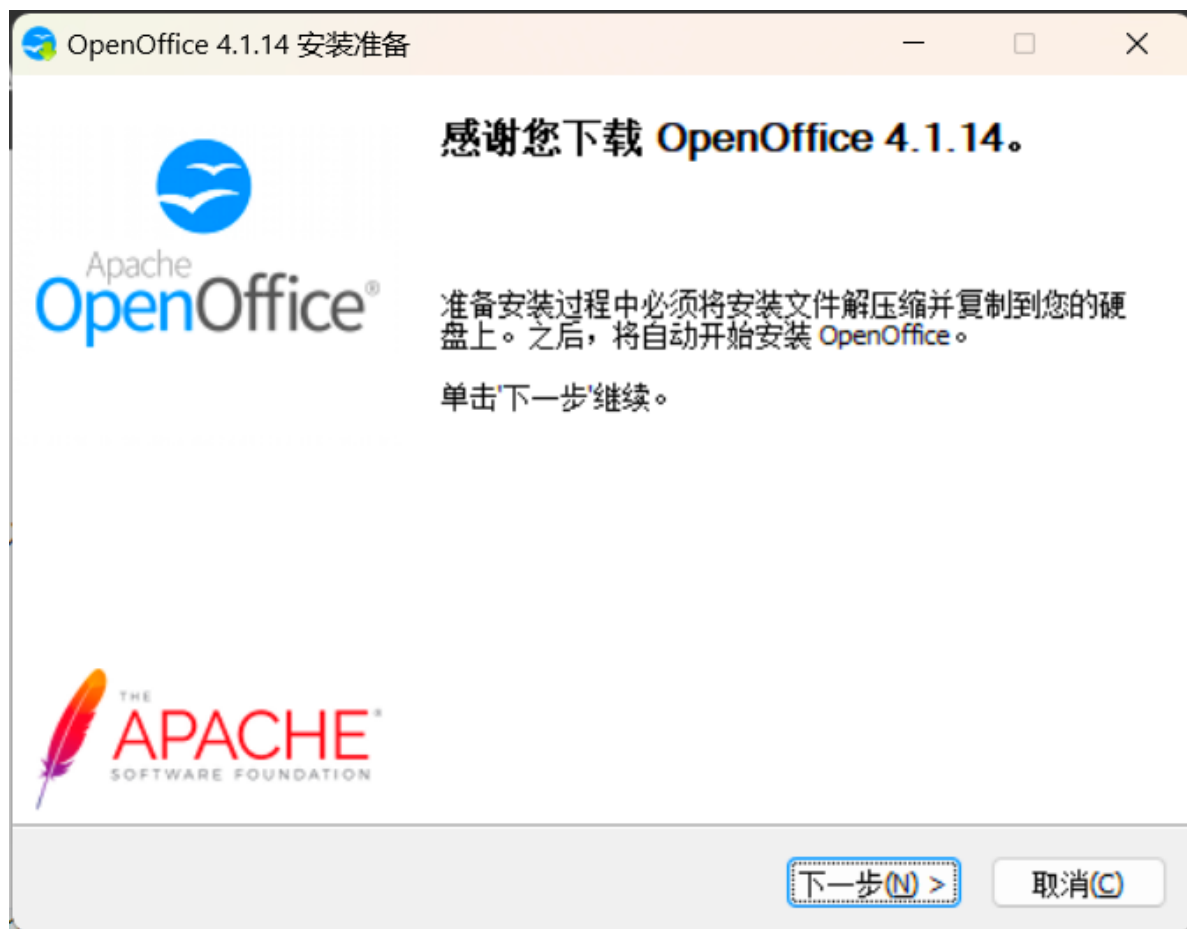
Apache\_OpenOffice\_4.1.14\_Win\_x86\_install\_zh-CN.exe | Scanned for malware ✓

## Other Useful Business Software

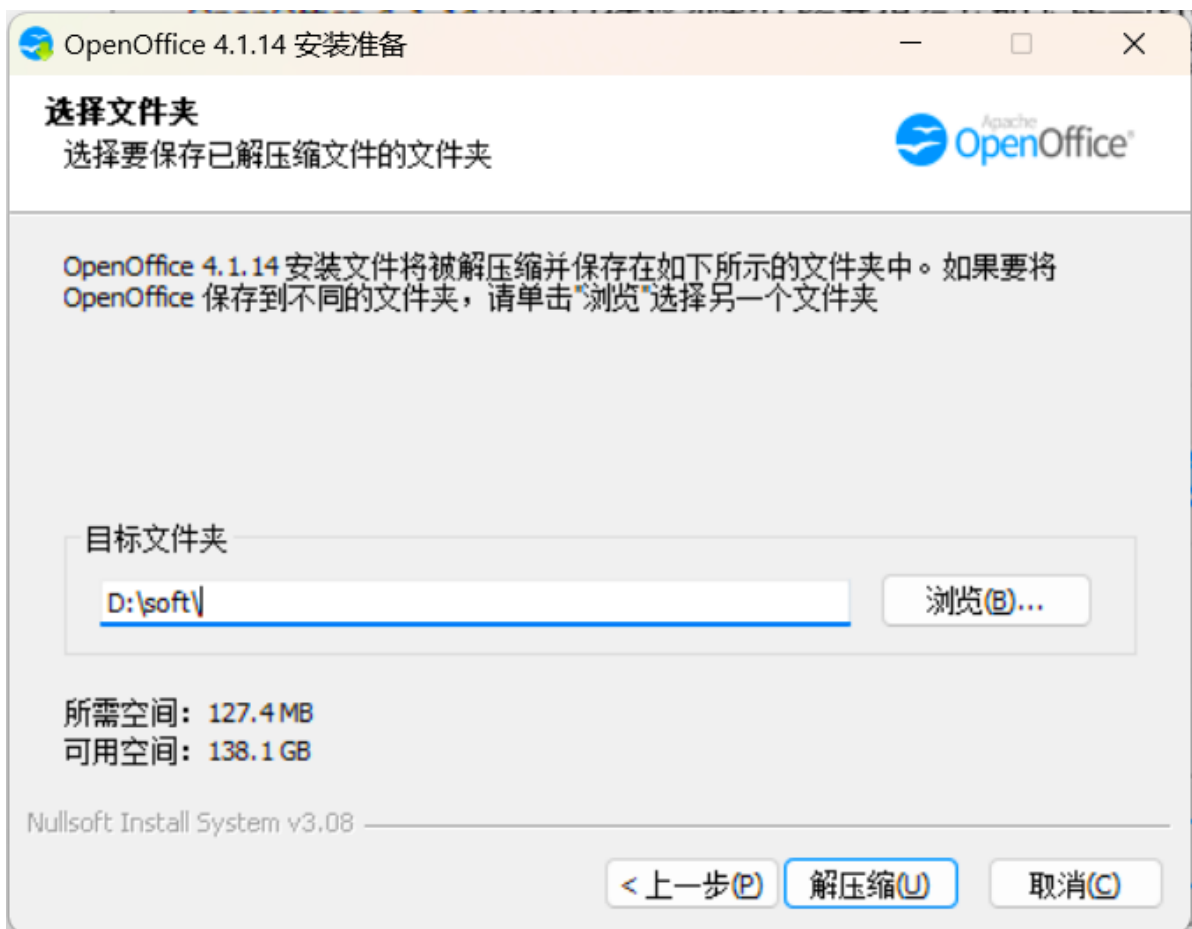
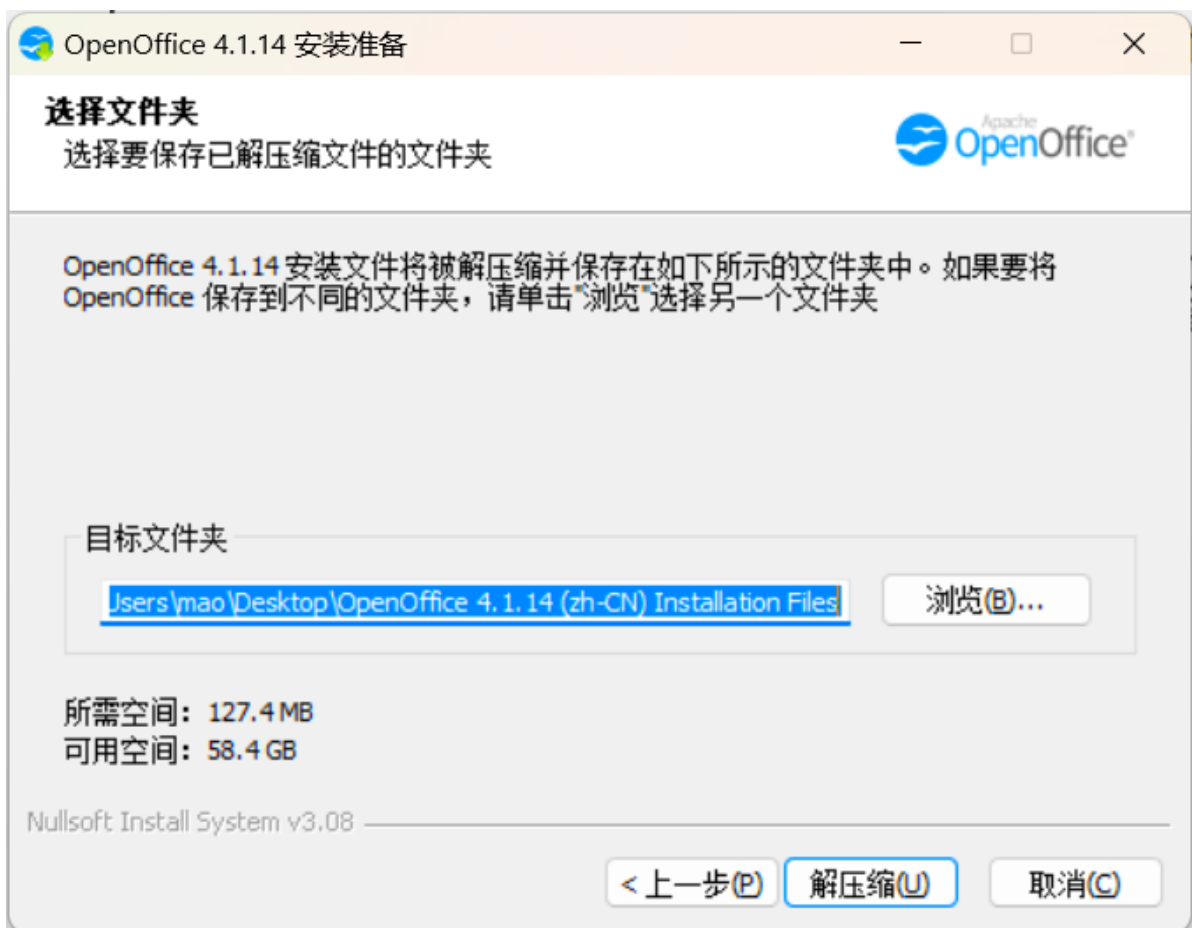
出现这个界面，请等待几秒，即将开始下载

## 第二步：安装

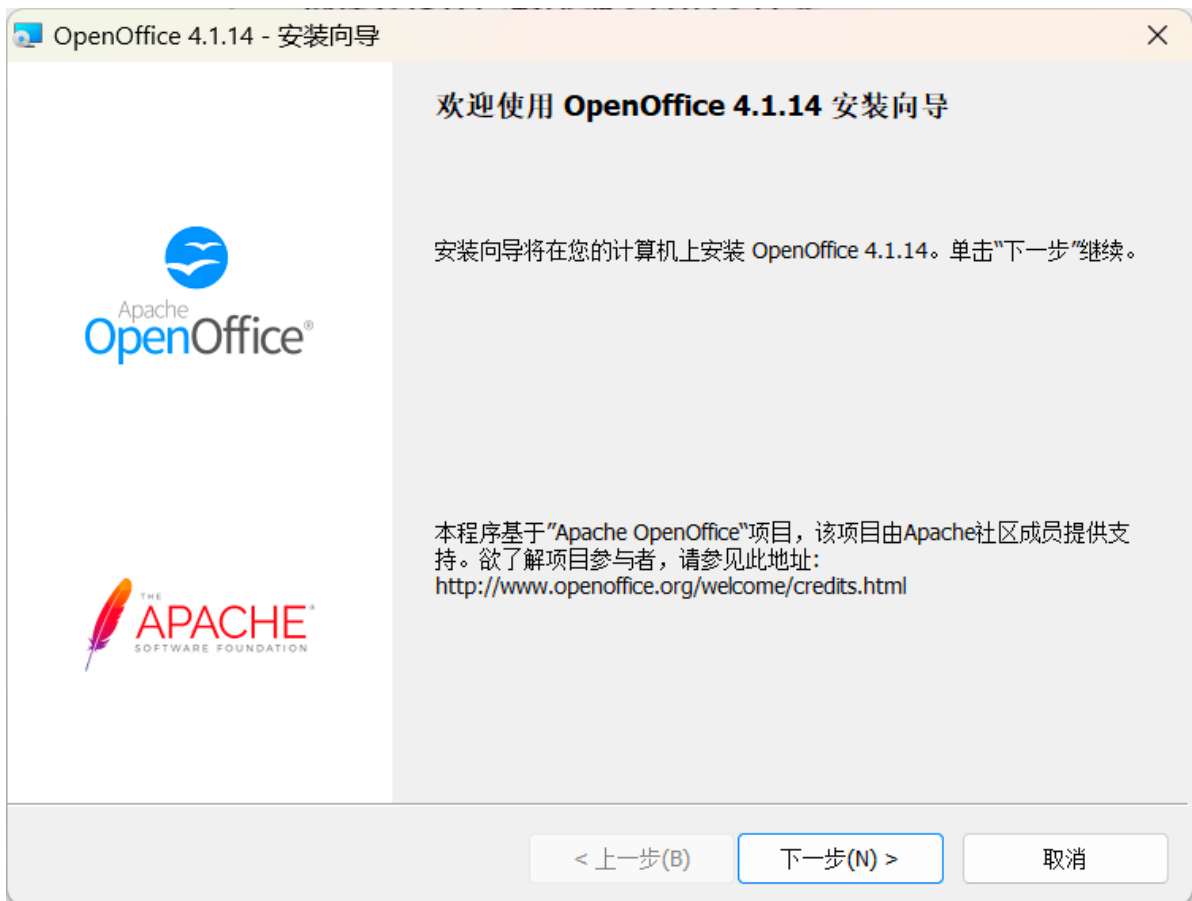
将下载完成的 Apache\_OpenOffice\_4.1.14\_win\_x86\_install\_zh-CN.exe 文件双击打开



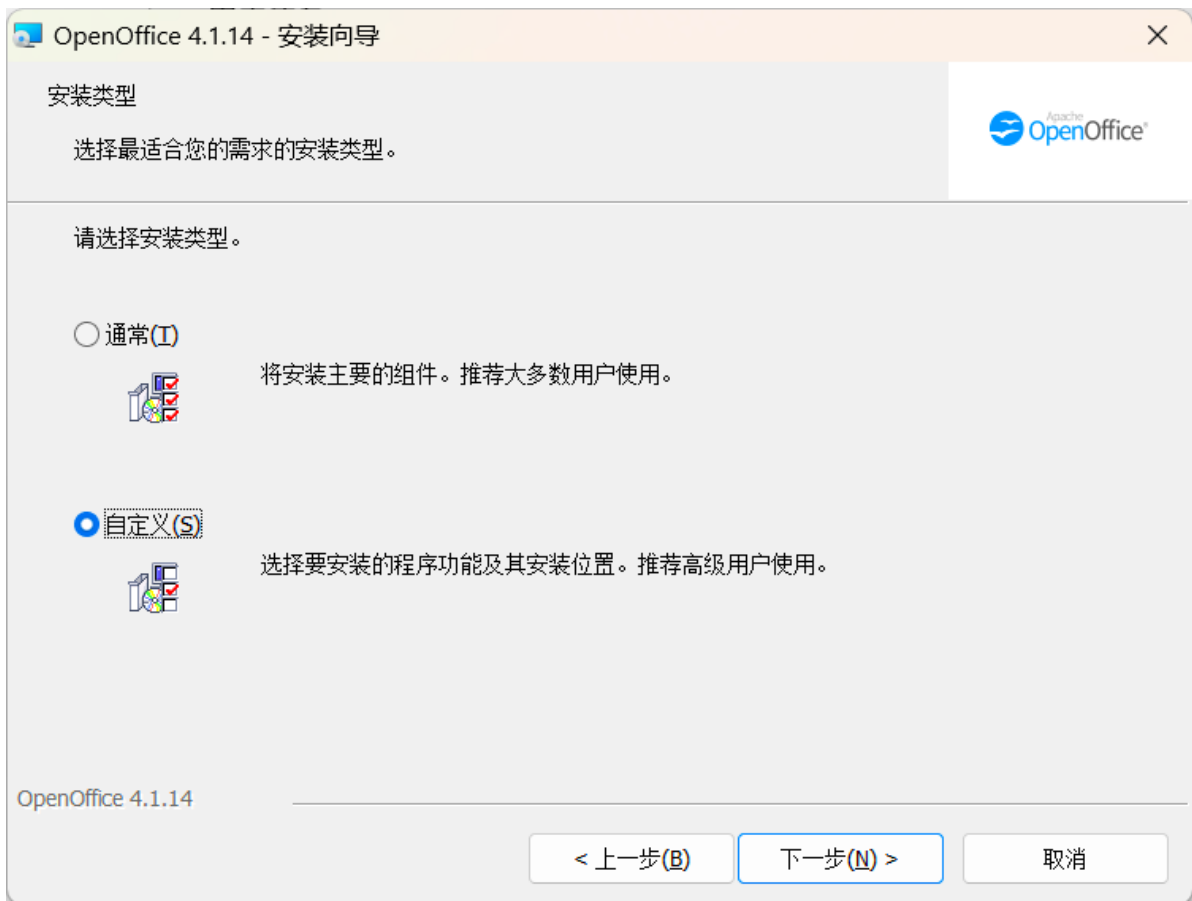
选择安装位置



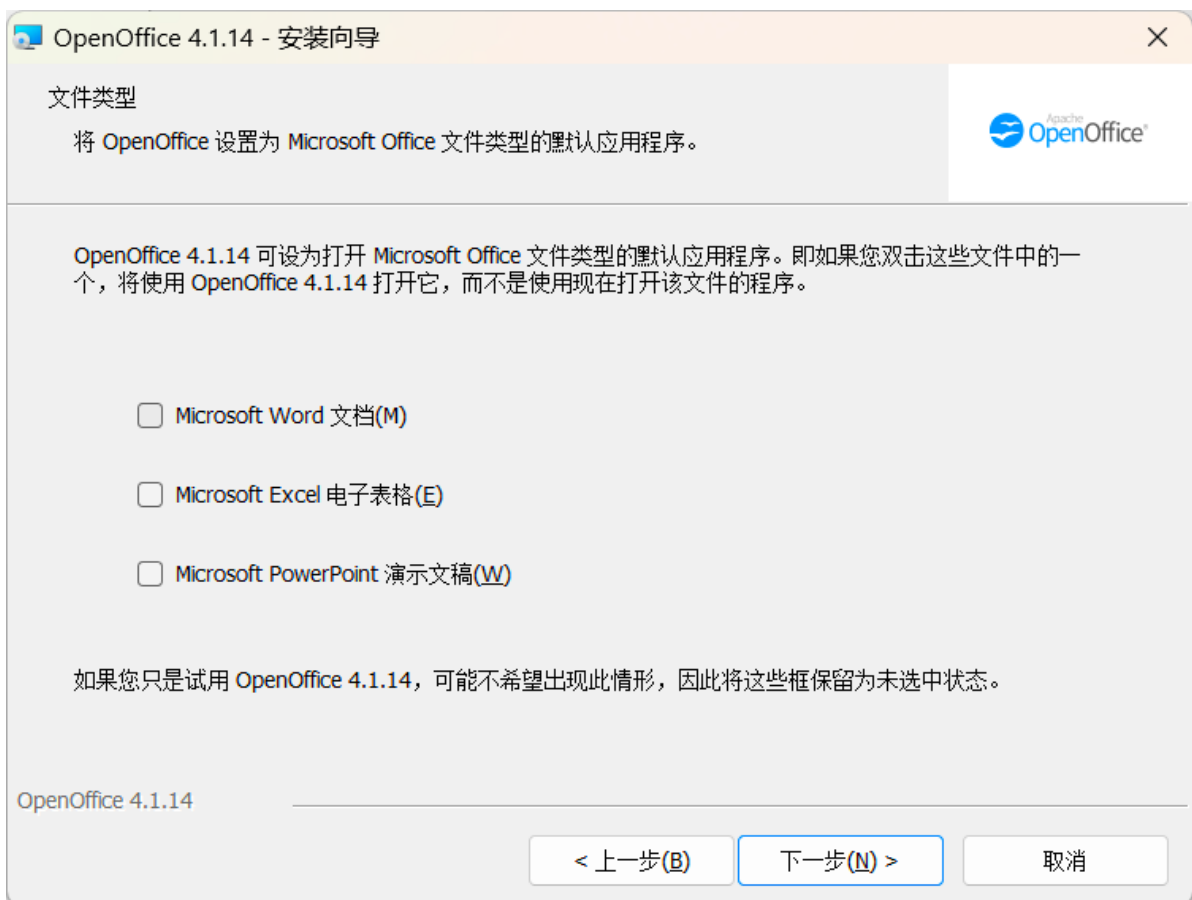
点击下一步



下一步

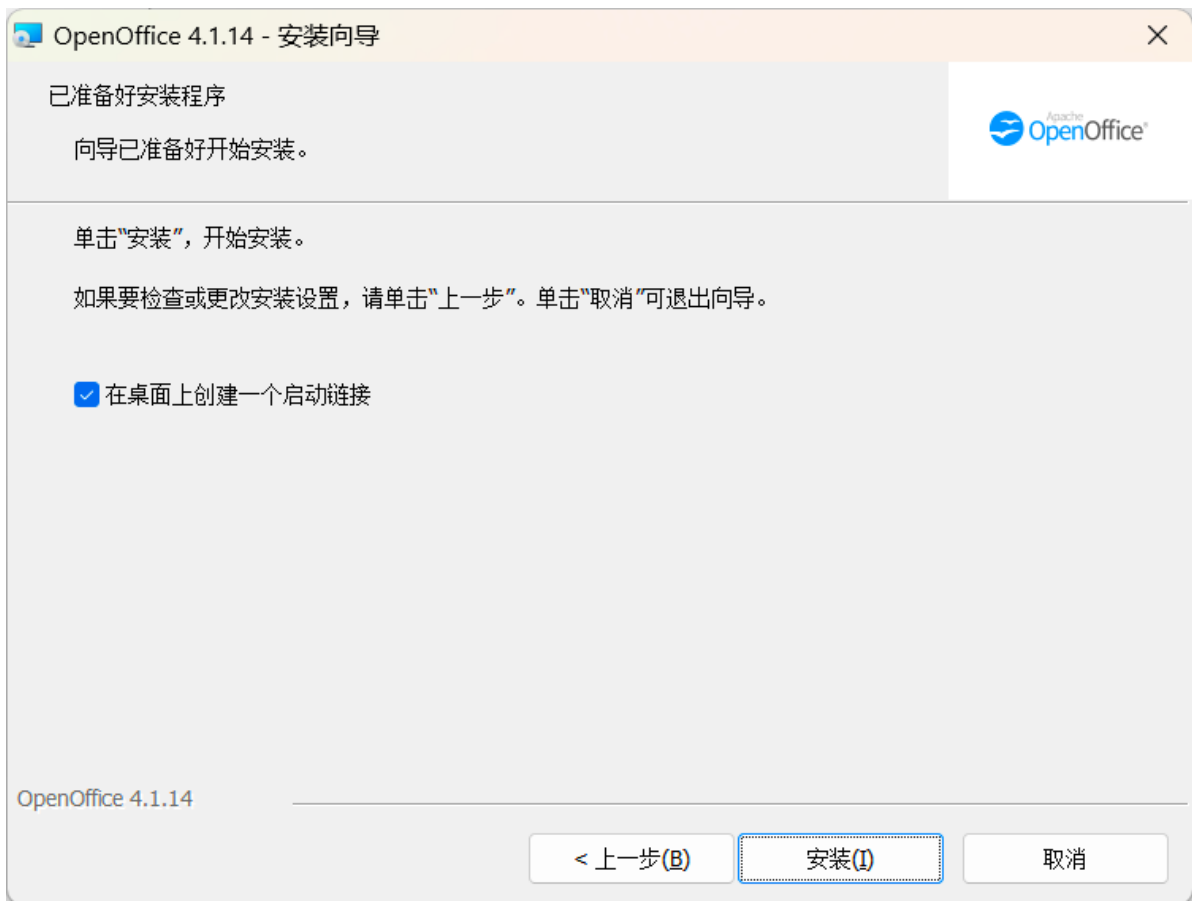


下一步

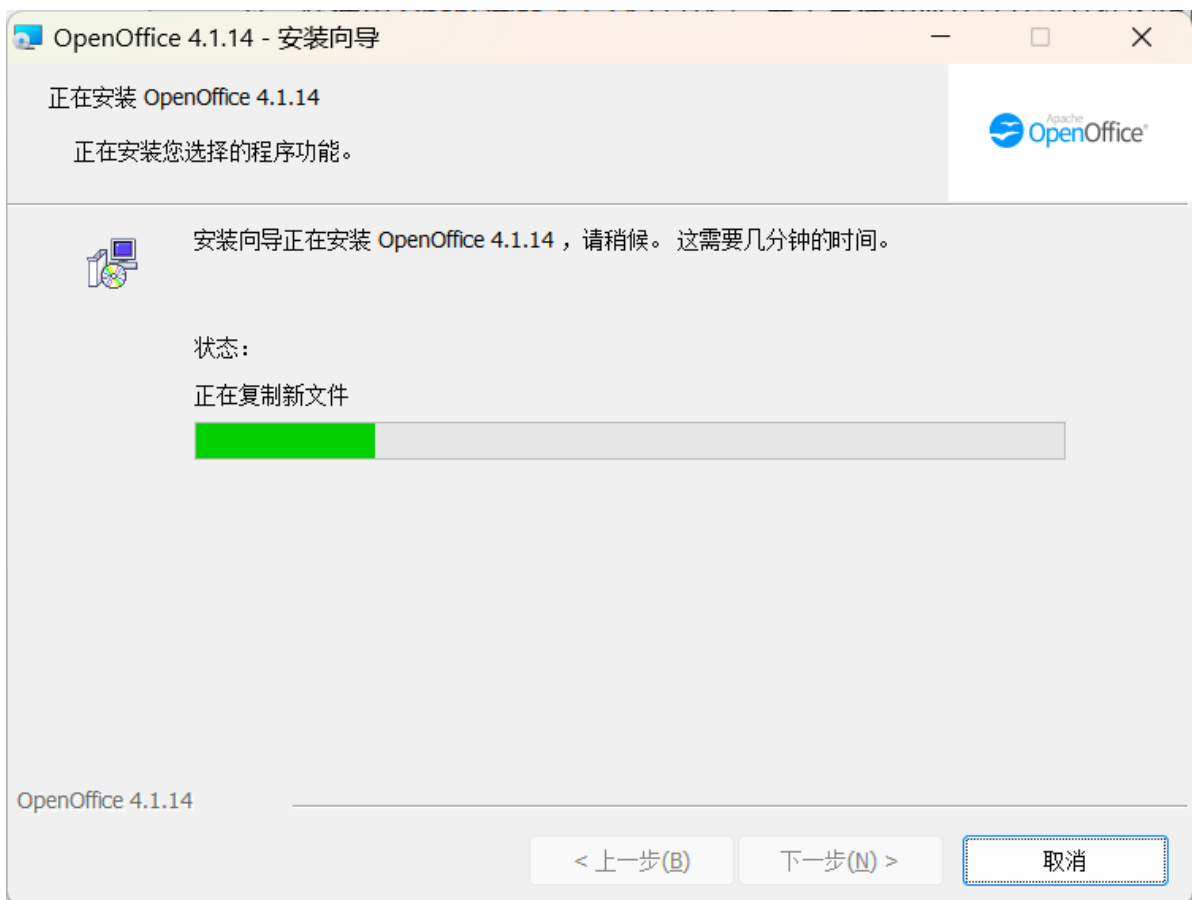


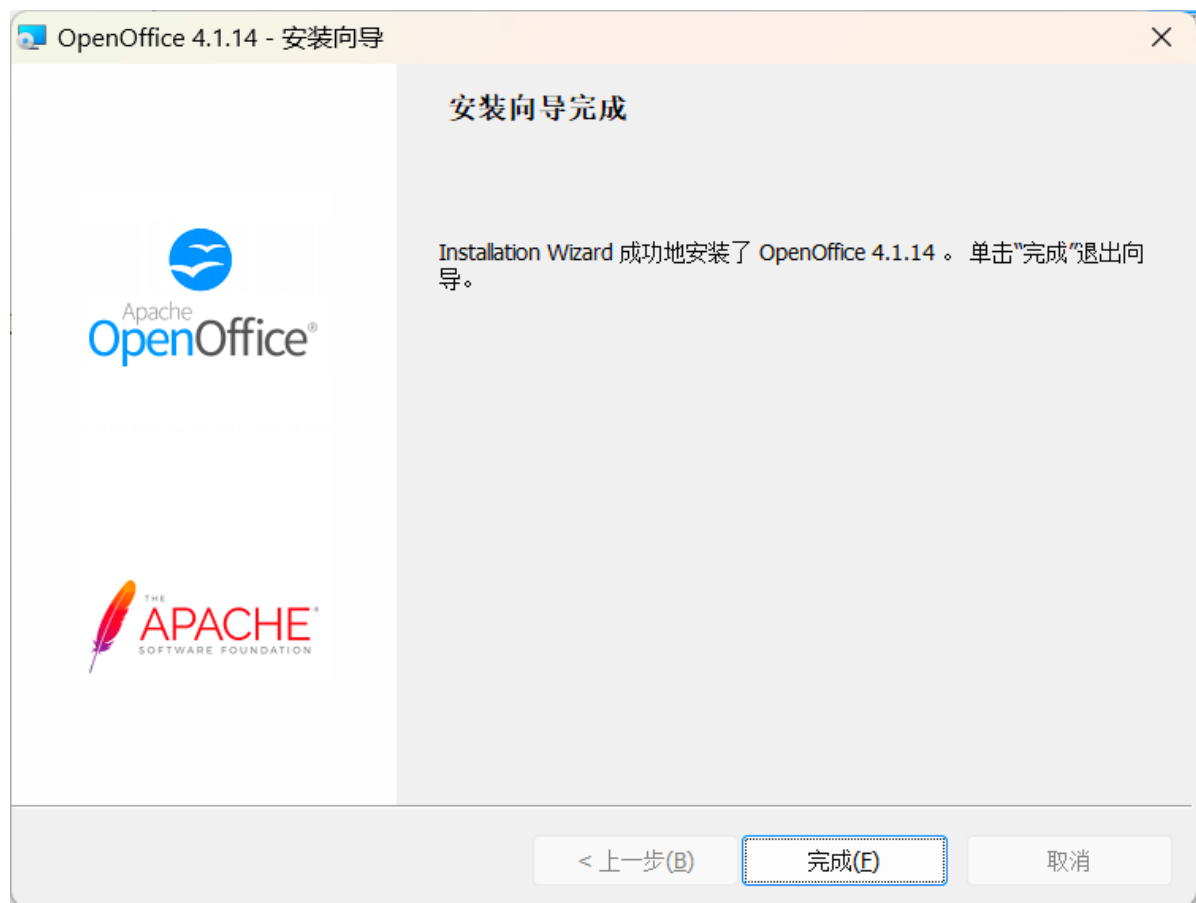
下一步





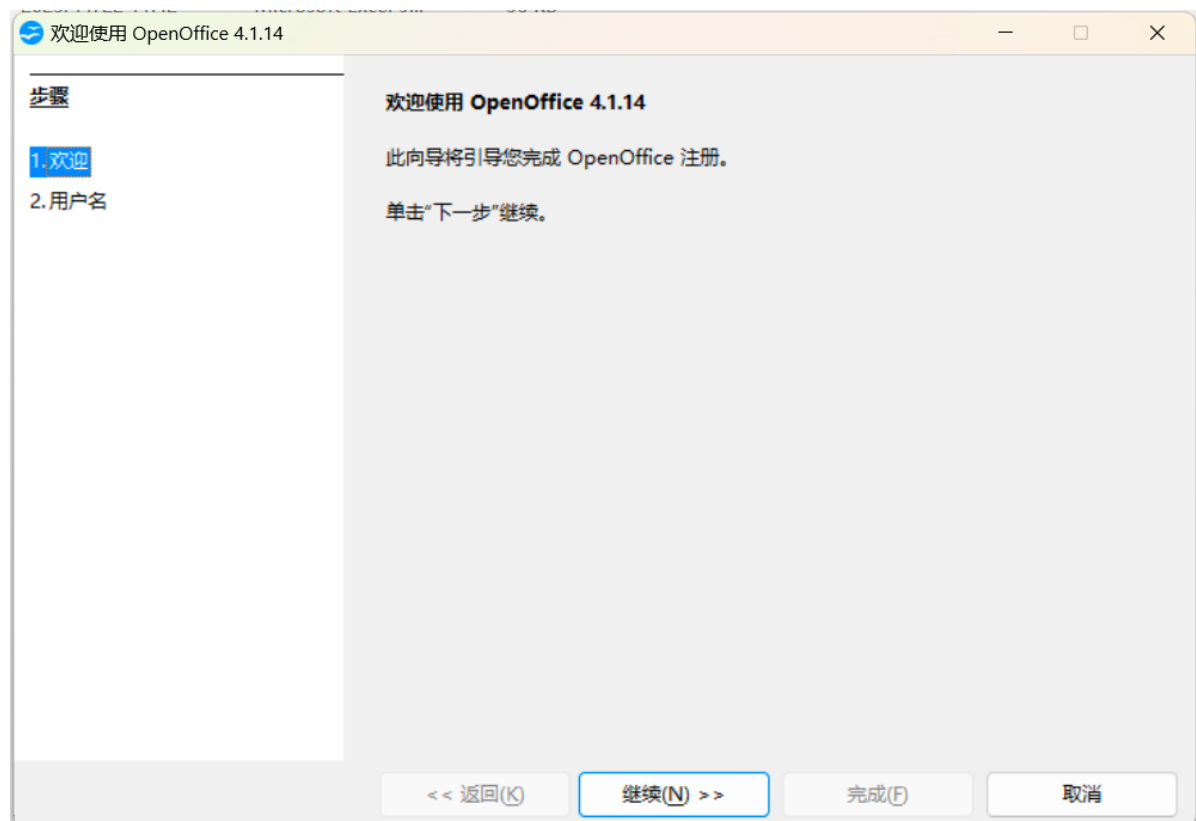
#### 等待安装

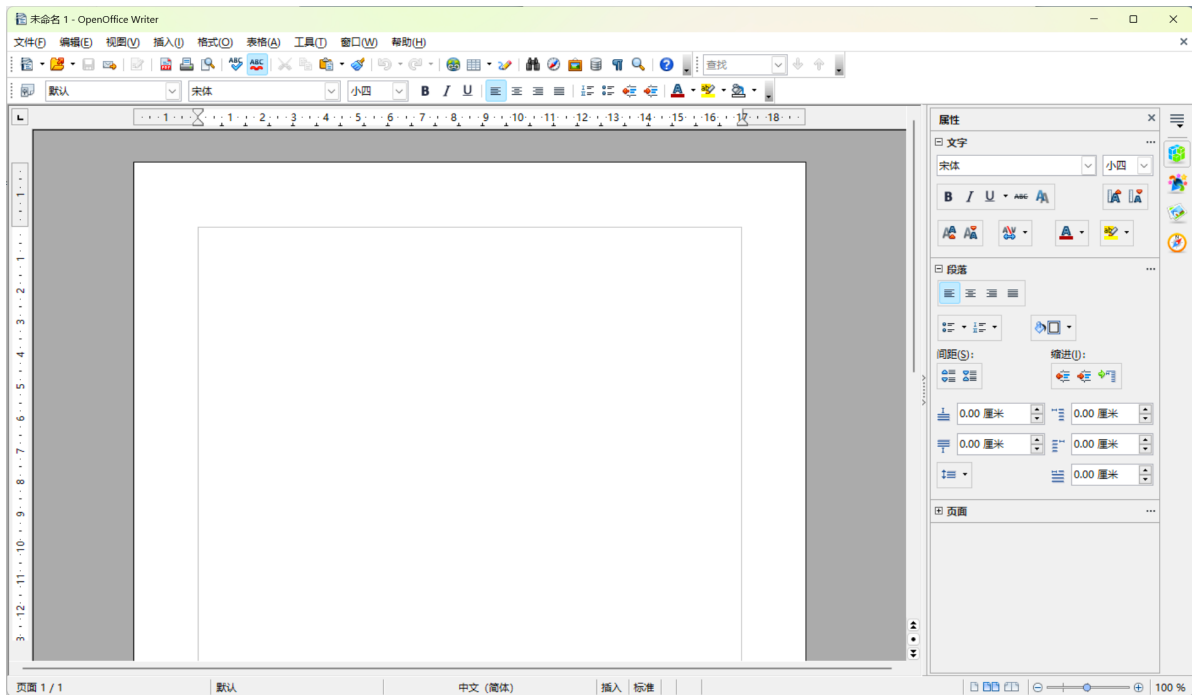
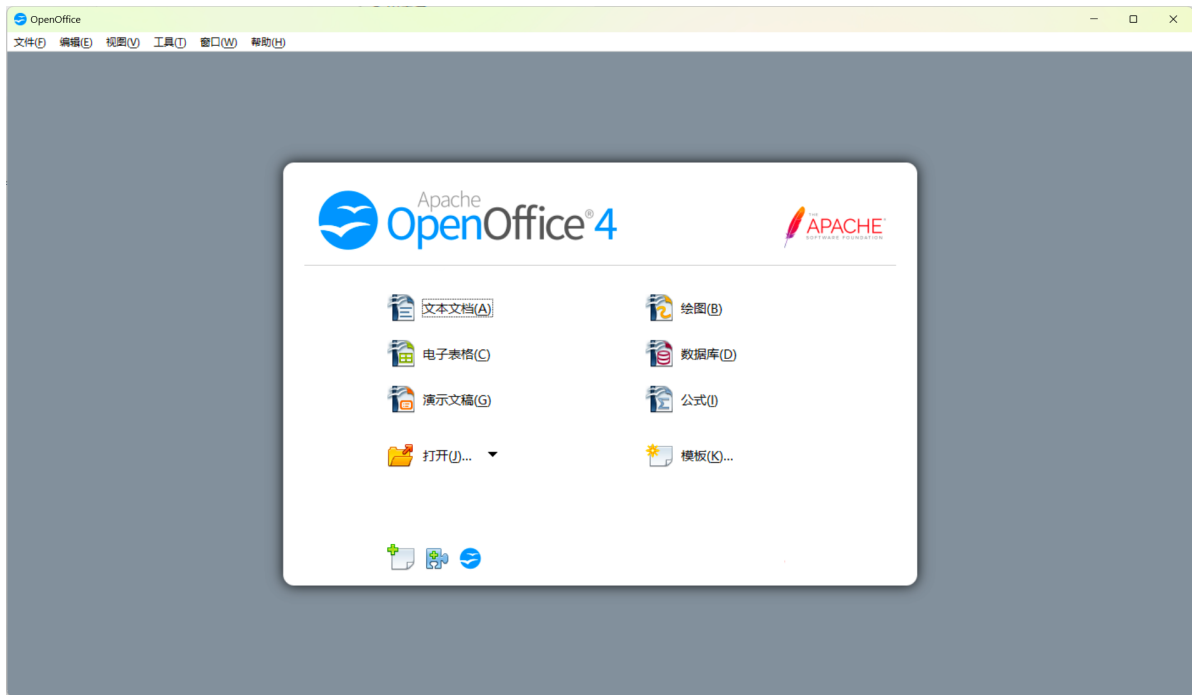




### 第三步：启动GUI程序

双击打开GUI文件，界面如下：





程序根目录如下：

名称	修改日期	类型	大小
help	2023/11/22 16:18	文件夹	
presets	2023/11/22 16:18	文件夹	
program	2023/11/22 16:18	文件夹	
readmes	2023/11/22 16:18	文件夹	
share	2023/11/22 16:18	文件夹	
readme.html	2023/2/9 18:39	Microsoft Edge H...	11 KB
readme.txt	2023/2/9 18:39	文本文档	10 KB

## 第四步：启动服务

进入安装目录program目录下

启动服务的命令：

```
1 ./soffice -headless -accept="socket,host=127.0.0.1,port=8100;urp;" -
  nofirststartwizard
```

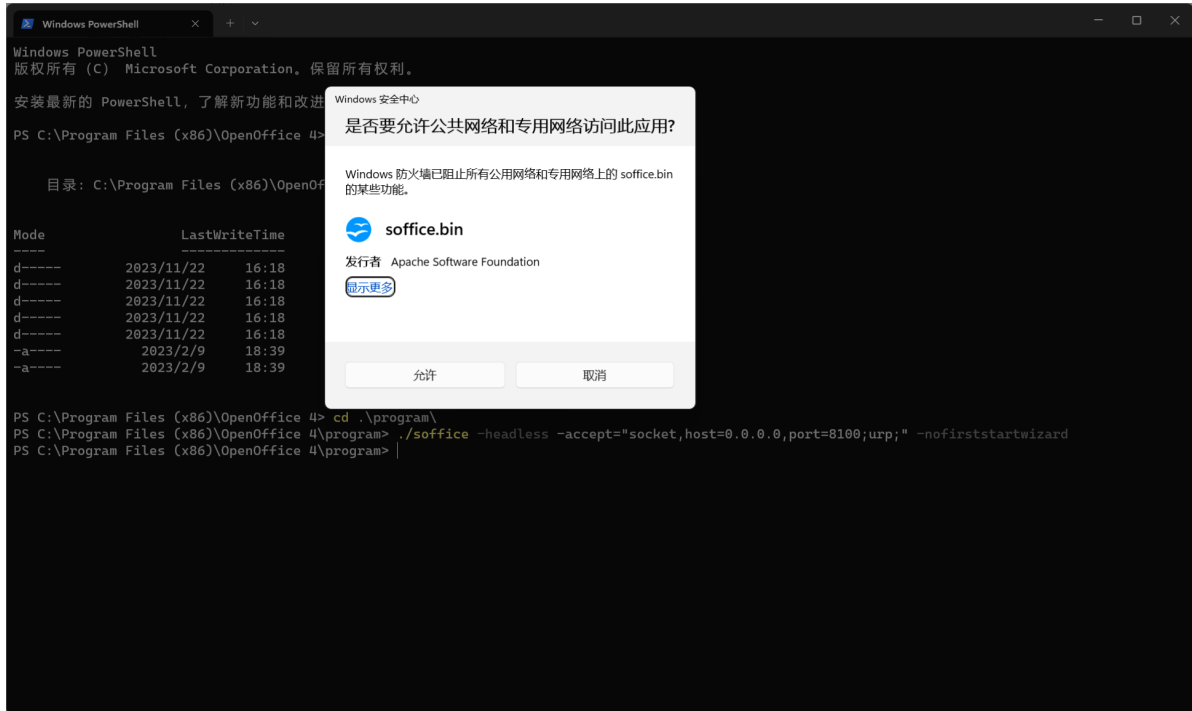
或者：

```
1 ./soffice -headless -accept="socket,host=0.0.0.0,port=8100;urp;" -
  nofirststartwizard
```

- **host**：绑定的ip，配置为0.0.0.0，则远程ip能使用，配置为127.0.0.1，则只能本地访问
- **port**：服务启动的端口号

1	PS C:\Program Files (x86)\OpenOffice 4> ls
2	
3	
4	目录: C:\Program Files (x86)\OpenOffice 4
5	
6	
7	Mode LastwriteTime Length Name
8	----
9	d----- 2023/11/22 16:18 help
10	d----- 2023/11/22 16:18 presets
11	d----- 2023/11/22 16:18 program
12	d----- 2023/11/22 16:18 readmes
13	d----- 2023/11/22 16:18 share
14	-a---- 2023/2/9 18:39 10440 readme.html

```
15 -a----                2023/2/9      18:39      10017 readme.txt
16
17
18 PS C:\Program Files (x86)\OpenOffice 4> cd .\program\
19 PS C:\Program Files (x86)\OpenOffice 4\program> ./soffice -headless -
accept="socket,host=0.0.0.0,port=8100;urp;" -nofirststartwizard
20 PS C:\Program Files (x86)\OpenOffice 4\program>
```



点击允许

不报错，为启动成功

## 第五步：检查是否启动成功

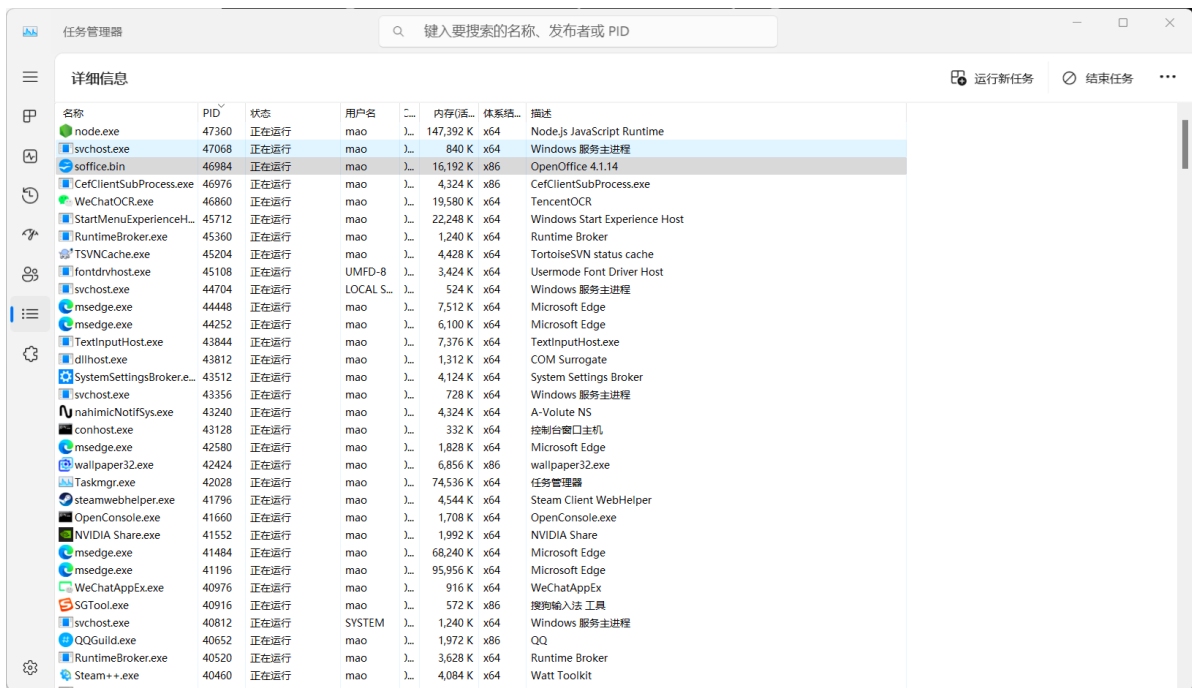
```
1 netstat -ano
```

```
PS C:\Program Files (x86)\OpenOffice 4\program> netstat -ano

活动连接

协议 本地地址 外部地址 状态 PID
TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 1248
TCP 0.0.0.0:443 0.0.0.0:0 LISTENING 20696
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING 4
TCP 0.0.0.0:902 0.0.0.0:0 LISTENING 40236
TCP 0.0.0.0:912 0.0.0.0:0 LISTENING 40236
TCP 0.0.0.0:2179 0.0.0.0:0 LISTENING 3980
TCP 0.0.0.0:3306 0.0.0.0:0 LISTENING 5860
TCP 0.0.0.0:3690 0.0.0.0:0 LISTENING 6480
TCP 0.0.0.0:5040 0.0.0.0:0 LISTENING 5148
TCP 0.0.0.0:8080 0.0.0.0:0 LISTENING 10040
TCP 0.0.0.0:8081 0.0.0.0:0 LISTENING 39444
TCP 0.0.0.0:8100 0.0.0.0:0 LISTENING 46984
TCP 0.0.0.0:8443 0.0.0.0:0 LISTENING 6384
TCP 0.0.0.0:13688 0.0.0.0:0 LISTENING 6204
TCP 0.0.0.0:26501 0.0.0.0:0 LISTENING 31692
TCP 0.0.0.0:27036 0.0.0.0:0 LISTENING 17548
TCP 0.0.0.0:33060 0.0.0.0:0 LISTENING 5860
```

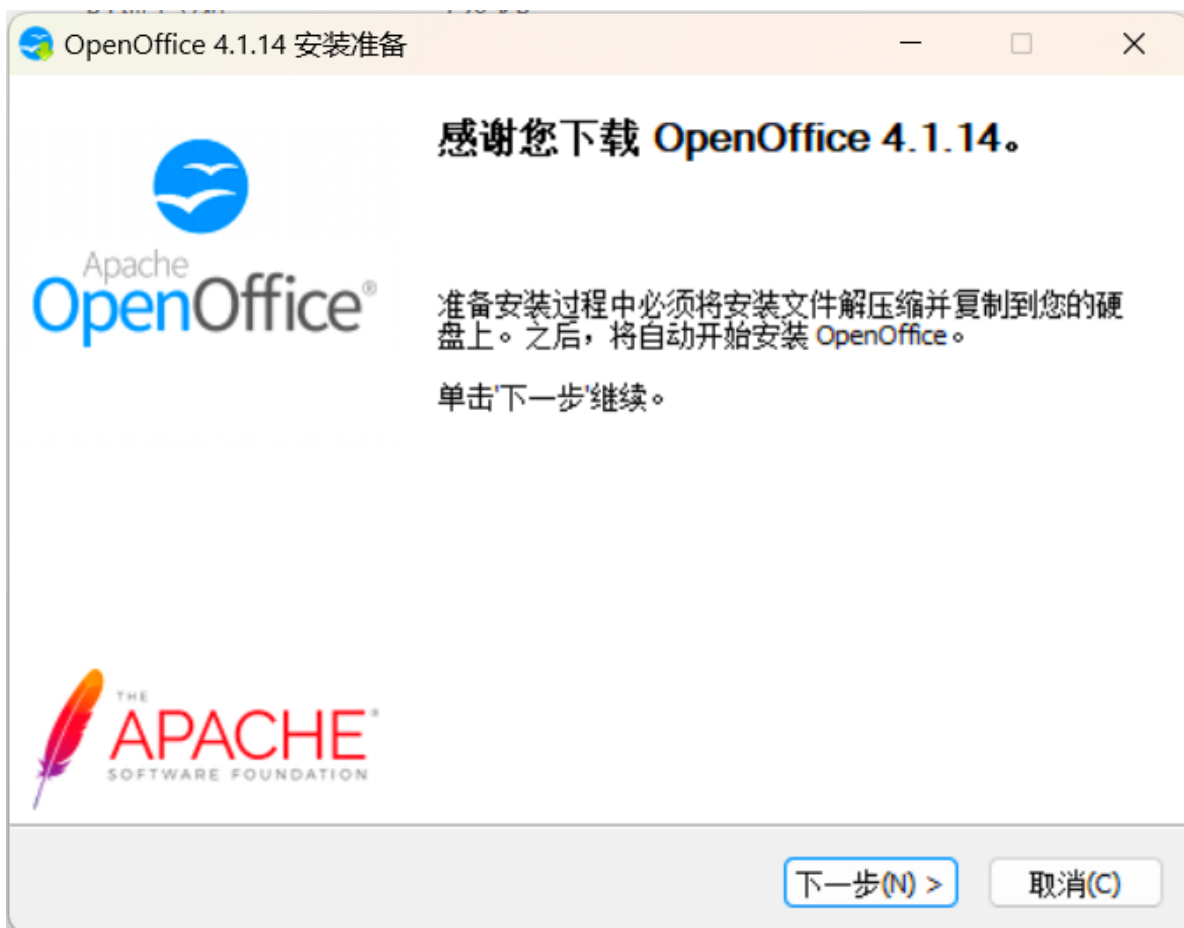
可以看到8100端口已经在使用了，pid为46984

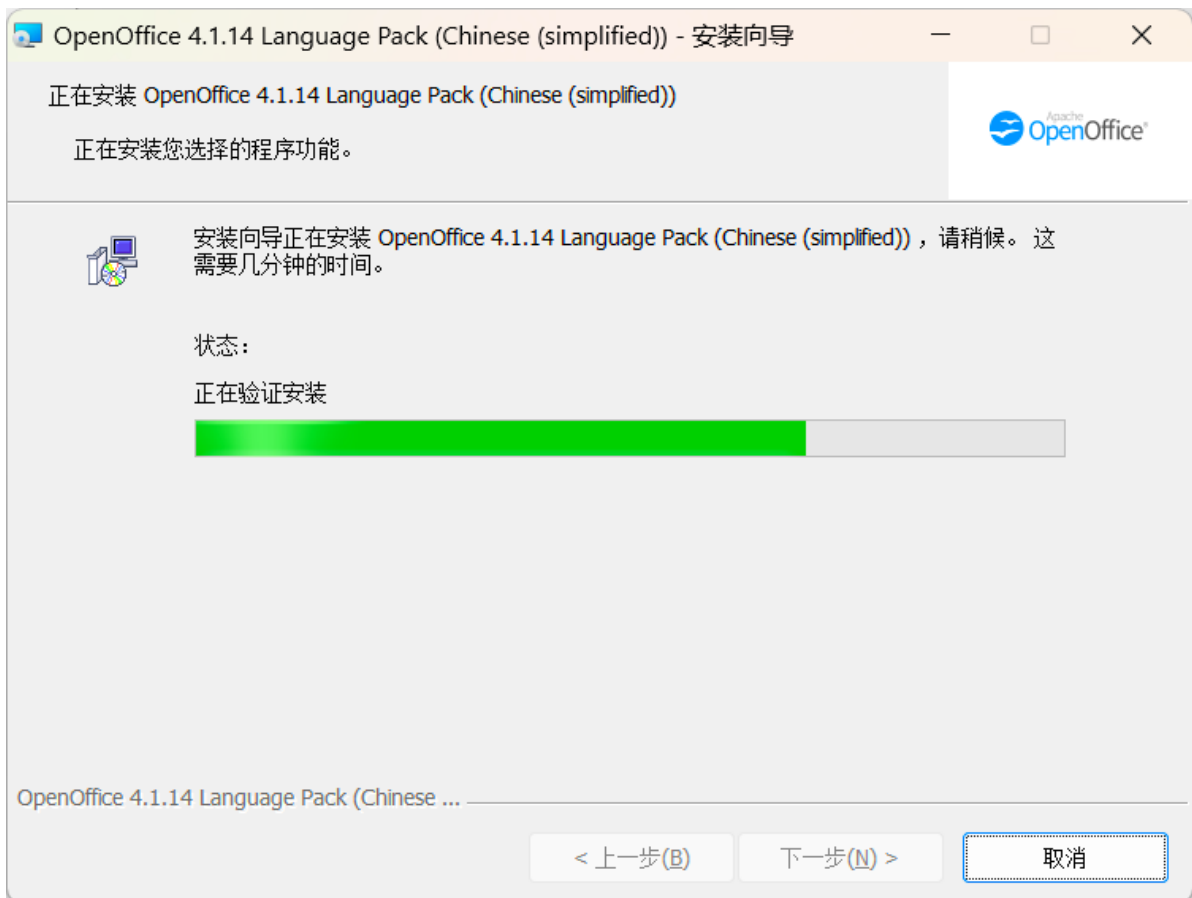


打开任务管理器，可以看到46984的确为openoffice

### 第六步：安装中文语言包

如果下载的openoffice本来就是中文的，则没有必要再安装，可以跳过此步骤







# Linux安装

以centos为例

## 第一步：下载

进入下载界面：<https://sourceforge.net/projects/openofficeorg.mirror/files/>

Home / 4.1.14			
Name	Modified	Size	Downloads / Week
Parent folder			
binaries	2023-02-23		560,728
source	2023-02-23		12
README.md	2023-02-23	143 Bytes	57
Totals: 3 Items		143 Bytes	560,797
For more information please have a look at our <a href="#">Release Notes</a> .			

tr	2023-02-23	1,971
vi	2023-02-23	243
zh-CN	2023-02-23	2,313
zh-TW	2023-02-23	3,028
Totals: 42 Items		560,728

Apache_OpenOffice_4.1.14_Win_x86_install_zh-CN.exe	2023-02-23	131.1 MB	1,587
Apache_OpenOffice_4.1.14_Win_x86_langpack_zh-CN.exe	2023-02-23	13.3 MB	113
Apache_OpenOffice_4.1.14_MacOS_x86-64_install_zh-CN.dmg	2023-02-13	165.4 MB	91
Apache_OpenOffice_4.1.14_MacOS_x86-64_langpack_zh-CN.dmg	2023-02-13	13.1 MB	18
Apache_OpenOffice_4.1.14_Linux_x86_install-rpm_zh-CN.tar.gz	2023-02-13	148.5 MB	11
Apache_OpenOffice_4.1.14_Linux_x86_langpack-deb_zh-CN.tar.gz	2023-02-13	13.9 MB	1
Apache_OpenOffice_4.1.14_Linux_x86_langpack-rpm_zh-CN.tar.gz	2023-02-13	13.8 MB	2
Apache_OpenOffice_4.1.14_Linux_x86_install-deb_zh-CN.tar.gz	2023-02-13	146.8 MB	6
Apache_OpenOffice_4.1.14_Linux_x86-64_langpack-deb_zh-CN.tar.gz	2023-02-13	13.9 MB	17
Apache_OpenOffice_4.1.14_Linux_x86-64_langpack-rpm_zh-CN.tar.gz	2023-02-13	13.8 MB	28
Apache_OpenOffice_4.1.14_Linux_x86-64_install-rpm_zh-CN.tar.gz	2023-02-13	156.6 MB	350
Apache_OpenOffice_4.1.14_Linux_x86-64_install-deb_zh-CN.tar.gz	2023-02-13	155.0 MB	89

这次选择第五个：[Apache\\_OpenOffice\\_4.1.14\\_Linux\\_x86\\_install-rpm\\_zh-CN.tar.gz](#)

## 第二步：创建文件夹

连接linux服务器，在/usr/local目录下创建openoffice文件夹

```
1 | cd /usr/local
2 | mkdir openoffice
```

## 第三步：上传文件

将刚刚下载完成的压缩包上传到linux服务器的openoffice文件夹里

## 第四步：解压文件

命令：

```
1 | tar -zxvf Apache_OpenOffice_4.1.14_Linux_x86_install-rpm_zh-CN.tar.gz
```

## 第五步：安装rpm文件

进入 `zh-CN/RPMS/` 目录下安装rpm文件

命令：

```
1 | yum localinstall *.rpm
```

安装完后会在当前目录下生成一个desktop-integration目录

进入desktop-integration目录，执行：

```
1 | yum localinstall openoffice4.1.14-redhat-menus-4.1.14-9808.noarch.rpm
```

## 第六步：启动

命令：

```
1 | /opt/openoffice4/program/soffice -headless -  
    accept="socket,host=127.0.0.1,port=8100;urp;" -nofirststartwizard
```

- 如果报错：错误：Could not find a Java Runtime Environment，这是没有安装jdk导致的
- 如果报错：/opt/openoffice4/program/soffice.bin: error while loading shared libraries: libxext.so.6: cannot open shared object file: No such file or directory，这是由于缺少lib库文件，执行yum install libxext.x86\_64，然后执行cp -a usr/lib64/libxext.so.6 /opt/openoffice4/program/ 复制到openoffice目录
- 如果报错：no suitable windowing system found, exiting，这是没有图形界面支持导致的，openoffice运行需要图形界面，执行yum groupinstall "X window System"
- 预览如果出现中文乱码、中文不显示的问题，还需要安装安装中文字体，这个问题只有在linux才会有

## Docker安装

### 第一步：搜索镜像

命令：

```
1 | docker search openoffice
```

```
1 | PS C:\Users\mao> docker search openoffice  
2 | NAME                                DESCRIPTION  
3 |                                     STARS      OFFICIAL  AUTOMATED  
3 | kasmweb/openoffice  
   | 2  
4 | rafaelthelho/openoffice3-daemon      Openoffice 3 headless daemon Image  
   | running t... 3  
5 | umnelevator/openoffice  
   | 4  
6 | xiaojun207/openoffice4-daemon         Image running the OpenOffice 4 soffice  
   | daemo... 7  
7 | expertsystems/openoffice             OpenOffice in server mode  
   | 5  
   | [OK]
```

8	upresentit/openoffice port 81... 1	OpenOffice 4.1 running as service on
9	timonier/openoffice suite 0	The free and Open Source productivity [OK]
10	chrisdaish/openoffice forwar... 0	Unofficial build of Apache OpenOffice [OK]
11	rolesle/openoffice 0	Apache OpenOffice4
12	xzxiaoshan/openoffice 主... 1	openoffice 镜像，默认启动openoffice服务，
13	cafetime/openoffice-jdk 0	openoffice-jdk [OK]
14	evan11/openoffice4-java8 spring... 2	基于openjdk:8-jdk 集成openoffice4 部署
15	bdhbdhh/openoffice4 0	OpenOffice 4 server mode
16	goodrainapps/openoffice 0	
17	kuschzzp/openoffice 文... 1	openoffice:4.1.13镜像，内置了字体，解决中
18	patriciorangles/openoffice-headless 0	Headless de OpenOffice para OpenERP v7
19	akiraheid/openoffice processing... 0	An office software suite for word
20	wangzhishuai/openoffice EXPOSE 8100 1	openoffice4.1.6+jdk1.8+linux64
21	xzxiaoshan/openoffice-base 0	OpenOffice 基础环境镜像
22	wjf8882300/openoffice 0	用于word、excel、powerpoint转为pdf文件
23	eliu/openoffice 问... 0	容器化启动 openoffice 并解决中文无法显示的
24	saflloresg/openoffice 0	
25	bdhbdhh/openoffice3 0	OpenOffice 3 server mode
26	9541/openoffice 0	
27	openoffice200/docker101tutorial 0	
28	PS C:\Users\mao>	

本次使用xiaojun207的镜像

## 第二步：拉取镜像

命令：

```
1 | docker pull xiaojun207/openoffice4-daemon
```

等待下载完成:

```
openoffice200/docker101tutorial 0
PS C:\Users\mao> docker pull xiaojun207/openoffice4-daemon
Using default tag: latest
latest: Pulling from xiaojun207/openoffice4-daemon
ab5ef0e58194: Extracting [=====>] 75.78MB/75.78MB
b345d96ba7d9: Download complete
984a22aaecc3: Download complete
94cb19074099: Download complete
cc43233dd098: Downloading [=====>] 13.88MB/31.44MB
2804181f079c: Downloading [=====>] 9.51MB/102.7MB
235c5d6b9cf4: Waiting
9205c67b4251: Waiting
ecde98a5a60d: Waiting
383110ac3d5d: Waiting
|
```

```
1 PS C:\Users\mao> docker pull xiaojun207/openoffice4-daemon
2 Using default tag: latest
3 latest: Pulling from xiaojun207/openoffice4-daemon
4 ab5ef0e58194: Pull complete
5 b345d96ba7d9: Pull complete
6 984a22aaecc3: Pull complete
7 94cb19074099: Pull complete
8 cc43233dd098: Pull complete
9 2804181f079c: Pull complete
10 235c5d6b9cf4: Pull complete
11 9205c67b4251: Pull complete
12 ecde98a5a60d: Pull complete
13 383110ac3d5d: Pull complete
14 Digest:
   sha256:ba9b546d8fefe4d623be319efa41735c8dd36a5c040e8f70461e2ca1ee842d62
15 Status: Downloaded newer image for xiaojun207/openoffice4-daemon:latest
16 docker.io/xiaojun207/openoffice4-daemon:latest
17
18 what's Next?
19   view summary of image vulnerabilities and recommendations → docker scout
   quickview xiaojun207/openoffice4-daemon
20 PS C:\Users\mao>
```

### 第三步: 查看镜像是否拉取成功

命令:

```
1 docker images
```

或者:

```
1 docker images xiaojun207/openoffice4-daemon
```

```

1 PS C:\Users\mao> docker images xiaojun207/openoffice4-daemon
2 REPOSITORY          TAG          IMAGE ID      CREATED      SIZE
3 xiaojun207/openoffice4-daemon  latest      c0a14707c109  3 years ago  1.2GB
4 PS C:\Users\mao>

```

## 第四步：运行镜像

命令：

```

1 docker run -d --name openoffice -p 8300:8100 -v D:/docker/openoffice:/data/
  xiaojun207/openoffice4-daemon:latest

```

```

1 PS C:\Users\mao> docker run -d --name openoffice -p 8300:8100 -v
  D:/docker/openoffice:/data/ xiaojun207/openoffice4-daemon:latest
2 4da0e56c24da178145e531c05922ec2cd3592d88ab27ec2f4c1114f9a74dcdfa
3 PS C:\Users\mao>

```

## 第五步：查看容器是否运行

命令：

```

1 docker ps

```

```

1 PS C:\Users\mao> docker ps
2 CONTAINER ID   IMAGE                                COMMAND
3 4da0e56c24da   xiaojun207/openoffice4-daemon:latest  "uid_entrypoint /bin..."
   13 seconds ago Up 13 seconds    0.0.0.0:8300->8100/tcp   openoffice
4 619cebf9f2dd   7fff914c4a61                         "/dashboard --insecu..."
   15 minutes ago Up 15 minutes    k8s_kubernetes-
dashboard_kubernetes-dashboard-68955f84f4-ccwfd_kubernetes-
dashboard_fd1c36df-d824-4d4f-a52a-a5f7152d8fa2_70
5 d9aa185088d1   7801cfc6d5c0                         "/metrics-sidecar"
   15 minutes ago Up 15 minutes    k8s_dashboard-
metrics-scraper_dashboard-metrics-scraper-748b4f5b9d-g4h26_kubernetes-
dashboard_b76f2aa6-e30a-40f8-b865-2a9a97796ee0_37
6 PS C:\Users\mao>

```

第一个为openoffice，状态是up，则为启动成功，外部端口为8300，第二个和第三个为k8s

日志如下：

```
1 PS C:\Users\mao> docker logs openoffice
2 Starting soffice daemon in headless mode as user default [id
   uid=10001(default) gid=0(root) groups=0(root)]
3         soffice daemon will listen on Port 8100
4 PS C:\Users\mao>
```

## SpringBoot整合openoffice

---

### 概述

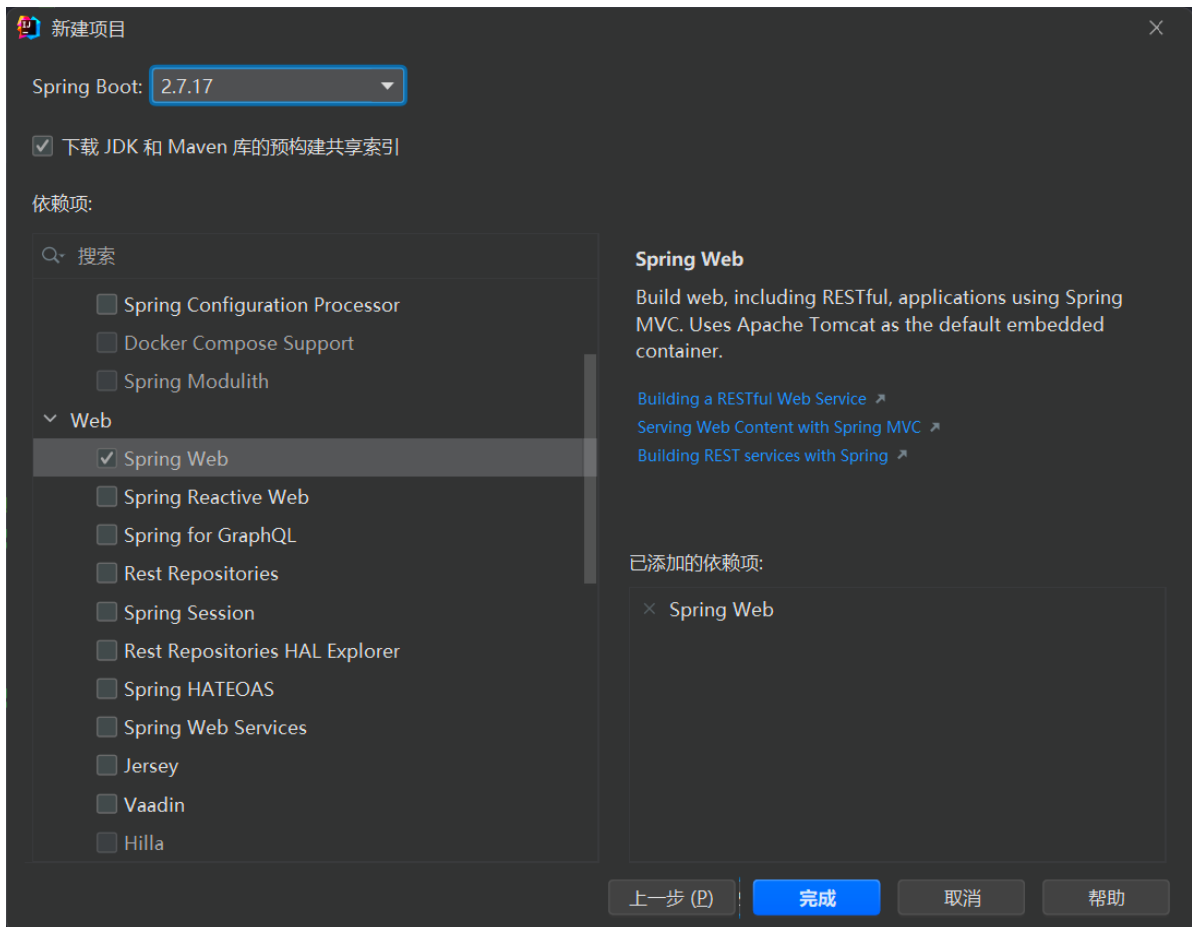
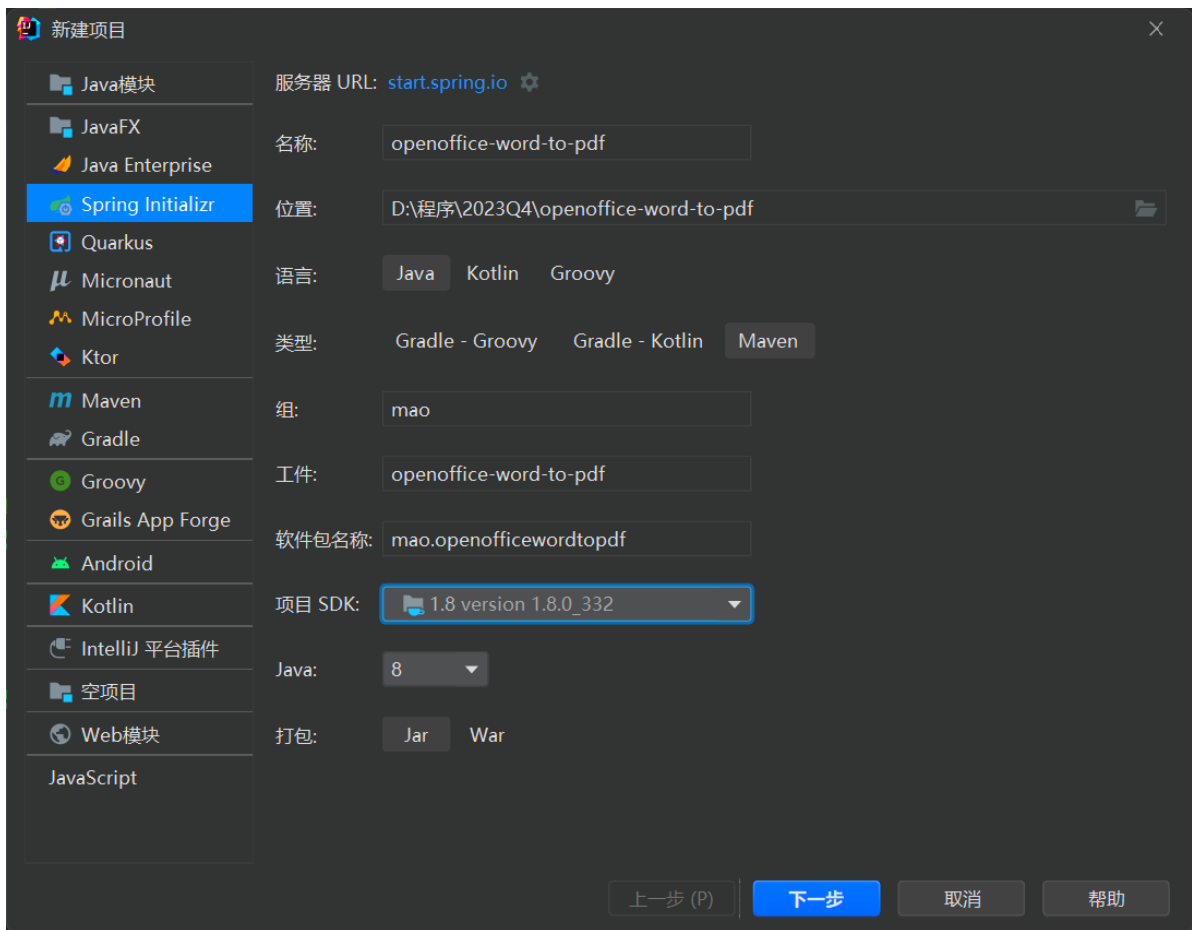
---

因业务需要，需要使用SpringBoot将docx文件转换成PDF文件，再将PDF文件响应给用户

### docx转pdf

---

#### 第一步：创建Springboot程序





## 第二步：配置maven

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6         <groupId>org.springframework.boot</groupId>
7         <artifactId>spring-boot-starter-parent</artifactId>
8         <version>2.7.1</version>
9         <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>mao</groupId>
12    <artifactId>openoffice-word-to-pdf</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>openoffice-word-to-pdf</name>
15    <description>openoffice-word-to-pdf</description>
16    <properties>
17        <java.version>1.8</java.version>
18    </properties>
19    <dependencies>
20        <dependency>
21            <groupId>org.springframework.boot</groupId>
22            <artifactId>spring-boot-starter-web</artifactId>
23        </dependency>
24
25        <dependency>
26            <groupId>org.springframework.boot</groupId>
27            <artifactId>spring-boot-starter-test</artifactId>
28            <scope>test</scope>
29        </dependency>
30
31        <!--spring-boot lombok-->
32        <dependency>
33            <groupId>org.projectlombok</groupId>
34            <artifactId>lombok</artifactId>
35            <optional>true</optional>
36        </dependency>
37
38        <!--spring boot 使用openoffice服务 maven依赖-->
39        <!--jodconverter 核心包 -->
40        <!--
  https://mvnrepository.com/artifact/org.jodconverter/jodconverter-core -->
41        <dependency>
42            <groupId>org.jodconverter</groupId>
43            <artifactId>jodconverter-core</artifactId>
44            <version>4.4.6</version>
45        </dependency>
46
47        <!--springboot支持包，里面包括了自动配置类 -->
48        <!--
  https://mvnrepository.com/artifact/org.jodconverter/jodconverter-spring-
  boot-starter -->
```

```

49     <dependency>
50         <groupId>org.jodconverter</groupId>
51         <artifactId>jodconverter-spring-boot-starter</artifactId>
52         <version>4.4.6</version>
53     </dependency>
54
55     <!--jodconverter 本地支持包 -->
56     <!--
57     https://mvnrepository.com/artifact/org.jodconverter/jodconverter-local -->
58     <dependency>
59         <groupId>org.jodconverter</groupId>
60         <artifactId>jodconverter-local</artifactId>
61         <version>4.4.6</version>
62     </dependency>
63 </dependencies>
64
65 <build>
66     <plugins>
67         <plugin>
68             <groupId>org.springframework.boot</groupId>
69             <artifactId>spring-boot-maven-plugin</artifactId>
70             <configuration>
71                 <image>
72                     <builder>paketobuildpacks/builder-jammy-
73 base:latest</builder>
74                 </image>
75             </configuration>
76         </plugin>
77     </plugins>
78 </build>
79 </project>

```

### 第三步：编写application.yml配置

```

1 jodconverter:
2   local:
3     enabled: true
4     # openoffice安装路径
5     office-home: 'C:\Program Files (x86)\OpenOffice 4'
6     max-tasks-per-process: 10
7     #openoffice端口号
8     port-numbers: 8100

```

## 第四步：启动服务

如果报以下错误：

```
1 "org.jodconverter.office.OfficeException: A process with acceptString
  'socket,host=127.0.0.1,port=8100,tcpNoDelay=1;urp;StarOffice.ServiceManager'
  started but its pid could not be found"
```

原因：

1. 项目启动目录中带有中文路径
2. soffice的端口异常了不能正常关闭
3. 操作系统环境问题
4. 依赖版本问题

启动成功的控制台日志如下：

```
1
2      .  _ _ _ _ _
3  ( ( ) \ _ _ | ' _ | ' _ | ' _ \ _ | \ \ \ \
4  \ \ _ _ ) | | ) | | | | | | | | ( | | ) ) ) )
5  ' | _ _ | . _ | | | | | | | | \ _ , | / / / /
6  ===== | _ | ===== | _ _ / _ / _ / _ /
7  :: Spring Boot ::                (v2.7.1)
8
9
10      _ _ _ _ _      _ _ _ _ _      _ _ _ _ _
11     / \      \      / \      \      / :: \      \
12     / :: \ _ _ \    / :: \      \    / :: \      \
13     / :: \ | | |    / :: \      \    / :: \      \
14     / :: \ | | |    / :: \      \    / :: \      \
15     / :: \ | | |    / :: \      \    / :: \      \
16     / :: \ | | |    / :: \      \    / :: \      \
17     / :: \ | | | _ _ _ _ _ / :: \      \ / :: \      \
18     / :: \ | | | :: \      \ / :: \      \ | | |    | | |
19 |
20 / :: \      | :: :: :: \ _ _ \ V :: \      \ :: \ _ _ \ | :: | _ _ |    | :: |
21 |
22 \ :: /      / ~~~~ / :: /      / \ :: /      \ :: /      \ / :: /      /
23 \ _ _ /      / :: /      / \ _ _ /      \ :: \ :: /      / \ :: /      /
24     / :: /      /      \ :: /      /      \ :: \      /      /
25     / :: /      /      \ :: /      /      \ :: \ :: /      /
26     / :: /      /      \ :: /      /      \ :: \ :: /      /
27     / :: /      /      \ :: /      /      \ :: \ _ _ /
28     \ :: /      /      \ :: /      /      \ ~~~
29     \ _ _ /      \ _ _ /
30 :: Github (https://github.com/maomao124) ::
31
```

```

32 2023-11-22 17:48:13.321 INFO 25232 --- [          main]
   m.o.OpenofficeWordToPdfApplication      : Starting
   OpenofficeWordToPdfApplication using Java 1.8.0_332 on mao with PID 25232
   (D:\程序\2023Q4\openoffice-word-to-pdf\target\classes started by mao in D:\程
   序\2023Q4\openoffice-word-to-pdf)
33 2023-11-22 17:48:13.322 INFO 25232 --- [          main]
   m.o.OpenofficeWordToPdfApplication      : No active profile set, falling
   back to 1 default profile: "default"
34 2023-11-22 17:48:13.759 INFO 25232 --- [          main]
   o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s):
   8080 (http)
35 2023-11-22 17:48:13.764 INFO 25232 --- [          main]
   o.apache.catalina.core.StandardService   : Starting service [Tomcat]
36 2023-11-22 17:48:13.764 INFO 25232 --- [          main]
   org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache
   Tomcat/9.0.64]
37 2023-11-22 17:48:13.808 INFO 25232 --- [          main] o.a.c.c.C.
   [Tomcat].[localhost].[/]                : Initializing Spring embedded
   webApplicationContext
38 2023-11-22 17:48:13.808 INFO 25232 --- [          main]
   w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
   initialization completed in 465 ms
39 2023-11-22 17:48:14.057 INFO 25232 --- [er-offprocmng-0]
   o.j.local.office.OfficeDescriptor         : soffice info (from exec path):
   Product: OpenOffice - Version: ??? - useLongOptionNameGnuStyle: false
40 2023-11-22 17:48:14.091 INFO 25232 --- [          main]
   o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080
   (http) with context path ''
41 2023-11-22 17:48:14.095 INFO 25232 --- [          main]
   m.o.OpenofficeWordToPdfApplication      : Started
   OpenofficeWordToPdfApplication in 1.005 seconds (JVM running for 1.461)
42 2023-11-22 17:48:14.157 WARN 25232 --- [er-offprocmng-0]
   o.j.l.office.LocalOfficeProcessManager   : Profile dir
   'C:\Users\mao\AppData\Local\Temp\.jodconverter_socket_host-127.0.0.1_port-
   8100_tcpNoDelay-1' already exists; deleting
43 2023-11-22 17:48:14.170 INFO 25232 --- [er-offprocmng-0]
   o.j.l.office.LocalOfficeProcessManager   : Starting process with --accept
   'socket,host=127.0.0.1,port=8100,tcpNoDelay=1;urp;StarOffice.ServiceManager'
   and profileDir 'C:\Users\mao\AppData\Local\Temp\.jodconverter_socket_host-
   127.0.0.1_port-8100_tcpNoDelay-1'
44 2023-11-22 17:48:14.809 INFO 25232 --- [er-offprocmng-0]
   o.j.local.office.OfficeConnection        : Connected:
   'socket,host=127.0.0.1,port=8100,tcpNoDelay=1'
45 2023-11-22 17:48:14.810 INFO 25232 --- [er-offprocmng-0]
   o.j.l.office.LocalOfficeProcessManager   : Started process; pid: 11904

```

## 第五步：编写单元测试

```
1 package mao.openofficewordtopdf;
2
3 import org.jodconverter.core.DocumentConverter;
4 import org.junit.jupiter.api.Test;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.boot.test.context.SpringBootTest;
7
8 import java.io.File;
9
10 @SpringBootTest
11 class OpenofficewordToPdfApplicationTests
12 {
13     @Autowired
14     private DocumentConverter documentConverter;
15
16     @Test
17     void contextLoads()
18     {
19         documentConverter.convert(new File("./test.docx")).to(new
20         File("./test.pdf"));
21     }
22 }
```

test.docx内容如下：

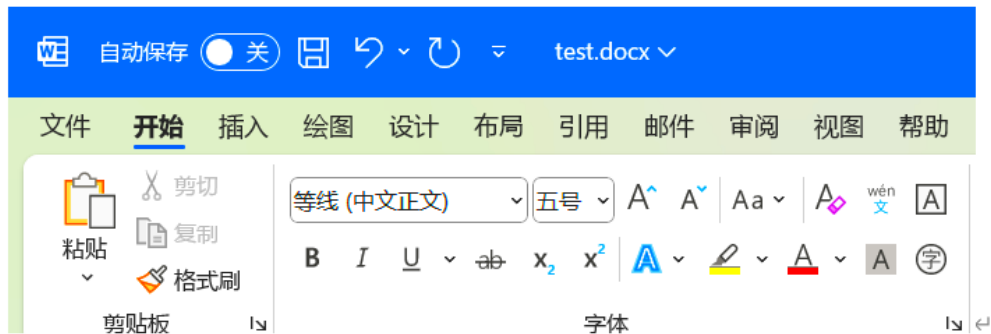
Wrewrwet↵

↵

Sfsfsgsdg↵

↵

↵



↵

↵

↵

↵

·Werwewe↵

↵

↵

中文↵

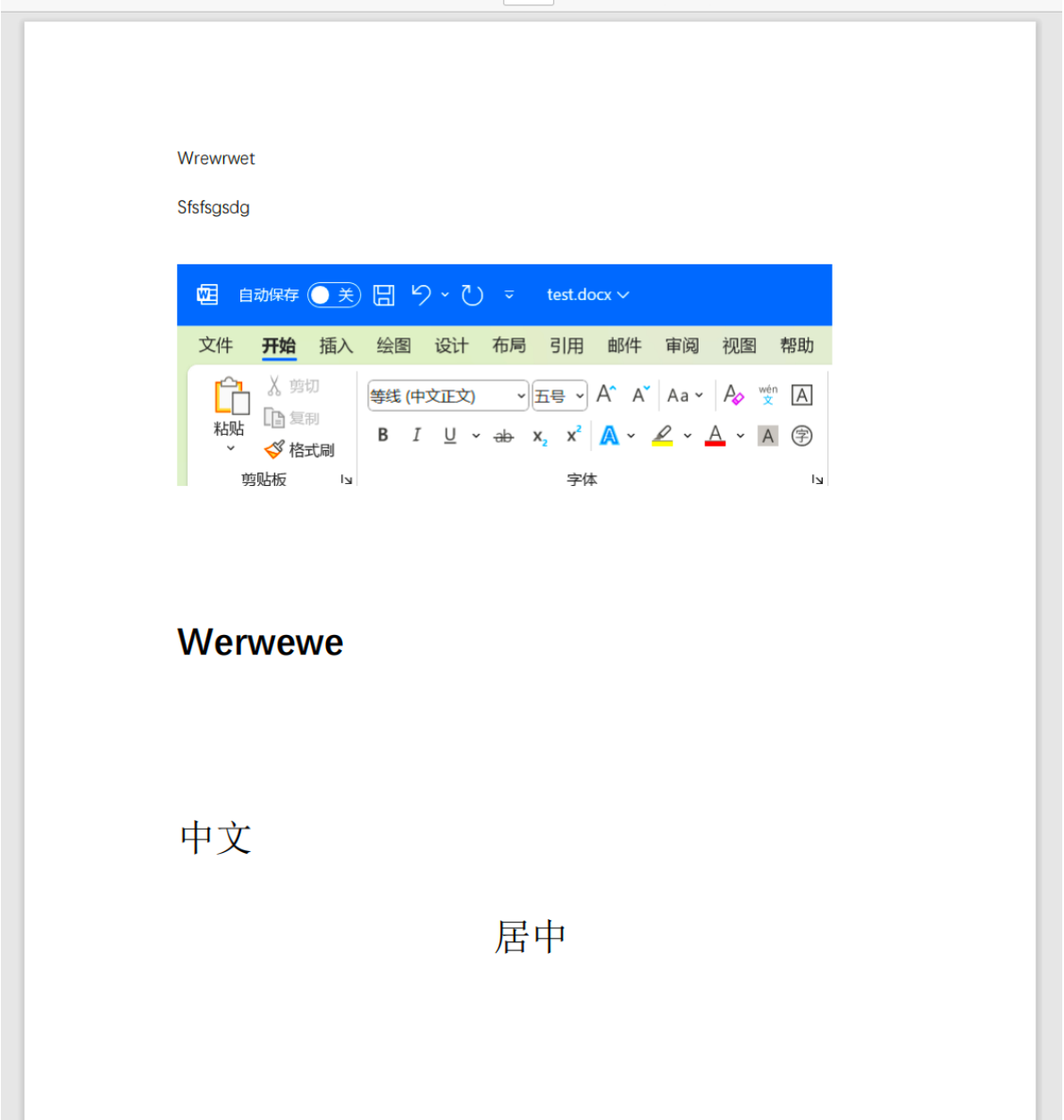
↵

居中↵

运行单元测试，日志输出如下：

```
1 2023-11-22 21:55:57.635 INFO 13336 --- [er-offprocmng-0]
   o.j.l.office.LocalOfficeProcessManager : Starting process with --accept
   'socket,host=127.0.0.1,port=8100,tcpNoDelay=1;urp;StarOffice.ServiceManager'
   and profileDir 'C:\Users\mao\AppData\Local\Temp\.jodconverter_socket_host-
   127.0.0.1_port-8100_tcpNoDelay-1'
2 2023-11-22 21:55:58.281 INFO 13336 --- [er-offprocmng-0]
   o.j.local.office.OfficeConnection      : Connected:
   'socket,host=127.0.0.1,port=8100,tcpNoDelay=1'
3 2023-11-22 21:55:58.281 INFO 13336 --- [er-offprocmng-0]
   o.j.l.office.LocalOfficeProcessManager : Started process; pid: 35016
4 2023-11-22 21:55:58.282 INFO 13336 --- [ter-poolentry-1]
   o.j.local.task.LocalConversionTask     : Executing local conversion task
   [docx -> pdf]...
```

输出的PDF文件如下：



文件流转换示例：

```

1 DocumentFormat documentFormatSrc =
  documentConverter.getFormatRegistry().getFormatByExtension("docx");
2     DocumentFormat documentFormatTarget =
  documentConverter.getFormatRegistry().getFormatByExtension("pdf");
3     documentConverter.convert(
4         new FileInputStream(new File("./test.docx")))
5         .as(documentFormatSrc)
6         .to(new FileOutputStream("./test.pdf"))
7         .as(documentFormatTarget).execute();

```

## 第六步：编写文档转换服务

```

1 package mao.openofficewordtopdf.service;
2
3 import java.io.File;
4 import java.io.InputStream;
5 import java.io.OutputStream;
6
7 /**
8  * Project name(项目名称): openoffice-word-to-pdf
9  * Package(包名): mao.openofficewordtopdf.service
10  * Interface(接口名): DocumentConverterService
11  * Author(作者): mao
12  * Author QQ: 1296193245
13  * GitHub: https://github.com/maomao124/
14  * Date(创建日期): 2023/11/23
15  * Time(创建时间): 8:49
16  * Version(版本): 1.0
17  * Description(描述): 文档转换服务
18  */
19
20 public interface DocumentConverterService
21 {
22
23     /**
24      * 文档转换
25      *
26      * @param sourcePath 来源路径字符串
27      * @param targetPath 目标路径字符串
28      */
29     void converter(String sourcePath, String targetPath);
30
31     /**
32      * 文档转换
33      *
34      * @param sourcePath 来源路径
35      * @param targetPath 目标路径
36      */
37     void converter(File sourcePath, File targetPath);

```



```

38
39     /**
40      * 文档转换
41      *
42      * @param inputStream      输入流
43      * @param outputStream      输出流
44      * @param sourceFileSuffix 源文件后缀
45      * @param targetFileSuffix 目标文件后缀
46      */
47     void converter(InputStream inputStream, OutputStream outputStream,
48                     String sourceFileSuffix, String targetFileSuffix);
49 }

```

## 第七步：编写服务实现类

```

1  package mao.openofficewordtopdf.service.impl;
2
3  import lombok.SneakyThrows;
4  import lombok.extern.slf4j.Slf4j;
5  import mao.openofficewordtopdf.service.DocumentConverterService;
6  import org.jodconverter.core.DocumentConverter;
7  import org.jodconverter.core.document.DocumentFormat;
8  import org.springframework.beans.factory.annotation.Autowired;
9  import org.springframework.stereotype.Service;
10
11 import java.io.File;
12 import java.io.InputStream;
13 import java.io.OutputStream;
14
15 /**
16  * Project name(项目名称): openoffice-word-to-pdf
17  * Package(包名): mao.openofficewordtopdf.service.impl
18  * Class(类名): DocumentConverterServiceImpl
19  * Author(作者): mao
20  * Author QQ: 1296193245
21  * GitHub: https://github.com/maomao124/
22  * Date(创建日期): 2023/11/23
23  * Time(创建时间): 8:50
24  * Version(版本): 1.0
25  * Description(描述): 文档转换服务实现
26  */
27
28 @Slf4j
29 @Service
30 public class DocumentConverterServiceImpl implements
31     DocumentConverterService
32 {
33     @Autowired
34     private DocumentConverter documentConverter;
35

```

```

36     @SneakyThrows
37     @Override
38     public void converter(String sourcePath, String targetPath)
39     {
40         documentConverter.convert(new File(sourcePath)).to(new
File(targetPath)).execute();
41     }
42
43     @SneakyThrows
44     @Override
45     public void converter(File sourcePath, File targetPath)
46     {
47         documentConverter.convert(sourcePath).to(targetPath).execute();
48     }
49
50     @SneakyThrows
51     @Override
52     public void converter(InputStream inputStream, OutputStream
outputStream, String sourceFileSuffix, String targetFileSuffix)
53     {
54         DocumentFormat sourceDocumentFormat =
documentConverter.getFormatRegistry()
55             .getFormatByExtension(sourceFileSuffix);
56         DocumentFormat targetDocumentFormat =
documentConverter.getFormatRegistry()
57             .getFormatByExtension(targetFileSuffix);
58         documentConverter
59             .convert(inputStream)
60             .as(sourceDocumentFormat)
61             .to(outputStream)
62             .as(targetDocumentFormat)
63             .execute();
64     }
65 }

```

该实现类的单元测试如下：

```

1  package mao.openofficewordtopdf.service.impl;
2
3  import lombok.SneakyThrows;
4  import org.junit.jupiter.api.Test;
5  import org.springframework.beans.factory.annotation.Autowired;
6  import org.springframework.boot.test.context.SpringBootTest;
7
8  import java.io.File;
9  import java.io.FileInputStream;
10 import java.io.FileOutputStream;
11
12 import static org.junit.jupiter.api.Assertions.*;
13
14 /**
15  * Project name(项目名称): openoffice-word-to-pdf

```

```

16  * Package(包名): mao.openofficewordtopdf.service.impl
17  * Class(测试类名): DocumentConverterServiceImplTest
18  * Author(作者): mao
19  * Author QQ: 1296193245
20  * GitHub: https://github.com/maomao124/
21  * Date(创建日期): 2023/11/23
22  * Time(创建时间): 9:03
23  * Version(版本): 1.0
24  * Description(描述): 测试类
25  */
26
27  @SpringBootTest
28  class DocumentConverterServiceImplTest
29  {
30      @Autowired
31      private DocumentConverterServiceImpl documentConverterService;
32
33      @Test
34      void converter()
35      {
36          documentConverterService.converter("./test.docx", "./test4.html");
37      }
38
39      @Test
40      void testConverter()
41      {
42          documentConverterService.converter(new File("./test.docx"), new
File("./test5.pdf"));
43      }
44
45      @SneakyThrows
46      @Test
47      void testConverter1()
48      {
49          documentConverterService.converter(new FileInputStream("./test.ppt")
, new FileOutputStream("./test6.pdf")
50              , "ppt", "pdf");
51      }
52  }
53  }

```

经测试, 一切正常

## 第八步: 编写接口

```

1  package mao.openofficewordtopdf.controller;
2
3  import lombok.SneakyThrows;
4  import lombok.extern.slf4j.Slf4j;
5  import mao.openofficewordtopdf.service.DocumentConverterService;
6  import org.springframework.beans.factory.annotation.Autowired;

```

```

7  import org.springframework.stereotype.Controller;
8  import org.springframework.web.bind.annotation.PathVariable;
9  import org.springframework.web.bind.annotation.PostMapping;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RestController;
12 import org.springframework.web.multipart.MultipartFile;
13
14 import javax.servlet.http.HttpServletResponse;
15
16 /**
17  * Project name(项目名称): openoffice-word-to-pdf
18  * Package(包名): mao.openofficewordtopdf.controller
19  * Class(类名): DocumentConverterController
20  * Author(作者): mao
21  * Author QQ: 1296193245
22  * GitHub: https://github.com/maomao124/
23  * Date(创建日期): 2023/11/23
24  * Time(创建时间): 9:13
25  * Version(版本): 1.0
26  * Description(描述): 无
27  */
28
29 @Slf4j
30 @Controller
31 @RequestMapping("/api")
32 public class DocumentConverterController
33 {
34
35     @Autowired
36     private DocumentConverterService documentConverterService;
37
38     /**
39      * 文档转换接口
40      *
41      * @param httpResponse HttpServletResponse对象
42      * @param sourceFileSuffix 源文件后缀
43      * @param targetFileSuffix 目标文件后缀
44      */
45     @SneakyThrows
46     @PostMapping("/converter/{sourceFileSuffix}/{targetFileSuffix}")
47     public void converter(HttpServletResponse httpResponse,
48                          @PathVariable String sourceFileSuffix,
49                          @PathVariable String targetFileSuffix,
50                          MultipartFile file)
51     {
52         String originalFilename = file.getOriginalFilename();
53         log.info(originalFilename);
54         String[] split = originalFilename.split("\\.");
55         //类型有问题
56         httpResponse.setContentType("application/" +
targetFileSuffix);
57         httpResponse.setHeader("Content-disposition",
58                                "attachment;filename=" + new String((split[0] + "." +
targetFileSuffix)
59                                .getBytes("utf-8"), "iso-8859-1"));

```

```

60         documentConverterService.converter(file.getInputStream(),
61             httpServletResponse.getOutputStream(),
62             sourceFileSuffix,
63             targetFileSuffix);
64     }
65 }
66

```

## 第九步：编写简单前端页面

```

1  <!DOCTYPE html>
2
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <title>文档转换服务</title>
7  </head>
8  <body>
9      <form id="from" enctype="multipart/form-data" onsubmit="return check()"
10         method="post" action="/api/converter/docx/pdf">
11          <input type="file" name="file">
12          <input type="submit" name="提交">
13      </form>
14
15      <br>
16      <input id="p1" type="text" placeholder="输入的文件后缀名">
17      <br>
18      <input id="p2" type="text" placeholder="输出的文件后缀名">
19
20      <script>
21          function check()
22          {
23              var from = document.getElementById("from")
24              var p1 = document.getElementById("p1").value
25              var p2 = document.getElementById("p2").value
26              from.action = '/api/converter/' + p1 + '/' + p2;
27              return true;
28          }
29      </script>
30  </body>
31 </html>

```

## 第十步：启动项目测试

```
1 2023-11-24 23:25:43.069 INFO 38168 --- [          main]
   m.o.OpenofficewordToPdfApplication      : Starting
   OpenofficewordToPdfApplication using Java 1.8.0_332 on mao with PID 38168
   (D:\程序\2023Q4\openoffice-word-to-pdf\target\classes started by mao in D:\程
   序\2023Q4\openoffice-word-to-pdf)
2 2023-11-24 23:25:43.071 INFO 38168 --- [          main]
   m.o.OpenofficewordToPdfApplication      : No active profile set, falling
   back to 1 default profile: "default"
3 2023-11-24 23:25:43.441 INFO 38168 --- [          main]
   o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port(s):
   8090 (http)
4 2023-11-24 23:25:43.444 INFO 38168 --- [          main]
   o.apache.catalina.core.StandardService  : Starting service [Tomcat]
5 2023-11-24 23:25:43.445 INFO 38168 --- [          main]
   org.apache.catalina.core.StandardEngine  : Starting Servlet engine: [Apache
   Tomcat/9.0.64]
6 2023-11-24 23:25:43.481 INFO 38168 --- [          main] o.a.c.c.C.
   [Tomcat].[localhost].[/]               : Initializing Spring embedded
   WebApplicationContext
7 2023-11-24 23:25:43.481 INFO 38168 --- [          main]
   w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
   initialization completed in 391 ms
8 2023-11-24 23:25:43.639 INFO 38168 --- [er-offproc-mng-0]
   o.j.local.office.OfficeDescriptor        : soffice info (from exec path):
   Product: OpenOffice - Version: ??? - useLongOptionNameGnuStyle: false
9 2023-11-24 23:25:43.725 INFO 38168 --- [          main]
   o.s.b.a.w.s.welcomePageHandlerMapping    : Adding welcome page: class path
   resource [static/index.html]
10 2023-11-24 23:25:43.777 INFO 38168 --- [er-offproc-mng-0]
   o.j.l.office.LocalOfficeProcessManager   : Starting process with --accept
   'socket,host=127.0.0.1,port=8100,tcpNoDelay=1;urp;StarOffice.ServiceManager'
   and profileDir 'C:\Users\mao\AppData\Local\Temp\.jodconverter_socket_host-
   127.0.0.1_port-8100_tcpNoDelay-1'
11 2023-11-24 23:25:43.785 INFO 38168 --- [          main]
   o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8090
   (http) with context path ''
12 2023-11-24 23:25:43.789 INFO 38168 --- [          main]
   m.o.OpenofficewordToPdfApplication      : Started
   OpenofficewordToPdfApplication in 0.889 seconds (JVM running for 1.419)
13 2023-11-24 23:25:44.456 INFO 38168 --- [er-offproc-mng-0]
   o.j.local.office.OfficeConnection        : Connected:
   'socket,host=127.0.0.1,port=8100,tcpNoDelay=1'
14 2023-11-24 23:25:44.457 INFO 38168 --- [er-offproc-mng-0]
   o.j.l.office.LocalOfficeProcessManager   : Started process; pid: 10832
```

访问: <http://localhost:8090/>

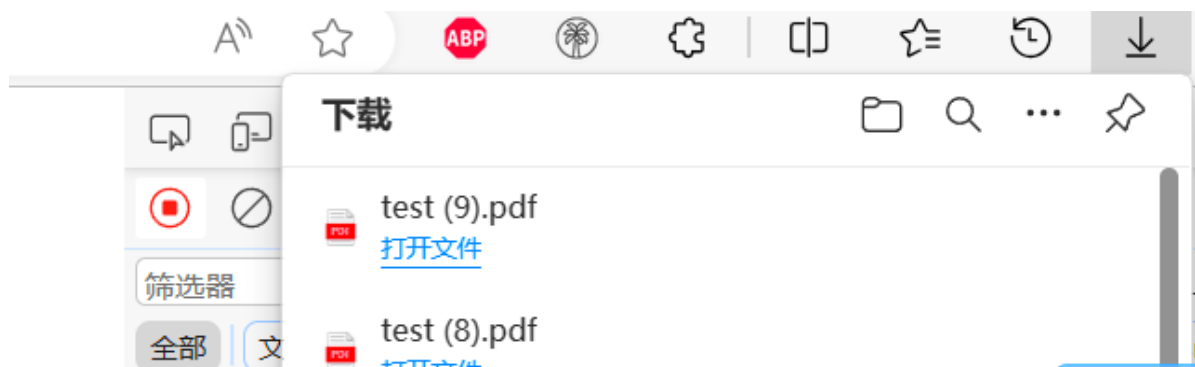
选择文件名测试

选择文件 test.doc

提交

doc

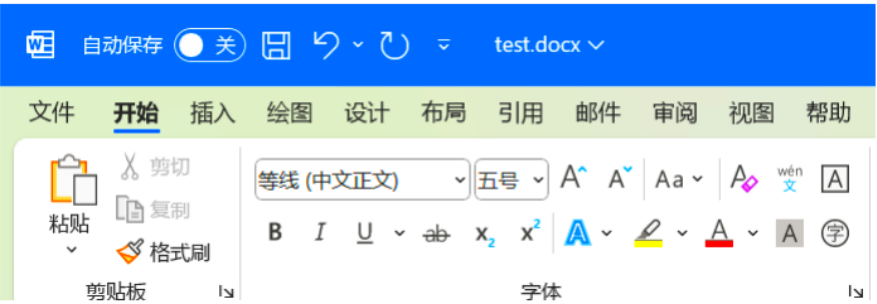
pdf



转换无问题

Wrewrwet

Sfsfsgsdg



Werwewe

中文

居中

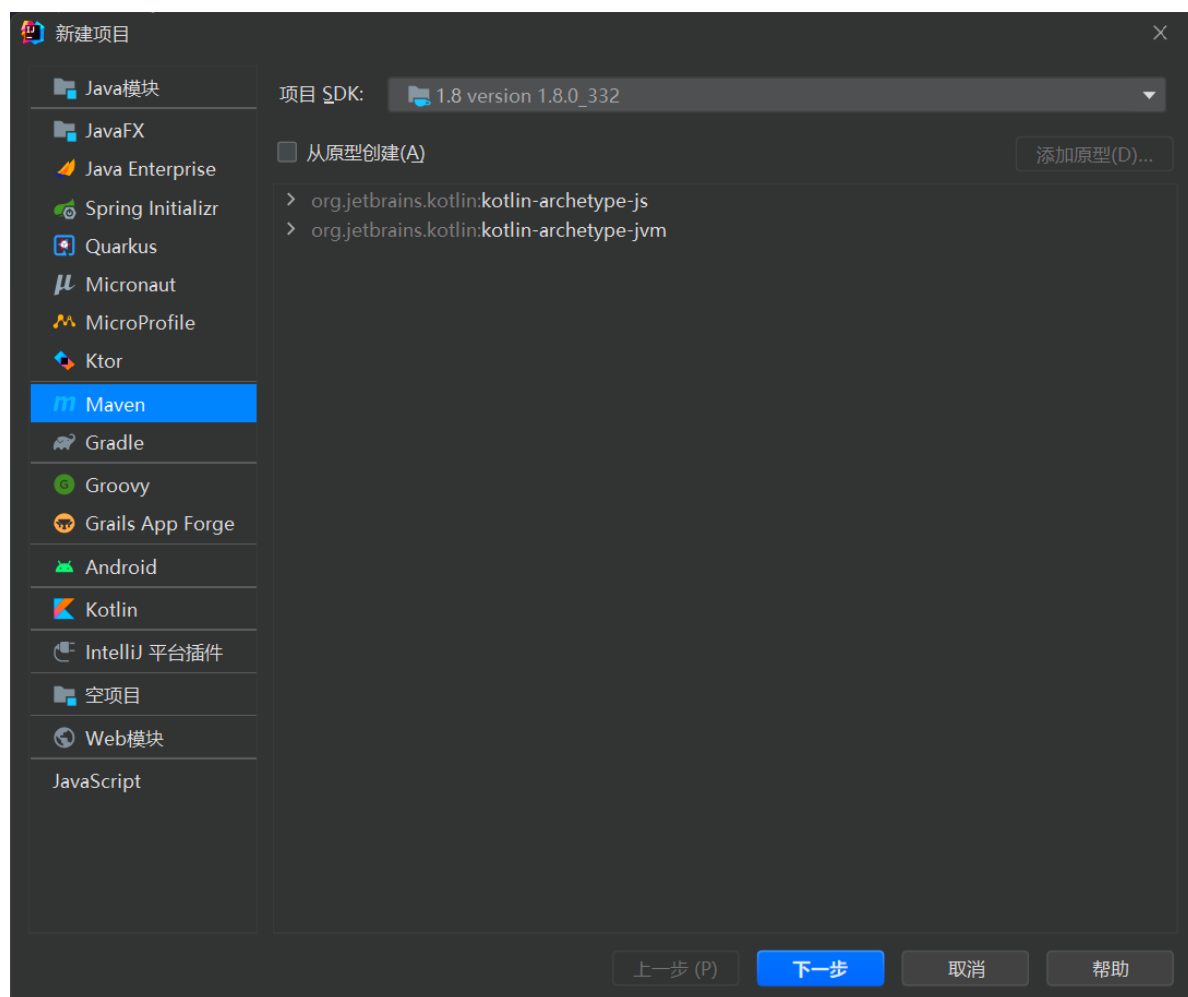


# 概述

并不是所有的项目都是springboot项目，下面将不使用springboot项目来调用openoffice服务

## docx转pdf

### 第一步：新建maven项目



这里项目名称为 `openoffice-word-to-pdf2`

### 第二步：配置maven

依赖如下：

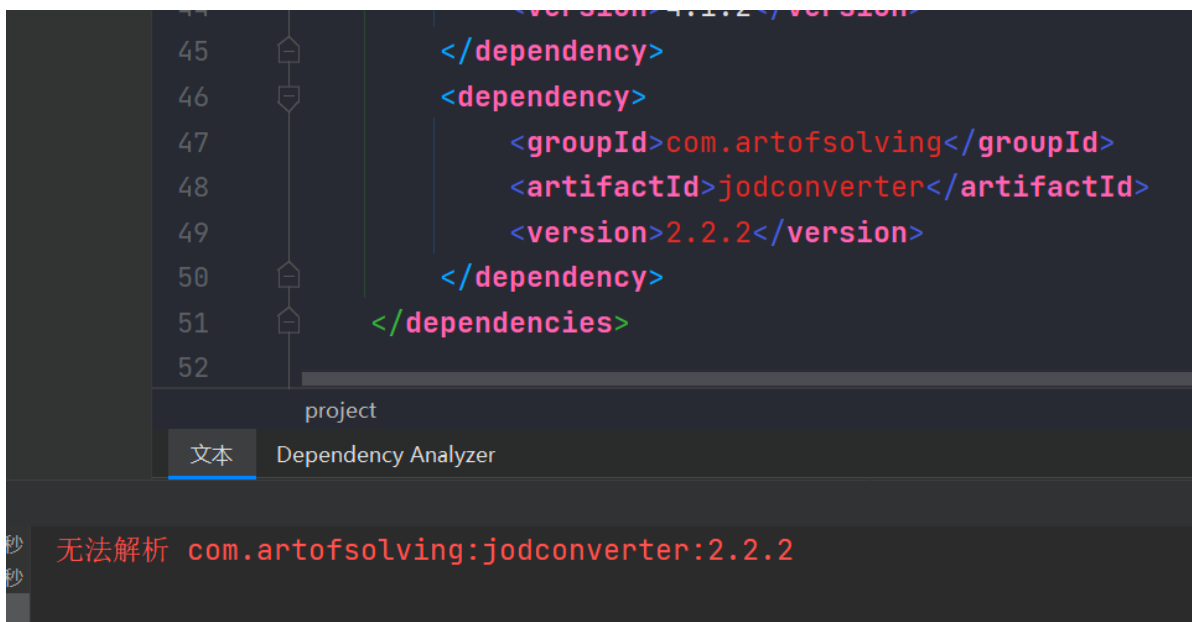
```
1 <dependencies>
2     <dependency>
```

```

3         <groupId>org.openoffice</groupId>
4         <artifactId>juh</artifactId>
5         <version>4.1.2</version>
6     </dependency>
7     <dependency>
8         <groupId>org.openoffice</groupId>
9         <artifactId>jurt</artifactId>
10        <version>4.1.2</version>
11    </dependency>
12    <dependency>
13        <groupId>org.openoffice</groupId>
14        <artifactId>ridl</artifactId>
15        <version>4.1.2</version>
16    </dependency>
17    <dependency>
18        <groupId>org.openoffice</groupId>
19        <artifactId>unoil</artifactId>
20        <version>4.1.2</version>
21    </dependency>
22    <dependency>
23        <groupId>com.artofsolving</groupId>
24        <artifactId>jodconverter</artifactId>
25        <version>2.2.2</version>
26    </dependency>
27 </dependencies>

```

jodconverter需要单独下载，因为中央仓库里没有



下载地址: <https://sourceforge.net/projects/jodconverter/files/>

SummaryFilesReviewsSupportCode

Download Latest Version  
jodconverter-2.2.2.zip (2.1 MB)

Get Updates

Home

Name	Modified	Size	Downloads / Week
JODConverter	2009-04-11		39
Totals: 1 Item			39

Download Latest Version  
jodconverter-2.2.2.zip (2.1 MB)

Get Updates

Home / JODConverter

Name	Modified	Size	Downloads / Week
Parent folder			
2.2.2	2009-04-11		39
2.2.1	2007-12-19		0
2.2.0	2007-06-03		0
2.1.1	2007-02-13		0
2.1.0	2006-11-20		0
2.1rc2	2006-09-07		0
2.1rc1	2006-07-27		0
2.0.0	2006-07-13		0
2.0rc2	2005-10-30		0
1.0b4	2004-07-03		0
Totals: 10 Items			39

点击2.2.2版本

Download Latest Version  
jodconverter-2.2.2.zip (2.1 MB)

Get Updates

Home / JODConverter / 2.2.2

Name	Modified	Size	Downloads / Week
Parent folder			
jodconverter-tomcat-2.2.2.zip	2009-04-11	8.3 MB	3
jodconverter-webapp-2.2.2.zip	2009-04-11	3.3 MB	7
jodconverter-2.2.2.zip	2009-04-11	2.1 MB	29
Totals: 3 Items		13.7 MB	39

下载: [jodconverter-2.2.2.zip](#)

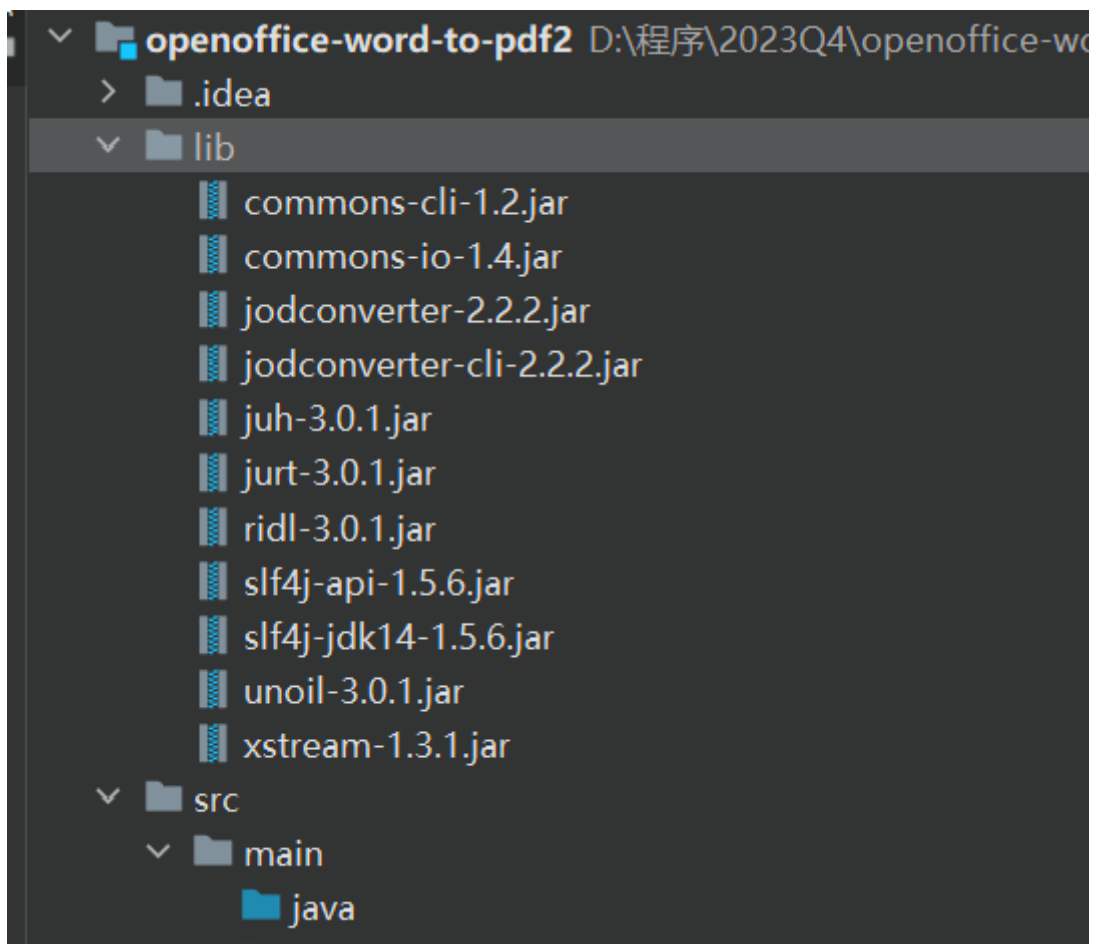
下载完成后，解压，可以得到这样的目录：

名称	修改日期	类型	大小
docs	2009/4/11 20:40	文件夹	
lib	2009/4/11 20:38	文件夹	
src	2009/4/11 20:39	文件夹	
ChangeLog.txt	2009/4/11 20:39	文本文档	4 KB
document-formats.xml	2009/4/11 20:31	XML 文件	14 KB
LICENSE.txt	2007/12/20 1:44	文本文档	26 KB
README.txt	2009/4/11 20:39	文本文档	2 KB

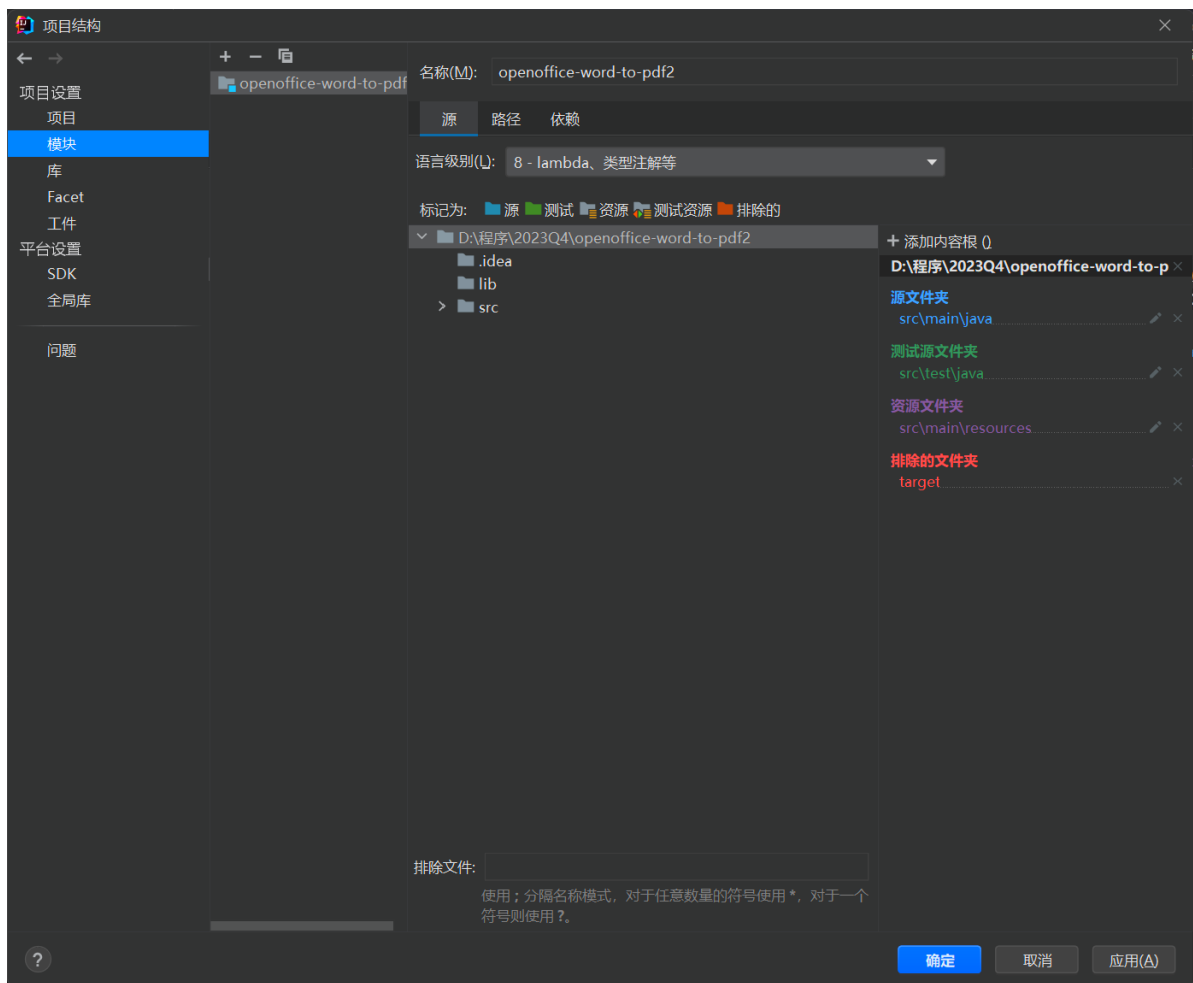
在lib目录下运行：

```
1 mvn install:install-file -Dfile="jodconverter-2.2.2.jar" -
  DgroupId=com.artofsolving -DartifactId=jodconverter -Dversion=2.2.2 -
  Dpackaging=jar
```

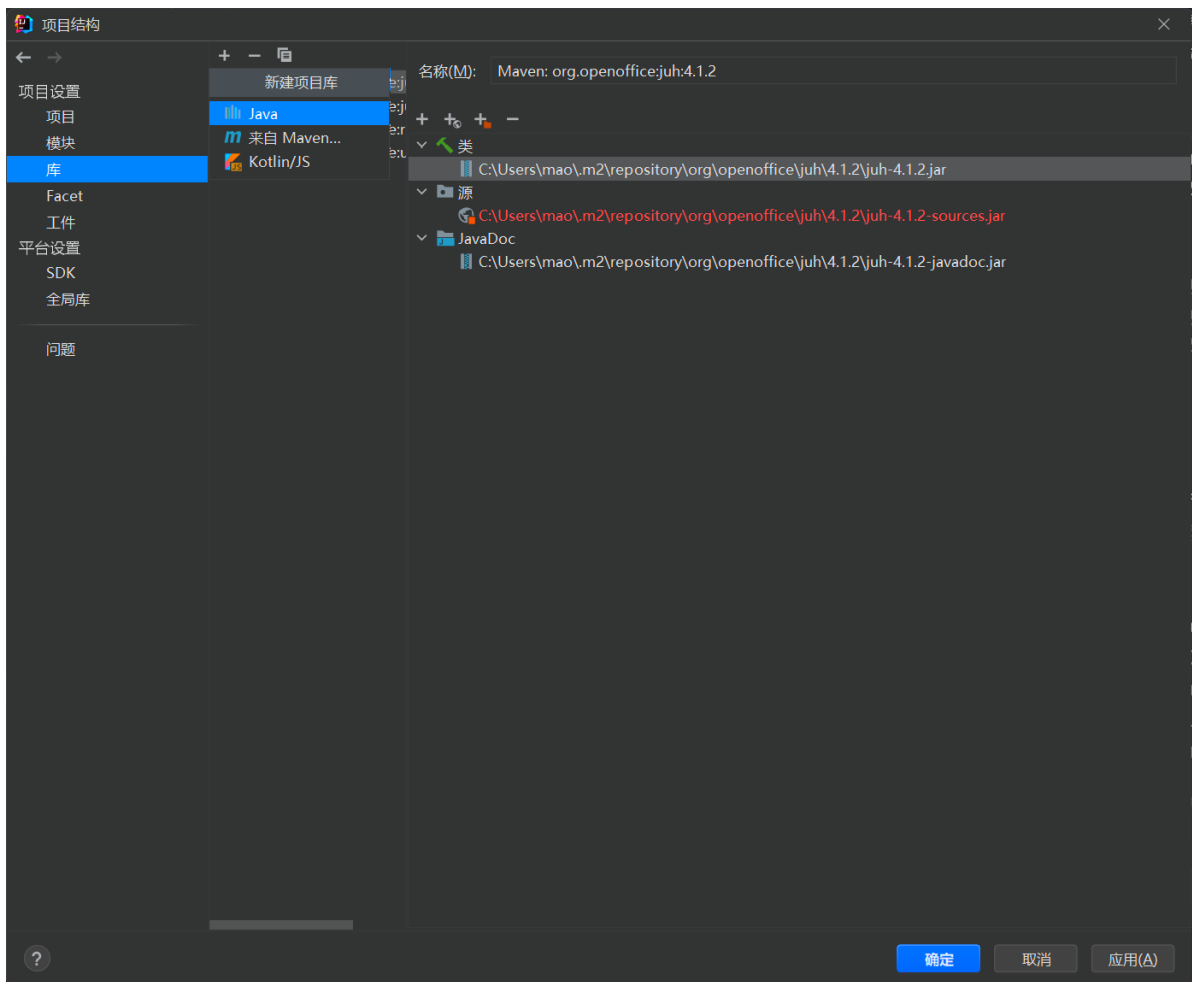
或者将lib目录拷贝到项目目录下



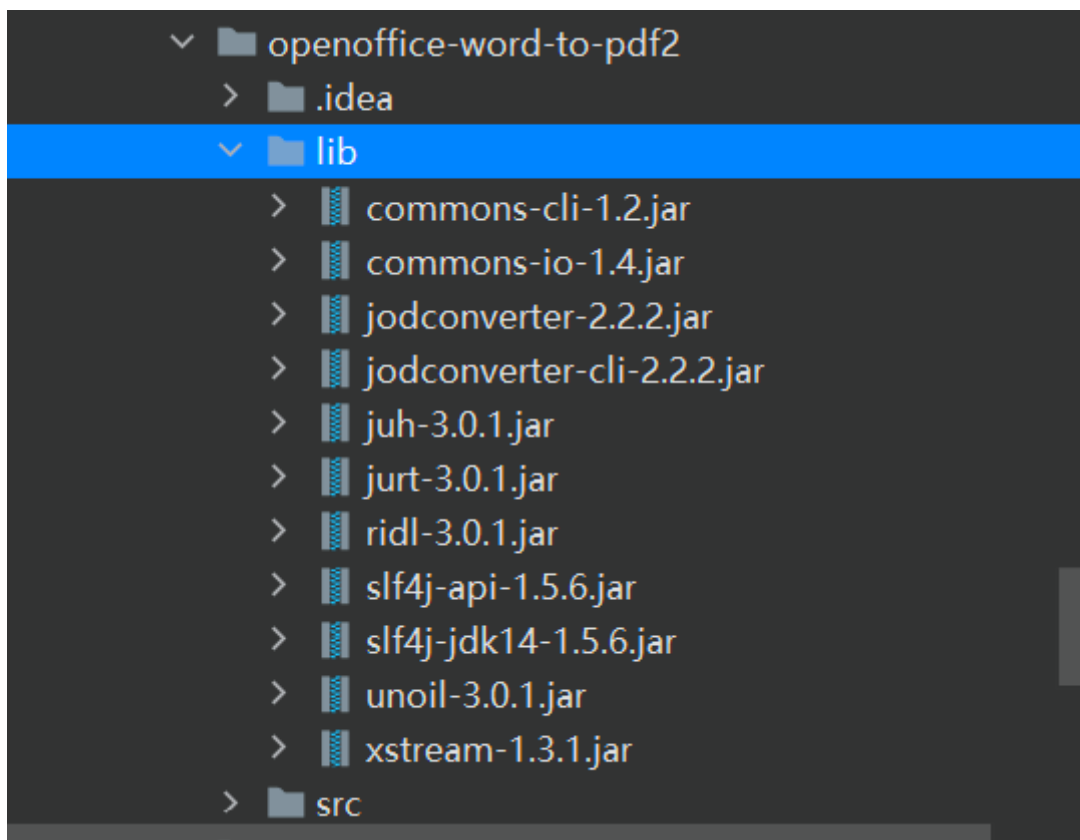
然后再打开项目结构



选择库



点击新建，选择lib包目录





### 第三步：编写DocumentConverterUtil

```
1 package mao;
2
3 import com.artofsolving.jodconverter.DefaultDocumentFormatRegistry;
4 import com.artofsolving.jodconverter.DocumentConverter;
5 import com.artofsolving.jodconverter.DocumentFormat;
6 import
  com.artofsolving.jodconverter.openoffice.connection.SocketOpenOfficeConnect
  ion;
7 import
  com.artofsolving.jodconverter.openoffice.converter.StreamOpenOfficeDocument
  Converter;
8
9 import java.io.File;
10 import java.io.IOException;
11 import java.io.InputStream;
12 import java.io.OutputStream;
13 import java.net.ConnectException;
14 import java.util.Properties;
15
16 /**
17  * Project name(项目名称): openoffice-word-to-pdf2
18  * Package(包名): mao
19  * Class(类名): DocumentConverterUtil
20  * Author(作者): mao
21  * Author QQ: 1296193245
22  * GitHub: https://github.com/maomao124/
23  * Date(创建日期): 2023/11/24
24  * Time(创建时间): 23:53
25  * Version(版本): 1.0
26  * Description(描述): 文档转换工具
27  */
28
29 public class DocumentConverterUtil
30 {
31     private static SocketOpenOfficeConnection connection = null;
32
33     private static DocumentConverter documentConverter = null;
34
35     private static Properties properties = null;
36 }
```

```

37     private static int serviceCheckCycle;
38
39     static
40     {
41         InputStream inputStream =
DocumentConverterUtil.class.getClassLoader().getResourceAsStream("conf/open
office.properties");
42         try
43         {
44             properties = new Properties();
45             properties.load(inputStream);
46             String host = properties.getProperty("server.host");
47             Integer port =
Integer.parseInt(properties.getProperty("server.port"));
48             serviceCheckCycle =
Integer.parseInt(properties.getProperty("serviceCheckCycle"));
49             connection = new SocketOpenOfficeConnection(host, port);
50             connection.connect();
51             documentConverter = new
StreamOpenOfficeDocumentConverter(connection);
52         }
53         catch (ConnectException e)
54         {
55             e.printStackTrace();
56             System.out.println("文档转换服务连接失败");
57             //throw new RuntimeException(e);
58         }
59         catch (IOException e)
60         {
61             e.printStackTrace();
62             System.out.println("加载配置文件失败或者其它问题");
63             //throw new RuntimeException(e);
64         }
65         finally
66         {
67             //服务检查
68             new Thread(new Runnable()
69             {
70                 @Override
71                 public void run()
72                 {
73                     while (true)
74                     {
75                         try
76                         {
77                             Thread.sleep(serviceCheckCycle);
78                             if (isNotConnection())
79                             {
80                                 System.out.println("openoffice服务异常，正在
尝试重连");
81                                 retry();
82                             }
83                         }
84                         catch (InterruptedException e)
85                     {

```



```

86         e.printStackTrace();
87     }
88 }
89 }
90 }).start();
91
92 Runtime.getRuntime().addShutdownHook(new Thread(() ->
93 {
94     if (connection.isConnected())
95     {
96         System.out.println("正在关闭openoffice连接...");
97         connection.disconnect();
98     }
99 }));
100 }
101 }
102
103 /**
104  * 服务重连
105  *
106  * @return 重连结果
107  */
108 public static boolean retry()
109 {
110     System.out.println("尝试重连openoffice");
111     String host = properties.getProperty("server.host");
112     Integer port =
113 Integer.parseInt(properties.getProperty("server.port"));
114     connection = new SocketOpenOfficeConnection(host, port);
115     documentConverter = new
116 StreamOpenOfficeDocumentConverter(connection);
117     try
118     {
119         connection.connect();
120         if (isConnection())
121         {
122             return true;
123         }
124         return false;
125     }
126     catch (Exception e)
127     {
128         System.out.println("重连失败");
129         e.printStackTrace();
130         return false;
131     }
132 }
133
134 /**
135  * 得到当前properties
136  *
137  * @return properties对象
138  */
139 public static Properties getProperties()
140 {

```

```

139         return properties;
140     }
141
142     /**
143      * 判断服务是否正常连接
144      *
145      * @return 成功连接为true
146      */
147     public static boolean isConnection()
148     {
149         if (connection == null || !connection.isConnected())
150         {
151             return false;
152         }
153         return true;
154     }
155
156     /**
157      * 判断服务是否没有连接
158      *
159      * @return 成功连接为false
160      */
161     public static boolean isNotConnection()
162     {
163         return !isConnection();
164     }
165
166
167     /**
168      * 文档转换
169      *
170      * @param sourcePath 源路径
171      * @param targetPath 目标路径
172      */
173     public static void converter(String sourcePath, String targetPath)
174     {
175         documentConverter.convert(new File(sourcePath), new
176         File(targetPath));
177     }
178
179     /**
180      * 文档转换
181      *
182      * @param sourcePath 源路径
183      * @param targetPath 目标路径
184      */
185     public static void converter(File sourcePath, File targetPath)
186     {
187         documentConverter.convert(sourcePath, targetPath);
188     }
189
190     /**
191      * 文档转换
192      *
193      * @param inputStream 输入流

```

```

193     * @param outputStream    输出流
194     * @param sourceFileSuffix 源文件后缀
195     * @param targetFileSuffix 目标文件后缀
196     */
197     public static void converter(InputStream inputStream, OutputStream
outputStream,
198                                   String sourceFileSuffix, String
targetFileSuffix)
199     {
200         DocumentFormat sourceDocumentFormat = new
DefaultDocumentFormatRegistry()
201             .getFormatByFileExtension(sourceFileSuffix);
202         DocumentFormat targetDocumentFormat = new
DefaultDocumentFormatRegistry()
203             .getFormatByFileExtension(targetFileSuffix);
204         documentConverter.convert(inputStream, sourceDocumentFormat,
outputStream, targetDocumentFormat);
205     }
206 }

```

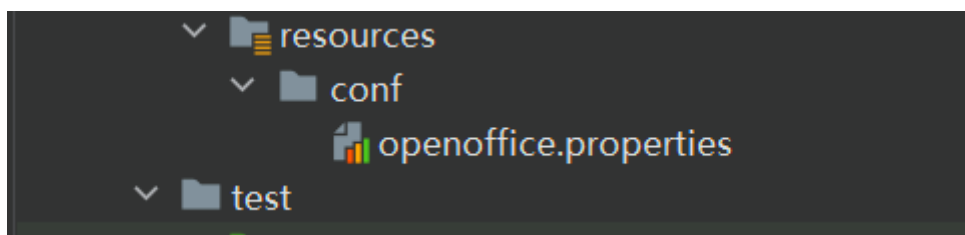
## 第四步：编写openoffice.properties

```

1  # openoffice服务所在地址
2  server.host=127.0.0.1
3  # openoffice服务端口号
4  server.port=8100
5  # 服务连接检查周期，单位是毫秒，600000为10分钟。如果服务连接挂了，将会尝试重新连接
6  serviceCheckCycle=600000

```

配置文件位置：



## 第五步：编写测试类

```

1  package mao;
2
3  import java.io.FileInputStream;
4  import java.io.FileNotFoundException;
5  import java.io.FileOutputStream;
6

```

```

7  /**
8   * Project name(项目名称): openoffice-word-to-pdf2
9   * Package(包名): mao
10  * Class(类名): Test
11  * Author(作者): mao
12  * Author QQ: 1296193245
13  * GitHub: https://github.com/maomao124/
14  * Date(创建日期): 2023/11/25
15  * Time(创建时间): 0:05
16  * Version(版本): 1.0
17  * Description(描述): 无
18  */
19
20 public class Test
21 {
22     public static void main(String[] args) throws FileNotFoundException
23     {
24         DocumentConverterUtil.converter("./test.docx", "./test.pdf");
25         DocumentConverterUtil.converter("./test.docx", "./test2.pdf");
26         DocumentConverterUtil.converter("./test.docx", "./test3.pdf");
27         DocumentConverterUtil.converter(new FileInputStream("./test.docx"),
28             new FileOutputStream("./test4.pdf"), "docx", "pdf");
29         System.out.println("转换完成");
30     }
31 }

```

## 第六步：运行查看结果

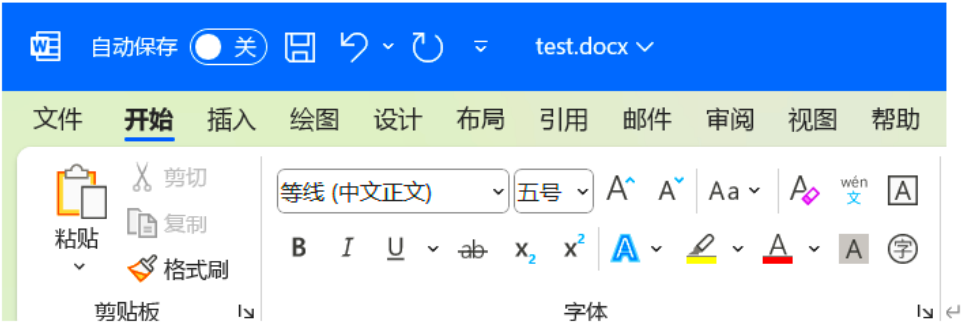
```

1  十一月 25, 2023 12:17:56 上午
   com.artofsolving.jodconverter.openoffice.connection.AbstractOpenOfficeConnect
   ion connect
2  信息: connected
3  转换完成

```

源文档内容：

Wrewrwet  
Sfsfsgsdg



←  
←  
←  
←

·Werwewe

←  
←

中文

←

居中

12

5644

131314

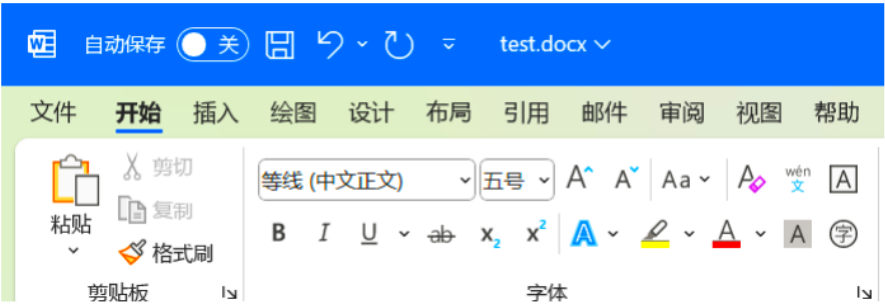
1	2	3
4	5	6
7	8	9

←

转换后：

Wrewrwet

Sfsfsgsdg



Werwewe

中文

居中  
12  
5644  
131314

1	2	3
4	5	6
7	8	9

结果正确，性能在1秒内

---

end