



链滴

基于 aspose.word 的 Java 文档转换项目

作者: [liumapp](#)

原文链接: <https://ld246.com/article/1555310416327>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

抽象工厂模式设计的一款工具项目，方便快速实现html转PDF、doc转PDF等功能

项目源代码: [github/simple-convert](https://github.com/liumapp/simple-convert)

如何使用

- 使用Maven下载依赖

```
<dependency>
  <groupId>com.liumapp.simple.convert</groupId>
  <artifactId>simple-convert</artifactId>
  <version>v1.0.0</version>
</dependency>
```

- 拷贝repo目录到本地项目中，否则会提示找不到aspose的jar包

当然，您也可以自行将libs目录下的aspose这个jar包部署到自己的nexus私服或者导入maven的本地库中

- html转doc

- 通过文件目录转换

```
BasicConverter converter = new HtmlToPdfConverterFactory().getInstance();
String htmlFilePath = HtmlToPdfConverterTest.class.getClassLoader().getResource("test.htm")
).getPath();
String pdfResultPath = "./result.pdf";
converter.convertByFilePath(htmlFilePath, pdfResultPath);
```

执行后将htmlFilePath所指向的html文件转换为pdf文件，并保存在pdfResultPath路径下

- 通过输入流转换

```
BasicConverter converter = new HtmlToPdfConverterFactory().getInstance();
String targetFilePath = HtmlToPdfConverterTest.class.getClassLoader().getResource("test.html")
).getPath();
InputStream is = new FileInputStream(targetFilePath);
OutputStream os = new FileOutputStream(new File("./result2.pdf"));
converter.convertByStream(is, os);
os.flush();
is.close();
os.close();
```

将要转换的html文件作为输入流输出，输出流为要存储的pdf文件输出流，也可以使用ByteArrayOutputStream暂存在内存中

- 通过base64转换

```
BasicConverter converter = new HtmlToPdfConverterFactory().getInstance();
String targetFilePath = HtmlToPdfConverterTest.class.getClassLoader().getResource("test.html")
).getPath();
InputStream is = new FileInputStream(targetFilePath);
String inputBase64 = Base64FileTool.inputStreamToBase64(is);
String resultBase64 = converter.convertByBase64(inputBase64);
is.close();
```

inputBase64为html文件内容的base64值，输出的resultBase64为转换后的pdf base64值

- 直接传入html字符串转pdf文件

同通过base64转换一样，将html字符串加密为base64值，将转换后的base64解密存储即可得到pdf文件

```
BasicConverter converter = new HtmlToPdfConverterFactory().getInstance();
String htmlContents = "<h3>你的第一个html转PDF文档出来啦!! </h3>\n" +
    "<br>\n" +
    "<div style=\"color: aquamarine\">\n" +
    "    注意: html5以及css3的支持还不够完善!!! \n" +
    "</div>\n";
String inputBase64 = Base64.getEncoder().encodeToString(htmlContents.getBytes());
String resultBase64 = converter.convertByBase64(inputBase64);
Base64FileTool.saveBase64File(resultBase64, "./result10.pdf");
```

Base64FileTool这个类来自于qtools-file依赖

• doc转pdf

- 通过文件目录转换

```
BasicConverter converter = new DocToPdfConverterFactory().getInstance();
converter.convertByFilePath("./data/test.doc", "./result4.pdf");
```

- 通过输入流转换

```
BasicConverter converter = new DocToPdfConverterFactory().getInstance();
FileInputStream is = new FileInputStream("./data/test.doc");
FileOutputStream os = new FileOutputStream("./result5.pdf");
converter.convertByStream(is, os);
is.close();
os.close();
```

注意事项

- 在pom.xml中，不要使用system scope引入jar包，而要通过在项目设立一个maven本地仓库：rep目录，将所需要的第三方jar包deploy进去（不能直接从maven下载，原因你懂的）
 - system scope引入的包，在使用jar-with-dependencies打包时将不会被包含，可以使用resources将本地包打进jar-with-dependencies
 - 关于本地repositor的创建和使用，可以参考 [这里](#)
- html转PDF的功能还不够完善，不能完美支持：html5 + css3（或者说能够完美支持html5 + css3破解版本还没有出来）
- 所有转换默认是以A4纸作为最终的PDF页面大小，如果要进行更改的话，请直接使用BasicConverter的getDocument和getDocumentBuilder方法，在获取到Document对象或者DocumentBuilder对象后，修改pageSetup的相关属性，具体请参考aspose的[文档](#)

或者直接参考这一段代码：

```
BasicConverter converter = new HtmlToPdfConverterFactory().getInstance();
DocumentBuilder builder = converter.getDocumentBuilder();
PageSetup pageSetup = builder.getPageSetup();
pageSetup.setPageWidth(2000);
pageSetup.setPageHeight(100);
String htmlFilePath = HtmlToPdfConverterTest.class.getClassLoader().getResource("test.htm").getPath();
```

```
String pdfResultPath = "./result11.pdf";  
converter.convertByFilePath(htmlFilePath, pdfResultPath);  
assertEquals(true, FileTool.isFileExists("./result11.pdf"));
```