**[JAVA使用pdfbox将pdf转换成图片](http://blog.csdn.net/z714303584/article/details/48438601)**

2015-09-14 14:23 2376人阅读 [评论](http://blog.csdn.net/z714303584/article/details/48438601#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/z714303584/article/details/48438601#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

Java（13） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

使用的工具包：

pdfbox-1.8

fontbox-1.8

maven中导入：

<dependency>

<groupId>org.apache.pdfbox</groupId>

<artifactId>pdfbox</artifactId>

<version>1.8.10</version>

</dependency>

注意: 对要求过高项目不建议使用(转化过程中出现字符重叠现象).

下面看代码

**[java]** [view plain](http://blog.csdn.net/z714303584/article/details/48438601) [copy](http://blog.csdn.net/z714303584/article/details/48438601)

1. **package** com.yqrt.core.pdf2img;
3. **import** java.awt.image.BufferedImage;
4. **import** java.io.BufferedInputStream;
5. **import** java.io.FileInputStream;
6. **import** java.io.FileNotFoundException;
7. **import** java.io.FileOutputStream;
8. **import** java.io.IOException;
9. **import** java.io.InputStream;
10. **import** java.util.Iterator;
11. **import** java.util.List;
13. **import** javax.imageio.IIOImage;
14. **import** javax.imageio.ImageIO;
15. **import** javax.imageio.ImageWriter;
16. **import** javax.imageio.stream.ImageOutputStream;
18. **import** org.apache.pdfbox.exceptions.COSVisitorException;
19. **import** org.apache.pdfbox.pdfparser.PDFParser;
20. **import** org.apache.pdfbox.pdmodel.PDDocument;
21. **import** org.apache.pdfbox.pdmodel.PDPage;
23. /\*\*
24. \* Hello world!
25. \*
26. \*/
27. **public** **class** App
28. {
30. **public** **final**  **static** String  IMG\_TYPE\_JPG = "jpg";
31. **public** **final**  **static** String  IMG\_TYPE\_PNG = "png";
32. **public** **static** **void** main( String[] args ) **throws** IOException
33. {
34. App app = **new** App();
35. app.pdf2img("D:\\api.pdf", "D:",IMG\_TYPE\_PNG);
36. //          PDDocument pdDocument = app.pdfInfo("D:\\api.pdf");


40. }

43. /\*\*
44. \*
45. \* @param pdfPath pdf文件的路径
46. \* @param savePath 图片保存的地址
47. \* @param imgType 图片保存方式
48. \*/
49. **public** **void** pdf2img(String pdfPath,String savePath,String imgType){
50. String fileName = pdfPath.substring(pdfPath.lastIndexOf("\\")+1, pdfPath.length());
51. fileName = fileName.substring(0,fileName.lastIndexOf("."));
52. InputStream  is = **null**;
53. PDDocument pdDocument = **null**;
54. **try** {
55. is = **new** BufferedInputStream(**new** FileInputStream(pdfPath));
56. PDFParser parser = **new** PDFParser(is);
57. parser.parse();
58. pdDocument = parser.getPDDocument();
59. List<PDPage> pages = pdDocument.getDocumentCatalog().getAllPages();
60. **for** (**int** i = 0; i < pages.size(); i++) {
61. String saveFileName = savePath+"\\"+fileName+i+"."+imgType;
62. PDPage page =  pages.get(i);
63. pdfPage2Img(page,saveFileName,imgType);
64. }
65. } **catch** (Exception e) {
66. e.printStackTrace();
67. }**finally**{
68. **if**(pdDocument != **null**){
69. **try** {
70. pdDocument.close();
71. } **catch** (IOException e) {
72. e.printStackTrace();
73. }
74. }
75. }

78. }
80. /\*\*
81. \* pdf页转换成图片
82. \* @param page
83. \* @param saveFileName
84. \* @throws IOException
85. \*/
86. **public** **void** pdfPage2Img(PDPage page,String saveFileName,String imgType) **throws** IOException{
87. BufferedImage img\_temp  = page.convertToImage();
88. Iterator<ImageWriter> it = ImageIO.getImageWritersBySuffix(imgType);
89. ImageWriter writer = (ImageWriter) it.next();
90. ImageOutputStream imageout = ImageIO.createImageOutputStream(**new** FileOutputStream(saveFileName));
91. writer.setOutput(imageout);
92. writer.write(**new** IIOImage(img\_temp, **null**, **null**));
93. }


97. **public** PDDocument pdfInfo(String filePath) **throws** IOException{
98. InputStream  is  = **new** BufferedInputStream(**new** FileInputStream(filePath));
99. PDFParser parser = **new** PDFParser(is);
100. parser.parse();
101. PDDocument pdDocument =  parser.getPDDocument();
102. System.out.println("pageNum:"+pdDocument.getNumberOfPages());
103. **return** pdDocument;
104. }




110. **public** **void** createPdf() **throws** COSVisitorException, IOException{
111. PDDocument document = **new** PDDocument();
112. PDPage blankPage = **new** PDPage();
113. document.addPage(blankPage);
114. document.save("D:\\test.pdf");
115. document.close();
116. }
117. }