[java 实现word 转PDF （采用第三方技术 IText、Poi、Jsoup）](http://blog.csdn.net/ptzrbin/article/details/43449701)

标签： [poi word pdf](http://www.csdn.net/tag/poi%20word%20pdf)

2015-02-03 14:48 31005人阅读 [评论](http://blog.csdn.net/ptzrbin/article/details/43449701#comments)(22) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/ptzrbin/article/details/43449701#report)

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

J2EE（31） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

    先讲讲思路：

                       第一步：使用 poi 将word转换成 html，这里代码一搜一堆没什么好说的千篇一律。

                        （值得注意的地方是IText 根据html生成文件的是否会验证html文件是否标准，例如通过poi转换的出来的html文件的一些标签会缺少标签闭合 ，

                     举个例子：

<html>  
                                <head>  
                                                <META http-equiv="Content-Type" content="text/html; charset=UTF-8">

                  <img src="test/0.jpg" style="width:5.765972in;height:8.647917in;vertical-align:text-bottom;">

                               这是我直接用poi生成的html中的一部分, META、img 标签明显就没有闭合标签。如果用这种html进行转换是没有办法通过itext 的校验的。会出现以下异常

错误：        “The element type "meta" must be terminated by the matching end-tag "</meta>".”  
org.xhtmlrenderer.util.XRRuntimeException: Can't load the XML resource (using TRaX transformer). org.xml.sax.SAXParseException: The element type "meta" must be terminated by the matching end-tag "</meta>".  。

               从错误分析也知道是我们的html不规范拉，我们采用第三方 jar 包  Jsoup，  直接调用parse方法 我们的html就标准啦！

因为遇到这个问题让我头疼了半天，没想到就这么轻松的解决了，发个博文支援一下遇到该问题的小伙伴们 ！

下面是poi转换html 的代码：

**[java]** [view plaincopy](http://blog.csdn.net/zk_spring/article/details/43412221)

1. package com.smart.sys.core.service.io.poi;
3. import org.apache.poi.hwpf.HWPFDocument;
4. import org.apache.poi.hwpf.converter.PicturesManager;
5. import org.apache.poi.hwpf.converter.WordToHtmlConverter;
6. import org.apache.poi.hwpf.usermodel.Picture;
7. import org.apache.poi.hwpf.usermodel.PictureType;
8. import org.jsoup.Jsoup;
9. import org.w3c.dom.Document;
11. import javax.xml.parsers.DocumentBuilderFactory;
12. import javax.xml.parsers.ParserConfigurationException;
13. import javax.xml.transform.OutputKeys;
14. import javax.xml.transform.Transformer;
15. import javax.xml.transform.TransformerException;
16. import javax.xml.transform.TransformerFactory;
17. import javax.xml.transform.dom.DOMSource;
18. import javax.xml.transform.stream.StreamResult;
19. import java.io.\*;
20. import java.util.List;
22. /\*\*
23. \* Created by Carey on 15-2-2.
24. \*/
25. public class Word2Html {

28. public static void main(String argv[]) {
29. try {
30. convert2Html("D:\\新建 Microsoft Word 文档.doc","D:\\1.html");
31. } catch (Exception e) {
32. e.printStackTrace();
33. }
34. }
36. //输出html文件
37. public static void writeFile(String content, String path) {
38. FileOutputStream fos = null;

**[java]** [view plaincopy](http://blog.csdn.net/zk_spring/article/details/43412221)

1. BufferedWriter bw = null;
2. org.jsoup.nodes.Document doc = Jsoup.parse(content);
3. content=doc.html();
4. try {
5. File file = new File(path);
6. fos = new FileOutputStream(file);
7. bw = new BufferedWriter(new OutputStreamWriter(fos,"UTF-8"));
8. bw.write(content);
9. } catch (FileNotFoundException fnfe) {
10. fnfe.printStackTrace();
11. } catch (IOException ioe) {
12. ioe.printStackTrace();
13. } finally {
14. try {
15. if (bw != null)
16. bw.close();
17. if (fos != null)
18. fos.close();
19. } catch (IOException ie) {
20. }
21. }
22. }
24. //word 转 html
25. public static void convert2Html(String fileName, String outPutFile)
26. throws TransformerException, IOException,
27. ParserConfigurationException {
29. HWPFDocument wordDocument = new HWPFDocument(new FileInputStream(fileName));//WordToHtmlUtils.loadDoc(new FileInputStream(inputFile));
30. //兼容2007 以上版本
31. //        XSSFWorkbook  xssfwork=new XSSFWorkbook(new FileInputStream(fileName));
32. WordToHtmlConverter wordToHtmlConverter = new WordToHtmlConverter(
33. DocumentBuilderFactory.newInstance().newDocumentBuilder()
34. .newDocument());
35. wordToHtmlConverter.setPicturesManager( new PicturesManager()
36. {
37. public String savePicture( byte[] content,
38. PictureType pictureType, String suggestedName,
39. float widthInches, float heightInches )
40. {
41. return "test/"+suggestedName;
42. }
43. } );
44. wordToHtmlConverter.processDocument(wordDocument);
45. //save pictures
46. List pics=wordDocument.getPicturesTable().getAllPictures();
47. if(pics!=null){
48. for(int i=0;i<pics.size();i++){
49. Picture pic = (Picture)pics.get(i);
50. System.out.println();
51. try {
52. pic.writeImageContent(new FileOutputStream("D:/test/"
53. + pic.suggestFullFileName()));
54. } catch (FileNotFoundException e) {
55. e.printStackTrace();
56. }
57. }
58. }
59. Document htmlDocument = wordToHtmlConverter.getDocument();
61. ByteArrayOutputStream out = new ByteArrayOutputStream();
62. DOMSource domSource = new DOMSource(htmlDocument);
63. StreamResult streamResult = new StreamResult(out);

66. TransformerFactory tf = TransformerFactory.newInstance();
67. Transformer serializer = tf.newTransformer();
68. serializer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
69. serializer.setOutputProperty(OutputKeys.INDENT, "yes");
70. serializer.setOutputProperty(OutputKeys.METHOD, "HTML");
71. serializer.transform(domSource, streamResult);
72. out.close();
73. writeFile(new String(out.toByteArray()), outPutFile);
74. }
75. }

  好了第二步生成pdf ，我直接上代码了 ！

**[java]** [view plaincopy](http://blog.csdn.net/zk_spring/article/details/43412221)

1. package com.smart.sys.core.service.io.itext;
3. import com.lowagie.text.pdf.BaseFont;
4. import org.xhtmlrenderer.pdf.ITextFontResolver;
5. import org.xhtmlrenderer.pdf.ITextRenderer;
7. import java.io.File;
8. import java.io.FileOutputStream;
9. import java.io.OutputStream;
11. /\*\*
12. \* Created by Carey on 15-2-2.
13. \*/
14. public class Html2Pdf {

17. public boolean convertHtmlToPdf(String inputFile, String outputFile)
18. throws Exception {
20. OutputStream os = new FileOutputStream(outputFile);
21. ITextRenderer renderer = new ITextRenderer();
22. String url = new File(inputFile).toURI().toURL().toString();
23. renderer.setDocument(url);
24. // 解决中文支持问题
25. ITextFontResolver fontResolver = renderer.getFontResolver();
26. fontResolver.addFont("C:/Windows/Fonts/simsunb.ttf", BaseFont.IDENTITY\_H, BaseFont.NOT\_EMBEDDED);
27. //解决图片的相对路径问题
28. renderer.getSharedContext().setBaseURL("file:/D:/test");
29. renderer.layout();
30. renderer.createPDF(os);
31. os.flush();
32. os.close();
33. return true;
34. }

37. public   static  void  main(String [] args){
38. Html2Pdf html2Pdf =new Html2Pdf();
39. try {
40. html2Pdf.convertHtmlToPdf("D:\\1.html","D:\\index.pdf");
41. } catch (Exception e) {
42. e.printStackTrace();
43. }
44. }
45. }

 pdf 的转换的详细细节也是参考了一位大神的代码，人家写的非常详细，我这里也不再赘述 、这里附上链接 ：

          http://www.open-open.com/lib/view/open1341881830588.html

所需jar包

iText-2.0.8.jar

core-renderer.jar

iTextAsian.jar

iTextAsianCmaps.jar

jsoup-1.8.1.jar

下载地址：

http://download.csdn[**.NET**](http://lib.csdn.net/base/dotnet)/detail/ptzrbin/8419791