**Java使用poi将word转换为html**

作者：繁华穿越现实 字体：[[增加](javascript:turnbig()) [减小](javascript:turnsmall())] 类型：转载 时间：2016-12-30[我要评论](http://www.jb51.net/article/101730.htm#comments)

这篇文章主要为大家详细介绍了Java使用poi将word转换为html的相关资料，具有一定的参考价值，感兴趣的小伙伴们可以参考一下

使用poi将word转换为html，支持doc,docx，转换后可以保持图片、样式。

1.导入Maven包

[?](http://www.jb51.net/article/101730.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | <dependency>   <groupId>org.apache.poi</groupId>   <artifactId>poi</artifactId>   <version>3.14</version>  </dependency>  <dependency>   <groupId>org.apache.poi</groupId>   <artifactId>poi-scratchpad</artifactId>   <version>3.14</version>  </dependency>  <dependency>   <groupId>org.apache.poi</groupId>   <artifactId>poi-ooxml</artifactId>   <version>3.14</version>  </dependency>  <dependency>   <groupId>fr.opensagres.xdocreport</groupId>   <artifactId>xdocreport</artifactId>   <version>1.0.6</version>  </dependency>  <dependency>   <groupId>org.apache.poi</groupId>   <artifactId>poi-ooxml-schemas</artifactId>   <version>3.14</version>  </dependency>  <dependency>   <groupId>org.apache.poi</groupId>   <artifactId>ooxml-schemas</artifactId>   <version>1.3</version>  </dependency> |

2.转换代码

[?](http://www.jb51.net/article/101730.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72 | import org.apache.poi.hwpf.HWPFDocument;  import org.apache.poi.hwpf.converter.WordToHtmlConverter;  import org.apache.poi.xwpf.converter.core.BasicURIResolver;  import org.apache.poi.xwpf.converter.core.FileImageExtractor;  import org.apache.poi.xwpf.converter.xhtml.XHTMLConverter;  import org.apache.poi.xwpf.converter.xhtml.XHTMLOptions;  import org.apache.poi.xwpf.usermodel.XWPFDocument;  import org.w3c.dom.Document;    import javax.xml.parsers.DocumentBuilderFactory;  import javax.xml.transform.OutputKeys;  import javax.xml.transform.Transformer;  import javax.xml.transform.TransformerFactory;  import javax.xml.transform.dom.DOMSource;  import javax.xml.transform.stream.StreamResult;  import java.io.File;  import java.io.FileInputStream;  import java.io.FileOutputStream;  import java.io.OutputStreamWriter;    public class Test {    // doc转换为html    void docToHtml() throws Exception {      String sourceFileName = "C:\\doc\\test.doc";      String targetFileName = "C:\\html\\test.html";      String imagePathStr = "C:\\html\\image\\";      HWPFDocument wordDocument = new HWPFDocument(new FileInputStream(sourceFileName));      Document document = DocumentBuilderFactory.newInstance().newDocumentBuilder().newDocument();      WordToHtmlConverter wordToHtmlConverter = new WordToHtmlConverter(document);      // 保存图片，并返回图片的相对路径      wordToHtmlConverter.setPicturesManager((content, pictureType, name, width, height) -> {        try(FileOutputStream out = new FileOutputStream(imagePathStr + name)){           out.write(content);        } catch (Exception e) {          e.printStackTrace();        }        return "image/" + name;      });      wordToHtmlConverter.processDocument(wordDocument);      Document htmlDocument = wordToHtmlConverter.getDocument();      DOMSource domSource = new DOMSource(htmlDocument);      StreamResult streamResult = new StreamResult(new File(targetFileName));        TransformerFactory tf = TransformerFactory.newInstance();      Transformer serializer = tf.newTransformer();      serializer.setOutputProperty(OutputKeys.ENCODING, "utf-8");      serializer.setOutputProperty(OutputKeys.INDENT, "yes");      serializer.setOutputProperty(OutputKeys.METHOD, "html");      serializer.transform(domSource, streamResult);    }    // docx转换为html    public void docxToHtml() throws Exception {      String sourceFileName = "D:\\ac\\00.docx";      String targetFileName = "D:\\ac\\test.html";      String imagePathStr = "D:\\ac\\image\\";      OutputStreamWriter outputStreamWriter = null;      try {        XWPFDocument document = new XWPFDocument(new FileInputStream(sourceFileName));        XHTMLOptions options = XHTMLOptions.create();        // 存放图片的文件夹        options.setExtractor(new FileImageExtractor(new File(imagePathStr)));        // html中图片的路径        options.URIResolver(new BasicURIResolver("image"));        outputStreamWriter = new OutputStreamWriter(new FileOutputStream(targetFileName), "utf-8");        XHTMLConverter xhtmlConverter = (XHTMLConverter) XHTMLConverter.getInstance();        xhtmlConverter.convert(document, outputStreamWriter, options);      } finally {        if (outputStreamWriter != null) {          outputStreamWriter.close();        }      }    } |

演示地址: <https://www.xiaoyun.studio/app/preview.html>

以上就是本文的全部内容，希望对大家的学习有所帮助，也希望大家多多支持脚本之家。