



链滴

Java poi 将 word 转成 html, ppt 转成图片

作者: [xiaoli](#)

原文链接: <https://ld246.com/article/1530243840991>

来源网站: [链滴](#)

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

Java 容易被忽视的 poi 包

Poi 提取技术是 java 中一大亮点。前几日在做项目时需要将 ppt 转化为图片，将 word 文档输的功能。在百度搜了些资料，整理了一下，借用前辈们的经验整合了这两个小小的类

文章原出两处：[http://vtrtbb.iteye.com/blog/60167](https://ld246.com/forward?goto=http%3A%2F%2Fvtrtbb.iteye.com%2Fblog%2F601267)

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">http://pengenjing.iteye.com/blog/1901225
```

```
</span></span></code></pre>
```

来看下面的吧 word 文档转化为 html 格式的代码先：

```
<code>package com.hsp.util;
```

```
import java.io.BufferedWriter;
```

```
import java.io.File;
```

```
import java.io.FileInputStream;
```

```
import java.io.FileNotFoundException;
```

```
import java.io.FileOutputStream;
```

```
import java.io.IOException;
```

```
import java.io.OutputStream;
```

```
import java.io.OutputStreamWriter;
```

```
import java.util.List;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.parsers.ParserConfigurationException;
```

```
import javax.xml.transform.OutputKeys;
```

```
import javax.xml.transform.Transformer;
```

```
import javax.xml.transform.TransformerException;
```

```
import javax.xml.transform.TransformerFactory;
```

```
import javax.xml.transform.dom.DOMSource;
```

```
import javax.xml.transform.stream.StreamResult;
```

```
import org.apache.commons.io.output.ByteArrayOutputStream;
```

```
import org.apache.poi.hwpf.HWPFDocument;
```

```
import org.apache.poi.hwpf.converter.PicturesManager;
```

```
import org.apache.poi.hwpf.converter.WordToHtmlConverter;
```

```
import org.apache.poi.hwpf.model.PicturesTable;
```

```
import org.apache.poi.hwpf.usermodel.CharacterRun;
```

```
import org.apache.poi.hwpf.usermodel.Paragraph;
```

```
import org.apache.poi.hwpf.usermodel.Picture;
```

```
import org.apache.poi.hwpf.usermodel.PictureType;
```

```
import org.apache.poi.hwpf.usermodel.Range;
```

```
import org.apache.poi.hwpf.usermodel.Table;
```

```
import org.apache.poi.hwpf.usermodel.TableCell;
```

```
import org.apache.poi.hwpf.usermodel.TableIterator;
```

```
import org.apache.poi.hwpf.usermodel.TableRow;
```

```
import org.w3c.dom.Document;
```

```
public class Word2Html {
```

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">/*newrealpath是指word文档的原路径，realpath是指word文档的新路径，newsid是指该word文档存储在数据库中的唯一id值，用户没上传一个word文档，后台会根据生成的唯一id值创建一个.html文件，就是newsid.html*/
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">public static boolean changeWord2html(String newrealpath,String realpath,int newsid) {
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">boolean b=true;
```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> try {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> convert2Html
newrealpath,realpath+"//" +newsid+".html");
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> } catch (Excepti
n e) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> e.printStackTrace();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> b=false;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> return b;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p>//该方法是将文本写入路径之中</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">public static void writeFile(String content, String path) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> FileOutputStream
fos = null;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> BufferedWriter
w = null;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> try {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> File file = ne
File(path);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> fos = new Fil
OutputStream(file);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> bw = new Bu
feredWriter(new OutputStreamWriter(fos,"GB2312"));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> bw.write(con
ent);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> } catch (FileNot
FileNotFoundException fnfe) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> fnfe.printSta
kTrace();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> } catch (IOExcep
tion ioe) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> ioe.printStac

```

```

Trace();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> } finally {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> try {
</span></span><span class="highlight-line"><span class="highlight-cl"> if (bw != n
</span></span><span class="highlight-line"><span class="highlight-cl"> ll)
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> bw.close
);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> if (fos != n
ll)
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> fos.close
);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> } catch (IOEx
eption ie) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> public static void
onvert2Html(String fileName, String outPutFile)
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> throws Trans
ormerException, IOException,
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> ParserConfig
rationException {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> HWPFDocumen
wordDocument = new HWPFDocument(new FileInputStream(fileName));//创建一个文档
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> WordToHtmlC
nverter wordToHtmlConverter = new WordToHtmlConverter(
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> Document
uilderFactory.newInstance().newDocumentBuilder()
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> .new
ocument());//对普通文本的操作
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> wordToHtmlC
nverter.setPicturesManager( new PicturesManager()
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> public String

```

```

savePicture( byte[] content,
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    Picture
type pictureType, String suggestedName,
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    float width, float heightInches, float widthInches, float heightInches )
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    return "test";
</span></span><span class="highlight-line"><span class="highlight-cl">
    }
    };//对图片的
作
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    wordToHtmlConverter.processDocument(wordDocument);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    //保存图片
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    List<Picture> pics=wordDocument.getPicturesTable().getAllPictures();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
    if(pics!=null){
</span></span><span class="highlight-line"><span class="highlight-cl">
        for(int i=0;i<pics.size();i++){
</span></span><span class="highlight-line"><span class="highlight-cl">
            TransformerFactory tf = TransformerFactory.newInstance();
</span></span><span class="highlight-line"><span class="highlight-cl">
            Transformer serializer = tf.newTransformer();
</span></span><span class="highlight-line"><span class="highlight-cl">
            serializer.setOutputProperty(OutputKeys.ENCODING, "GB2312");
</span></span><span class="highlight-line"><span class="highlight-cl">
            serializer.setOutputProperty(OutputKeys.INDENT, "yes");
</span></span><span class="highlight-line"><span class="highlight-cl">
            serializer.setOutputProperty(OutputKeys.METHOD, "html");
</span></span><span class="highlight-line"><span class="highlight-cl">
            serializer.transform(domSource, streamResult);
</span></span><span class="highlight-line"><span class="highlight-cl">
            out.close();
</span></span><span class="highlight-line"><span class="highlight-cl">
            writeFile(new String(out.toByteArray()), outPutFile); //调用writeFile类
</span></span><span class="highlight-line"><span class="highlight-cl">
        }
    }
}

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span></code></pre>
<p></p>
<p>下面的我们来看看 ppt 转化为图片格式的代码: </p>
<p>package com.hsp.util;</p>
<p>import java.awt.Dimension;</p>
<p>import java.io.File;</p>
<p>import java.io.FileInputStream;</p>
<p>import java.io.FileNotFoundException;</p>
<p>import java.io.FileOutputStream;</p>
<p>import java.io.IOException;</p>
<p>import java.util.ArrayList;</p>
<p>import java.util.List;</p>
<p>import java.awt.Color;</p>
<p>import java.awt.Dimension;</p>
<p>import java.awt.Graphics2D;</p>
<p>import java.awt.geom.Rectangle2D;</p>
<p>import java.awt.image.BufferedImage;</p>
<p>import org.apache.poi.hslf.model.TextRun;</p>
<p>import org.apache.poi.hslf.record.Slide;</p>
<p>import org.apache.poi.hslf.usermodel.RichTextRun;</p>
<p>import org.apache.poi.hslf.usermodel.SlideShow;</p>
<p>public class PPTtoImage {</p>
<p></em>pptload 是指 ppt 的原路径, newfullnewpath 是指 ppt 的新路径, newsid 是指该 ppt
存储在数据库中的唯一 id 值, 用户没上传一个 ppt, 后台会根据生成的唯一 id 值创建一个唯一的文
夹, 该文件夹内存放该 ppt 转化的图片</em></p>
<p>public static List changppttoimage(String pptload,String newfullnewpath,int newsid){</p>

<p>// 读入 PPT 文件</p>
<p>File file = new File(pptload);</p>
<p>return doPPTtoImage(file,newfullnewpath,newsid);</p>
<p>}</p>
<p>public static List doPPTtoImage(File file,String newfullnewpath,int newsid){</p>
<p>/*boolean isppt = checkFile(file);</p>
<p>if (!isppt) {</p>
<p>System.out.println("你指定的文件不是 ppt 文档!");</p>
<p>return false;</p>
<p>}*</p>
<p>List al=new ArrayList();</p>
<p>try {</p>
<p>FileInputStream is = new FileInputStream(file);</p>
<p>SlideShow ppt = new SlideShow(is);</p>
<p>is.close();</p>
<p>Dimension pgsz = ppt.getPageSize();</p>
<p>org.apache.poi.hslf.model.Slide[] slide = ppt.getSlides();</p>
<p>for (int i = 0; i <= slide.length; i++) {</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight"
cl">//System.out.print("第" + i + "页。");
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">if(slide[i].getNote
Sheet()!=null&amp;&amp;slide[i].getNotesSheet().getTextRuns()!=null){
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> //获取第一个备注
</span></span><span class="highlight-line"><span class="highlight-cl">

```



```

</span></span><span class="highlight-line"><span class="highlight-cl"> //System.out.print
n("备注: " + slide[i].getNotesSheet().getTextRuns()[0].getText());
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">TextRun[] truns =
lide[i].getTextRuns();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">for (int k = 0; k <&lt;
truns.length; k++) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> RichTextRun[] rtr
ns = truns[k].getRichTextRuns();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> for (int l = 0; l <&lt;
rtruns.length; l++) {
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> rtruns[l].setFontI
dex(1);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> rtruns[l].setFont
ame("宋体");
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> // 获取文本列表
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> //System.out.prin
ln(rtruns[l].getText());
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">}
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">BufferedImage im
= new BufferedImage(pgsz.width,
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> pgsz.height, Bu
feredImage.TYPE_INT_RGB);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">Graphics2D graph
cs = img.createGraphics();
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">graphics.setPaint(
olor.white);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">graphics.fill(new R
ctangle2D.Float(0, 0, pgsz.width,
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> pgsz.height));
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">slide[i].draw(graph
cs);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">// 这里设置图片的
放路径和图片的格式(jpeg,png,bmp等等),注意生成文件路径

```

```

</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">File f=new File(new
fullnewpath);
</span></span></code></pre>
<p>//判断该文件夹是否存在, 如果不存在, 这创建一个新的文件夹</p>
<p>if(!f.isDirectory())</p>
<p>{</p>
<p>f.mkdirs();</p>
<p>}</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">FileOutputStream out = new FileOutputStream(newfullnewpath+"pict_"+(i + 1) + ".jpeg");
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">javax.imageio.ImageIO.write(img, "jpeg", out);
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">al.add("pict_"+(i + 1) + ".jpeg");
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">out.close();
</span></span></code></pre>
<p>}</p>
<p>/*System.out.println("ok");</p>
<p>return true;*/</p>
<p>} catch (FileNotFoundException e) {</p>
<p>System.out.println(e);</p>
<p>} catch (IOException e) {</p>
<p>e.printStackTrace();</p>
<p>}</p>
<p>return al;</p>
<p>}</p>
<p>// function 检查文件是否为 PPT</p>
<p>public static boolean checkFile(File file) {</p>
<p>boolean isppt = false;</p>
<p>String filename = file.getName();</p>
<p>String suffixname = null;</p>
<p>if (filename != null && filename.indexOf(".") != -1) {</p>
<p>suffixname = filename.substring(filename.indexOf("."));</p>
<p>if (suffixname.equals(".ppt")) {</p>
<pre><code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">isppt = true;
</span></span></code></pre>
<p>}</p>
<p>return isppt;</p>
<p>} else {</p>
<p>return isppt;</p>
<p>}</p>
<p>}</p>
<p>}</p>
<p>}</p>
<p>经测试是可行的, 在小编这里是可以运行成功的, 不过, 上面只是简单的两个类而已, 在项目中还要用 struts2 去限定转化文件的大小等等。</p>

```