**Java操作Word**

1. **Poi工具**

Apache的POI，是Apache软件基金会的开放源码函式库，POI提供API给Java程序对Microsoft Office格式档案读和写的功能。POI读写Excel功能强大、操作简单。但是POI操作时，一般只用它读取word文档，POI只能能够创建简单的word文档，相对而言POI操作时的功能太少。

**1、添加Poi依赖**

<dependency>

<groupId>org.apache.poi</groupId>  
 <artifactId>poi</artifactId>  
 <version>3.15</version>  
</dependency>  
<dependency>  
 <groupId>org.apache.poi</groupId>  
 <artifactId>poi-ooxml-schemas</artifactId>  
 <version>3.15</version>  
</dependency>  
<dependency>  
 <groupId>org.apache.poi</groupId>  
 <artifactId>poi-ooxml</artifactId>  
 <version>3.15</version>  
</dependency>

**2、导出**

1）导出语句如下。

public void upload() throws IOException {

XWPFDocument doc = new XWPFDocument();// 创建Word文件  
 XWPFParagraph p = doc.createParagraph();// 新建一个段落  
 p.setAlignment(ParagraphAlignment.*CENTER*);// 设置段落的对齐方式  
 p.setBorderBottom(Borders.*DOUBLE*);//设置下边框  
 p.setBorderTop(Borders.*DOUBLE*);//设置上边框  
 p.setBorderRight(Borders.*DOUBLE*);//设置右边框  
 p.setBorderLeft(Borders.*DOUBLE*);//设置左边框  
 XWPFRun r = p.createRun();//创建段落文本  
 r.setText("POI创建的Word段落文本");  
 r.setBold(true);//设置为粗体  
 r.setColor("FF0000");//设置颜色  
 p = doc.createParagraph();// 新建一个段落  
 r = p.createRun();  
 r.setText("POI读写Excel功能强大、操作简单。");  
 XWPFTable table= doc.createTable(3, 3);//创建一个表格  
 table.getRow(0).getCell(0).setText("表格1");  
 table.getRow(1).getCell(1).setText("表格2");  
 table.getRow(2).getCell(2).setText("表格3");  
 FileOutputStream out = new FileOutputStream("D:\\记事本\\公司\\word\\sample.doc");  
 doc.write(out);  
 out.close();  
}

1. 调用接口。导出后的word如下。

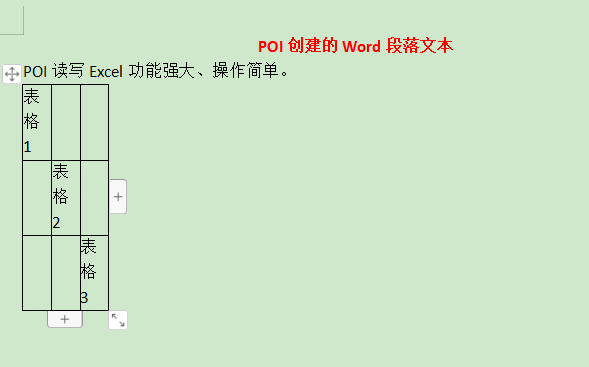


图1 导出结果图

**3、导入**

1）导入语句如下。

public void importWord(@RequestParam("file") MultipartFile file) throws IOException {

if (file == null) {  
 System.*out*.println("文件不能为空");  
 }  
 // 创建Word文件  
 XWPFDocument doc = new XWPFDocument(file.getInputStream());  
 //遍历段落  
 for (XWPFParagraph p : doc.getParagraphs())  
 {  
 System.*out*.println(p.getParagraphText());  
 }  
 //遍历表格  
 for (XWPFTable table : doc.getTables())  
 {  
 for (XWPFTableRow row : table.getRows()) {  
 for (XWPFTableCell cell : row.getTableCells()) {  
 System.*out*.println(cell.getText());  
 }  
 }  
 }  
}

2）调用接口，导入之前导出的word文本。导入后的控制台输出如下。

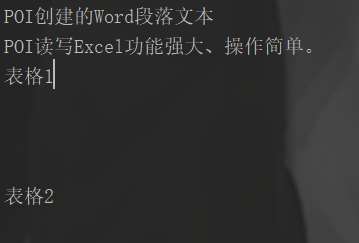


图2 导入结果图

1. **使用Freemarker模板进行生成**

**优点：**采用**FreeMarker**是导出Word的最佳实现，非常的灵活，能够按照自己指定的样式设置并输出内容，操作简单方便，代码实现也容易。代码量少，样式、内容容易控制，打印不变形，完全符合office标准。

**缺点：**需要提前设计好word模板，把需要替换的地方用特殊标记标出来。

**1、添加freemarker依赖**

<!--引入freemarker 模板依赖-->

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-freemarker</artifactId>  
</dependency>

**2、准备要导出的word模板**

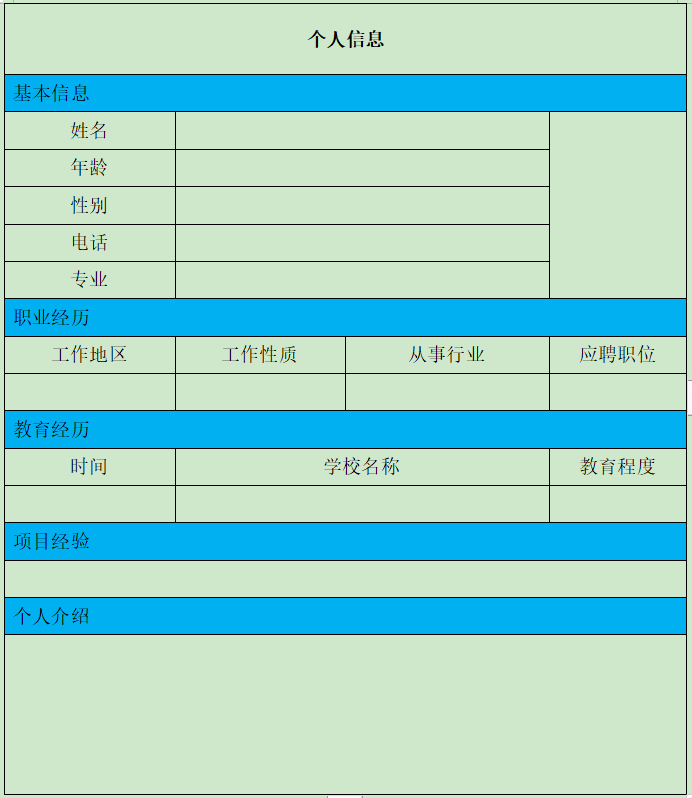


图3 导出模板图

**3、制作基本样式之后，另存为.xml格式文档**

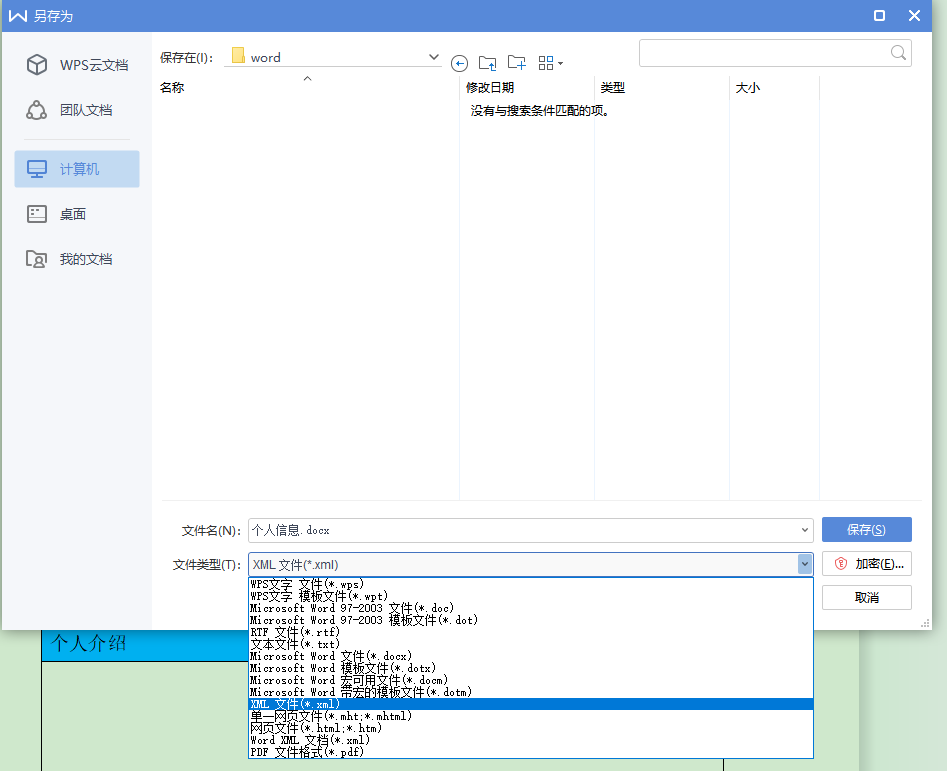


图4 xml保存图

**4、打开.xml文件，在相应的地方填入${content}表达式**



图5 填入参数图

**5、使用文本编辑器sublime打开**



图6 xml图

**5、复制黏贴至项目文件中，后缀名更改为.ftl**

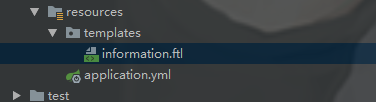


图7 修改成ftl图

**6、编写导出类**

package com.lamarsan.word\_demo.freemarker;

import freemarker.template.\*;  
import sun.misc.BASE64Encoder;  
  
import java.io.\*;  
import java.util.HashMap;  
import java.util.Map;  
import java.util.logging.Logger;  
  
*/\*\*  
 \* className: ExportMyWord  
 \* description: TODO  
 \*  
 \** ***@author*** *hasee  
 \** ***@version*** *1.0  
 \** ***@date*** *2019/7/15 14:54  
 \*/*public class ExportMyWord {  
 private Logger log = Logger.*getLogger*(ExportMyWord.class.toString());  
 private Configuration config = null;  
  
 public ExportMyWord() {  
 config = new Configuration();  
 config.setDefaultEncoding("utf-8");  
 }  
  
 */\*\*  
 \* FreeMarker生成Word  
 \** ***@param*** *dataMap 数据  
 \** ***@param*** *templateName 目标名  
 \** ***@param*** *saveFilePath 保存文件路径的全路径名（路径+文件名）  
 \** ***@Author*** *Huang Xiaocong 2018年12月15日 下午10:19:03  
 \*/* public void createWord(Map<String, Object> dataMap, String templateName, String saveFilePath) {  
 //加载模板(路径)数据  
 config.setClassForTemplateLoading(this.getClass(), "/templates");  
 //设置异常处理器 这样的话 即使没有属性也不会出错 如：${list.name}...不会报错  
 config.setTemplateExceptionHandler(TemplateExceptionHandler.*IGNORE\_HANDLER*);  
 Template template = null;  
 if(templateName.endsWith(".ftl")) {  
 templateName = templateName.substring(0, templateName.indexOf(".ftl"));  
 }  
 try {  
 template = config.getTemplate(templateName + ".ftl");  
 } catch (TemplateNotFoundException e) {  
 log.info("模板文件未找到");  
 e.printStackTrace();  
 } catch (MalformedTemplateNameException e) {  
 log.info("模板类型不正确");  
 e.printStackTrace();  
 } catch (IOException e) {  
 log.info("IO读取失败");  
 e.printStackTrace();  
 }  
 File outFile = new File(saveFilePath);  
 if(!outFile.getParentFile().exists()) {  
 outFile.getParentFile().mkdirs();  
 }  
 Writer out = null;  
 FileOutputStream fos = null;  
 try {  
 fos = new FileOutputStream(outFile);  
 } catch (FileNotFoundException e) {  
 log.info("输出文件时未找到文件");  
 e.printStackTrace();  
 }  
 out = new BufferedWriter(new OutputStreamWriter(fos));  
 //将模板中的预先的代码替换为数据  
 try {  
 template.process(dataMap, out);  
 } catch (TemplateException e) {  
 log.info("填充模板时异常");  
 e.printStackTrace();  
 } catch (IOException e) {  
 log.info("IO读取时异常");  
 e.printStackTrace();  
 }  
 log.info("由模板文件：" + templateName + ".ftl" + " 生成文件 ：" + saveFilePath + " 成功！！");  
 try {  
 out.close();//web项目不可关闭  
 } catch (IOException e) {  
 log.info("关闭Write对象出错");  
 e.printStackTrace();  
 }  
 }  
 */\*\*  
 \* 获得图片的Base64编码  
 \** ***@param*** *imgFile  
 \** ***@return*** *\** ***@Author*** *Huang Xiaocong 2018年12月15日 下午10:15:10  
 \*/* public String getImageStr(String imgFile) {  
 InputStream in = null;  
 byte[] data = null;  
 try {  
 in = new FileInputStream(imgFile);  
 } catch (FileNotFoundException e) {  
 log.info("加载图片未找到");  
 e.printStackTrace();  
 }  
 try {  
 data = new byte[in.available()];  
 //注：FileInputStream.available()方法可以从输入流中阻断由下一个方法调用这个输入流中读取的剩余字节数  
 in.read(data);  
 in.close();  
 } catch (IOException e) {  
 log.info("IO操作图片错误");  
 e.printStackTrace();  
 }  
 BASE64Encoder encoder = new BASE64Encoder();  
 return encoder.encode(data);  
  
 }  
  
}

需要注意的是，在加载config.setClassForTemplateLoading(this.getClass(), "/templates");时，需要指定存放freemarker文件的文件目录地址，即templates，不然会报org.springfra

mework.beans.factory.BeanCreationException: Error creating bean with…错误，即找不到模板文件。

**7、编写main方法**

public static void main(String[] args) {

ExportMyWord emw = new ExportMyWord();  
 Map<String, Object> dataMap = new HashMap<String, Object>();  
 dataMap.put("name", "傅泽鹏");  
 dataMap.put("age", 20);  
 dataMap.put("sex", "男");  
 dataMap.put("imgheader", emw.getImageStr("D:\\图片\\130.png"));  
 dataMap.put("phoneNum", "13588097790");  
 dataMap.put("project", "计算机");  
 dataMap.put("address", "杭州");  
 dataMap.put("natureWork", "全职");  
 dataMap.put("industry", "IT");  
 dataMap.put("application", "Java开发");  
 dataMap.put("time", "2017年-2021年");  
 dataMap.put("schoolname", "浙江科技学院");  
 dataMap.put("education", "本科");  
 dataMap.put("projectExperience", "辩论小程序开发");  
 dataMap.put("introduce", "喜欢二次元，编程的大学生");  
 emw.createWord(dataMap, "information.ftl", "D:\\记事本\\公司\\简历.docx");  
}

**8、修改ftl文件的图片代码**

如果立马点击生成docx文件，会发现图片变成了一堆超级长的代码。所以可以判断是ftl处的代码有误。进入ftl代码，查找${imgheader}，修改<w:p>标签如下。

<w:p>

<w:pPr>  
 <w:widowControl w:val="off"/>  
 <w:jc w:val="center"/>  
 <w:rPr>  
 <w:rFonts w:hint="default"/>  
 <w:b w:val="off"/>  
 <w:sz w:val="24"/>  
 <w:sz-cs w:val="32"/>  
 <w:vertAlign w:val="baseline"/>  
 <w:lang w:val="EN-US" w:fareast="ZH-CN"/>  
 </w:rPr>  
 </w:pPr>  
 <w:r>  
 <w:rPr>  
 <w:rFonts w:hint="fareast"/>  
 <w:b w:val="off"/>  
 <w:sz w:val="24"/>  
 <w:sz-cs w:val="32"/>  
 <w:vertAlign w:val="baseline"/>  
 <w:lang w:val="EN-US" w:fareast="ZH-CN"/>  
 </w:rPr>  
 <w:pict>  
 <w:binData w:name="wordml://1.png">**${**imgheader**}**</w:binData>  
 <v:shape id="\_x0000\_s1026" o:spt="75" alt="" type="#\_x0000\_t75"  
 style="height:119.8pt;width:74.65pt;" filled="f" o:preferrelative="t"  
 stroked="f" coordsize="21600,21600">  
 <v:path/>  
 <v:fill on="f" focussize="0,0"/>  
 <v:stroke on="f"/>  
 <v:imagedata src="wordml://1.png" o:title="140"/>  
 <o:lock v:ext="edit" aspectratio="t"/>  
 <w10:wrap type="none"/>  
 <w10:anchorlock/>  
 </v:shape>  
 </w:pict>  
 </w:r>  
</w:p>

**9、最终效果图**



图8 最终生成图

1. **补充循环操作**

如果需要多重循环可以用list来实现。

1）目标模板如下：



图9 最终生成图

1. 查找字段，在<w:tr>前加入<#list list as list>，</w:tr>后加入</#list>。

第一个list 不能变，第二个list 为变量名，第三个list 为别名。

1. 将time改成list.time,schoolname改成list.schoolname,education改成

list.eduxcation。

5）运行代码如下：

public static void main(String[] args) {

ExportMyWord emw = new ExportMyWord();  
 Map<String, Object> dataMap = new HashMap<String, Object>();  
 dataMap.put("name", "傅泽鹏");  
 dataMap.put("age", 20);  
 dataMap.put("sex", "男");  
 dataMap.put("imgheader", emw.getImageStr("D:\\图片\\130.png"));  
 dataMap.put("phoneNum", "13588097790");  
 dataMap.put("project", "计算机");  
 dataMap.put("address", "杭州");  
 dataMap.put("natureWork", "全职");  
 dataMap.put("industry", "IT");  
 dataMap.put("application", "Java开发");  
 List<Map<String, Object>> newsList = new ArrayList<Map<String, Object>>();  
 Map<String, Object> map = new HashMap<String, Object>();  
 Map<String, Object> map2 = new HashMap<String, Object>();  
 Map<String, Object> map3 = new HashMap<String, Object>();  
 map.put("time", "2017年-2021年");  
 map.put("schoolname", "浙江科技学院");  
 map.put("education", "本科");  
 map2.put("time", "2014年-2017年");  
 map2.put("schoolname", "瑞安四中");  
 map2.put("education", "高中");  
 map3.put("time", "2011年-2014年");  
 map3.put("schoolname", "莘塍一中");  
 map3.put("education", "初中");  
 newsList.add(map);  
 newsList.add(map2);  
 newsList.add(map3);  
 //注意list 的名字  
 dataMap.put("list", newsList);  
 dataMap.put("projectExperience", "辩论小程序开发");  
 dataMap.put("introduce", "喜欢二次元，编程的大学生");  
 emw.createWord(dataMap, "information\_list.ftl", "D:\\记事本\\公司\\简历2.docx");  
}

6）最终结果图：



图10 最终生成图