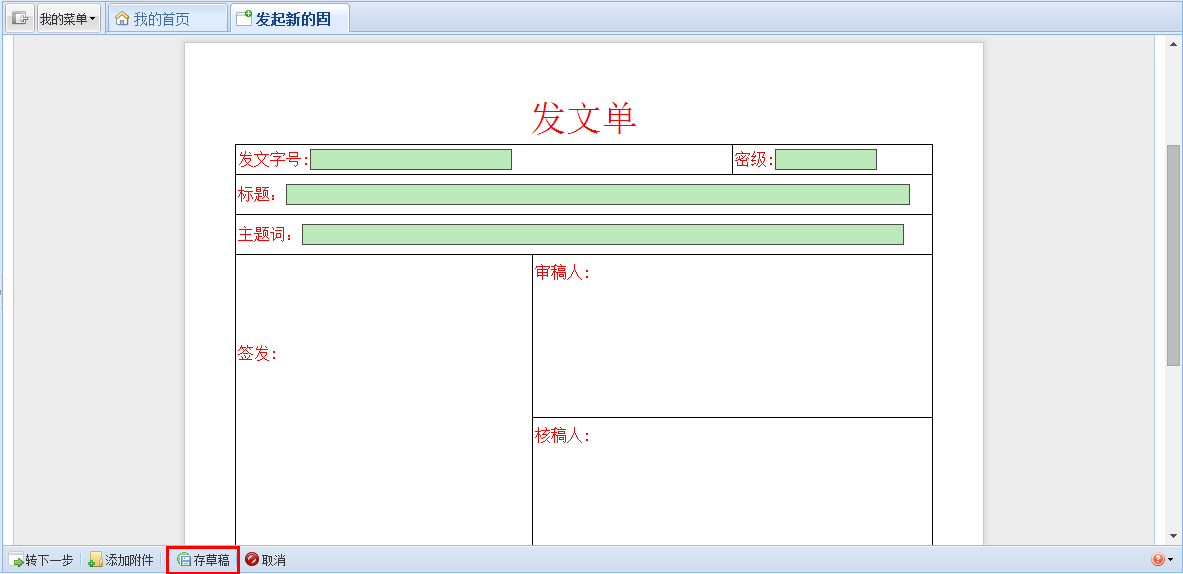
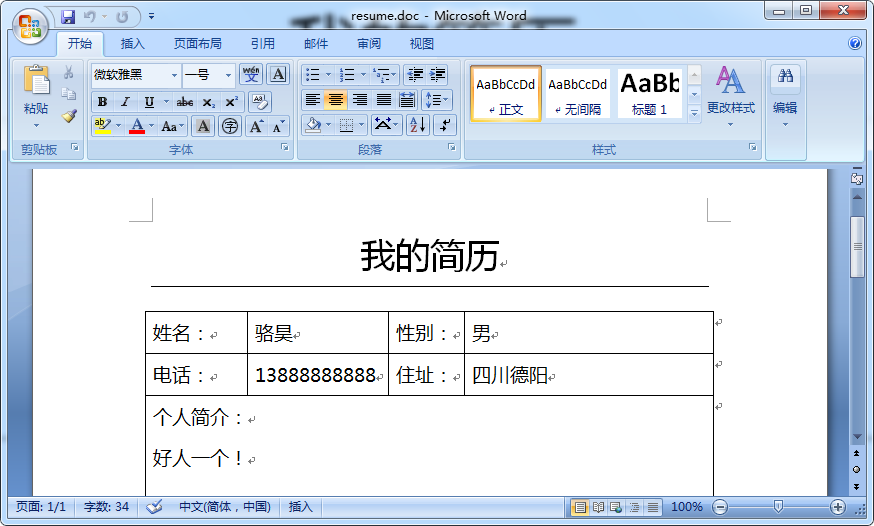
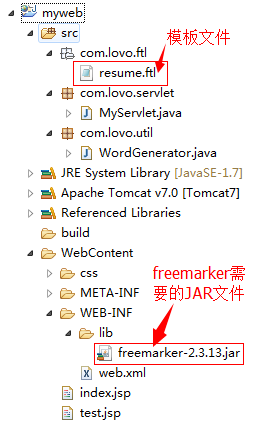
Web项目中生成Word文档的操作屡见不鲜，基于**[Java](http://lib.csdn.net/base/java" \t "_blank" \o "Java 知识库)**的解决方案也是很多的，包括使用[Jacob](http://sourceforge.net/projects/jacob-project/" \t "_blank)、[Apache POI](http://poi.apache.org/" \t "_blank)、Java2Word、[iText](http://itextpdf.com/" \t "_blank)等各种方式，其实在从Office 2003开始，就可以将Office文档转换成XML文件，这样只要将需要填入的内容放上${}占位符，就可以使用像Freemarker这样的模板引擎将出现占位符的地方替换成真实数据，这种方式较之其他的方案要更为简单。

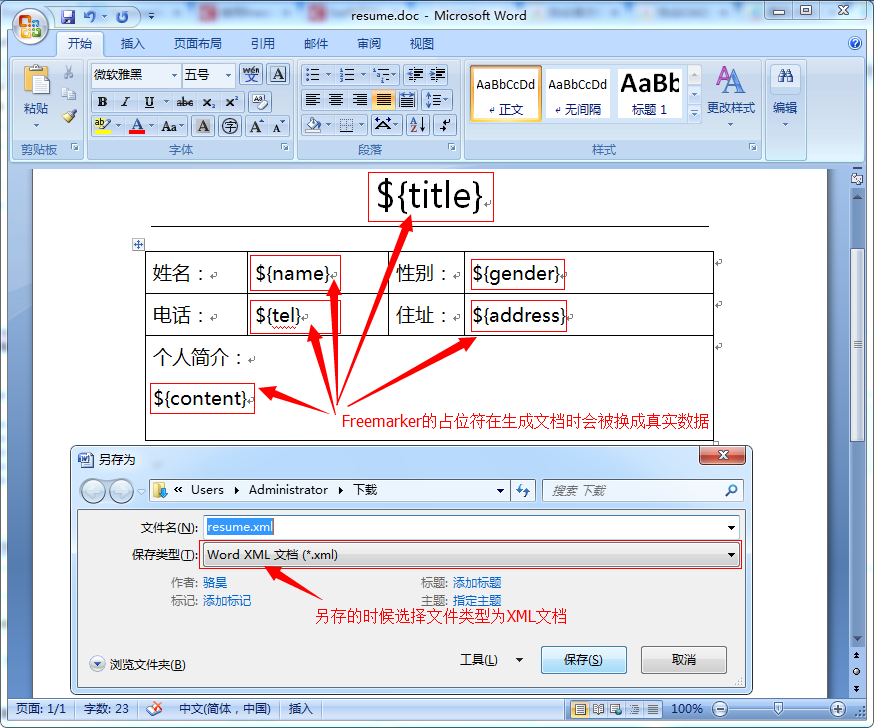
下面举一个简单的例子，比如在Web页面中填写个人简历，然后点击保存下载到本地，效果图如下所示。

打开下载的Word文件

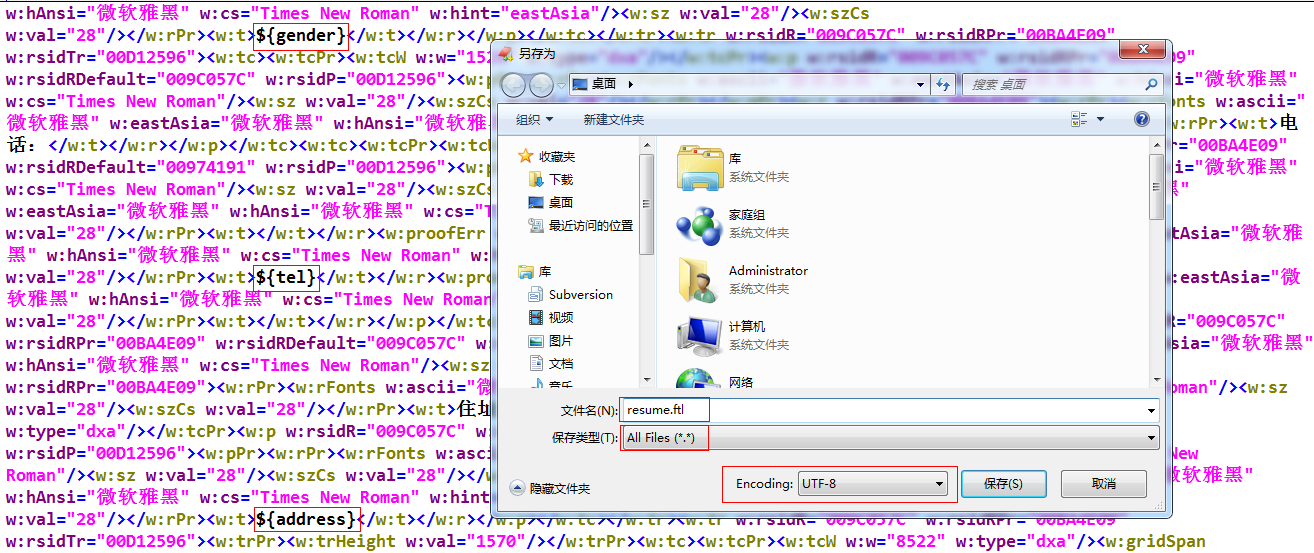
首先在Eclipse [**Java**](http://lib.csdn.net/base/java)EE版中新建一个Dynamic Web Project，项目结构如下图所示

需要向项目中加入freemarker的JAR文件，可以通过下面的链接获得Freemarker的最新版本：

<http://freemarker.org/freemarkerdownload.html>

模板文件resume.ftl是如何生成的呢，其实非常简单，将需要的Word文档做好之后，选择另存为XML文件，另存之后建议用Editplus、Notepad++、Sublime等工具打开查看一下，因为有的时候你写的占位符可能会被拆开，这样Freemarker就无法处理了。

打开XML文件看看吧，如果刚才你写的${title}、${name}被xml文件给拆散了，修改一下XML文件就OK了。

修改过后另存为resume.ftl模板文件，如下所示：接下来就是Servlet（也可以是Struts2的Action、**[spring](http://lib.csdn.net/base/javaee" \t "_blank" \o "Java EE知识库)** MVC的Controller等）和工具类WordGenerator的编写以及页面test.jsp的制作了，代码如下所示：

小服务的代码：

**[java]** [view plain](http://blog.csdn.net/jackfrued/article/details/39449021) [copy](http://blog.csdn.net/jackfrued/article/details/39449021)

1. **package** com.lovo.servlet;
3. **import** java.io.File;
4. **import** java.io.FileInputStream;
5. **import** java.io.IOException;
6. **import** java.io.InputStream;
7. **import** java.util.Enumeration;
8. **import** java.util.HashMap;
9. **import** java.util.Map;
11. **import** javax.servlet.ServletException;
12. **import** javax.servlet.ServletOutputStream;
13. **import** javax.servlet.annotation.WebServlet;
14. **import** javax.servlet.http.HttpServlet;
15. **import** javax.servlet.http.HttpServletRequest;
16. **import** javax.servlet.http.HttpServletResponse;
18. **import** com.lovo.util.WordGenerator;
20. /\*\*
21. \* Servlet implementation class MyServlet
22. \*/
23. @WebServlet("/saveDocServlet")
24. **public** **class** MyServlet **extends** HttpServlet {
25. **private** **static** **final** **long** serialVersionUID = 1L;
27. @Override
28. **protected** **void** service(HttpServletRequest req, HttpServletResponse resp)
29. **throws** ServletException, IOException {
30. req.setCharacterEncoding("utf-8");
31. Map<String, Object> map = **new** HashMap<String, Object>();
32. Enumeration<String> paramNames = req.getParameterNames();
33. // 通过循环将表单参数放入键值对映射中
34. **while**(paramNames.hasMoreElements()) {
35. String key = paramNames.nextElement();
36. String value = req.getParameter(key);
37. map.put(key, value);
38. }
40. // 提示：在调用工具类生成Word文档之前应当检查所有字段是否完整
41. // 否则Freemarker的模板殷勤在处理时可能会因为找不到值而报错 这里暂时忽略这个步骤了
42. File file = **null**;
43. InputStream fin = **null**;
44. ServletOutputStream out = **null**;
45. **try** {
46. // 调用工具类WordGenerator的createDoc方法生成Word文档
47. file = WordGenerator.createDoc(map, "resume");
48. fin = **new** FileInputStream(file);
50. resp.setCharacterEncoding("utf-8");
51. resp.setContentType("application/msword");
52. // 设置浏览器以下载的方式处理该文件默认名为resume.doc
53. resp.addHeader("Content-Disposition", "attachment;filename=resume.doc");
55. out = resp.getOutputStream();
56. **byte**[] buffer = **new** **byte**[512];  // 缓冲区
57. **int** bytesToRead = -1;
58. // 通过循环将读入的Word文件的内容输出到浏览器中
59. **while**((bytesToRead = fin.read(buffer)) != -1) {
60. out.write(buffer, 0, bytesToRead);
61. }
62. } **finally** {
63. **if**(fin != **null**) fin.close();
64. **if**(out != **null**) out.close();
65. **if**(file != **null**) file.delete(); // 删除临时文件
66. }
67. }
69. }

工具类的代码：

**[java]** [view plain](http://blog.csdn.net/jackfrued/article/details/39449021) [copy](http://blog.csdn.net/jackfrued/article/details/39449021)

1. **package** com.lovo.util;
3. **import** java.io.File;
4. **import** java.io.FileOutputStream;
5. **import** java.io.IOException;
6. **import** java.io.OutputStreamWriter;
7. **import** java.io.Writer;
8. **import** java.util.HashMap;
9. **import** java.util.Map;
11. **import** freemarker.template.Configuration;
12. **import** freemarker.template.Template;
14. **public** **class** WordGenerator {
15. **private** **static** Configuration configuration = **null**;
16. **private** **static** Map<String, Template> allTemplates = **null**;
18. **static** {
19. configuration = **new** Configuration();
20. configuration.setDefaultEncoding("utf-8");
21. configuration.setClassForTemplateLoading(WordGenerator.**class**, "/com/lovo/ftl");
22. allTemplates = **new** HashMap<>();   // Java 7 钻石语法
23. **try** {
24. allTemplates.put("resume", configuration.getTemplate("resume.ftl"));
25. } **catch** (IOException e) {
26. e.printStackTrace();
27. **throw** **new** RuntimeException(e);
28. }
29. }
31. **private** WordGenerator() {
32. **throw** **new** AssertionError();
33. }
35. **public** **static** File createDoc(Map<?, ?> dataMap, String type) {
36. String name = "temp" + (**int**) (Math.random() \* 100000) + ".doc";
37. File f = **new** File(name);
38. Template t = allTemplates.get(type);
39. **try** {
40. // 这个地方不能使用FileWriter因为需要指定编码类型否则生成的Word文档会因为有无法识别的编码而无法打开
41. Writer w = **new** OutputStreamWriter(**new** FileOutputStream(f), "utf-8");
42. t.process(dataMap, w);
43. w.close();
44. } **catch** (Exception ex) {
45. ex.printStackTrace();
46. **throw** **new** RuntimeException(ex);
47. }
48. **return** f;
49. }
51. }

JSP页面的代码：

**[html]** [view plain](http://blog.csdn.net/jackfrued/article/details/39449021) [copy](http://blog.csdn.net/jackfrued/article/details/39449021)

1. **<**%@ page pageEncoding="UTF-8"%**>**
2. <!DOCTYPE html**>**
3. **<html>**
4. **<head>**
5. **<meta** charset="UTF-8" **/>**
6. **<title>**Document**</title>**
7. **<style** type="text/css"**>**
8. \* { font-family: "微软雅黑"; }
9. .textField { border:none; border-bottom: 1px solid gray; text-align: center; }
10. #file { border:1px solid black; width: 80%; margin:0 auto; }
11. h1 input{ font-size:72px; }
12. td textarea { font-size: 14px; }
13. .key { width:125px; font-size:20px; }
14. **</style>**
15. **</head>**
16. **<body>**
17. **<form** action="saveDocServlet" method="post"**>**
18. **<div** id="file" align="center"**>**
19. **<h1><input** type="text" name="title" class="textField" value="我的简历"**/></h1>**
20. **<hr/>**
21. **<table>**
22. **<tr>**
23. **<td** class="key"**>**姓名：**</td>**
24. **<td><input** type="text" name="name" class="textField"**/></td>**
25. **<td** class="key"**>**性别：**</td>**
26. **<td>**
27. **<input** type="radio" name="gender" value="男" checked**/>**男
28. **<input** type="radio" name="gender" value="女" **/>**女
29. **</td>**
30. **</tr>**
31. **<tr>**
32. **<td** class="key"**>**联系电话：**</td>**
33. **<td><input** type="text" name="tel" class="textField"**/></td>**
34. **<td** class="key"**>**家庭住址：**</td>**
35. **<td><input** type="text" name="address" class="textField"**/></td>**
36. **</tr>**
37. **<tr>**
38. **<td** colspan="4" class="key"**>**个人简介：**</td>**
39. **</tr>**
40. **<tr>**
41. **<td** colspan="4"**>**
42. **<textarea** rows="10" cols="100" name="content"**></textarea>**
43. **</td>**
44. **</tr>**
45. **</table>**
46. **</div>**
47. **<div** align="center" style="margin-top:15px;"**>**
48. **<input** type="submit" value="保存Word文档" **/>**
49. **</div>**
50. **</form>**
51. **</body>**
52. **</html>**

说明：小服务是使用注解进行配置的，因此你的服务器需要支持Servlet 3规范，我使用的服务器是Tomcat 7.0.52。如果你的服务器不支持Servlet 3规范那就使用web.xml来配置你的小服务吧，其他地方没有不同。如果你不熟悉Servlet 3规范的新特性，可以阅读CSDN上另一篇文章，链接如下所示：

<http://blog.csdn.net/zhongweijian/article/details/8279650>

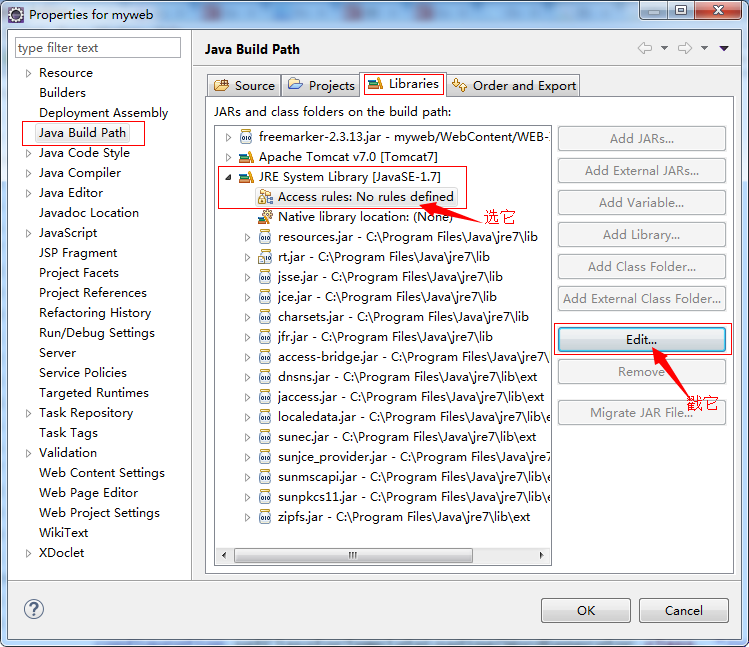
此外，如果你希望在Word文档中插入图片，可以把Word另存为的XML文件中代表图片的那个很长的字符串（[BASE64编码](http://baike.baidu.com/view/469071.htm" \t "_blank)的字符串）换成一个占位符，在将要插入Word文档的图片对象转换成BASE64编码的字符串，用该字符串替换掉占位符就可以了，示意图和代码如下所示：将图片转换成BASE64字符串的代码如下所示：

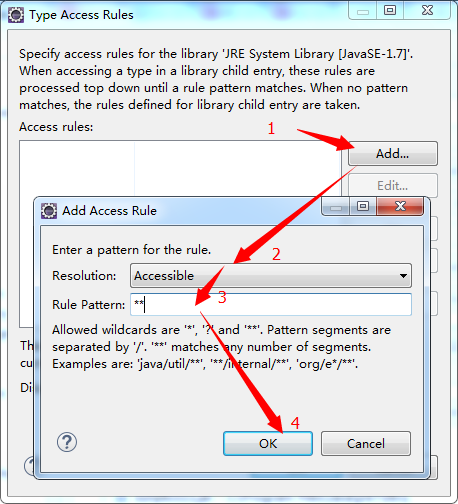
**[java]** [view plain](http://blog.csdn.net/jackfrued/article/details/39449021) [copy](http://blog.csdn.net/jackfrued/article/details/39449021)

1. **public** **static** String getImageString(String filename) **throws** IOException {
2. InputStream in = **null**;
3. **byte**[] data = **null**;
4. **try** {
5. in = **new** FileInputStream(filename);
6. data = **new** **byte**[in.available()];
7. in.read(data);
8. in.close();
9. } **catch** (IOException e) {
10. **throw** e;
11. } **finally** {
12. **if**(in != **null**) in.close();
13. }
14. BASE64Encoder encoder = **new** BASE64Encoder();
15. **return** data != **null** ? encoder.encode(data) : "";
16. }

注意：这里使用的BASE64Encoder类在sun.misc包下，rt.jar中有这个类，但是却无法直接使用，需要修改访问权限，在Eclipse中可以这样修改。

在项目上点右键选择Properties菜单项进入如下图所示的界面：





这样设置后就可以使用BASE64Encoder类了，在项目中调用getImageString方法指定要插入的图片的完整文件名（带路径的文件名），该方法返回的字符串就是将图片处理成BASE64编码后的字符串。但愿你按照上面的步骤一次成功！