

使用Maven编译项目遇到——“maven编码gbk的不可映射字符”解决办法 - 孤傲苍狼

一、问题描述

今天在MyEclipse中使用Maven编译项目源代码时，结果如下了如下的错误

```
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:2.5.1:compile (default-compile) on project ssh: Compilation failure: Compilation failure:
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\service\impl\MenuServiceImpl.java:[42,10] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\service\impl\MenuServiceImpl.java:[42,16] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\service\impl\MenuServiceImpl.java:[45,24] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\service\impl\MenuServiceImpl.java:[56,45] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[8,10] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[10,21] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[12,21] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[77,9] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[82,30] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[82,46] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[87,52] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[108,17] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[87,38] 未结束的字符串字面值
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[87,55] 需要 ';'
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[119,1] 非法的表达式开始
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[119,9] 非法的表达式开始
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[119,22] 需要 ';'
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[119,34] 需要 ".class"
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[119,37] 需要 ';'
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\util\Encrypt.java:[121,1] 进行语法规析时已到达文件结尾
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\MenuAction.java:[36,14] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[42,20] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[56,20] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[71,20] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[87,19] 编码GBK 的不可映射字符
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[42,12] 未结束的字符串字面值
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[42,23] 需要 ';'
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[43,4] "catch" 不带有 "try"
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[43,20] 需要 ';'
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[43,10] 不是语句
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[43,22] 需要 ';'
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[39,2] "try" 不带有 "catch" 或 "finally"
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[51,1] 非法的表达式开始
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[51,8] 非法的表达式开始
[ERROR] D:\MyEclipse8.5\workspace\Struts2AnnotationMavenProject\src\main\java\me\gacl\action\UserAction.java:[51,16] 需要 ';'

```

百思不得其解啊，java源码在MyEclipse中显示是没有任何错误的，可是执行“maven install”命令编译项目时就会出现上述的错误，导致项目编译失败。这个问题还是第一次遇到，还好在这里<http://capslk.iteye.com/blog/1419958>看到了解决问题的办法。

二、解决办法

解决这个问题的思路：在maven的编译插件中声明正确的字符集编码——编译使用的字符集编码与代码文件使用的字符集编码一致！！

安装系统之后，一般中文系统默认字符集是GBK。我们安装的软件一般都继承使用操作系统的默认字符集。所以当在中文XP或者win7系统开发，在使用maven (mvn compile) 编译项目的时候，就会出现“编码 gbk 的不可映射字符”：这个是由于代码使用的UTF-8，而maven编译的时候使用的GBK的缘故。可以通过修改项目的pom文件，可以告诉maven这个项目使用UTF-8编码来编译。在项目的pom.xml文件中添加下面的配置：

```
1 <!-- 指明编译源代码时使用的字符编码，maven编译的时候默认使用的GBK编码， 通过project.build.sourceEncoding
属性设置字符编码，告诉maven这个项目使用UTF-8来编译 -->
2     <properties>
3         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
4     </properties>
```

或者在pom.xml的/project/build/plugins/下的编译插件声明 中加入下面的配置:

```
1 <encoding>utf8</encoding>
```

即：



```
1 <plugin>
2   <artifactId>maven-compiler-plugin</artifactId>
3   <configuration>
4     <source>1.6</source>
5     <target>1.6</target>
6     <!-- 指明编译源代码时使用的字符编码，
7       maven编译的时候默认使用的GBK编码，
8       通过encoding属性设置字符编码，
9       告诉maven这个项目使用UTF-8来编译 -->
10    <encoding>utf8</encoding>
11  </configuration>
12 </plugin>
```

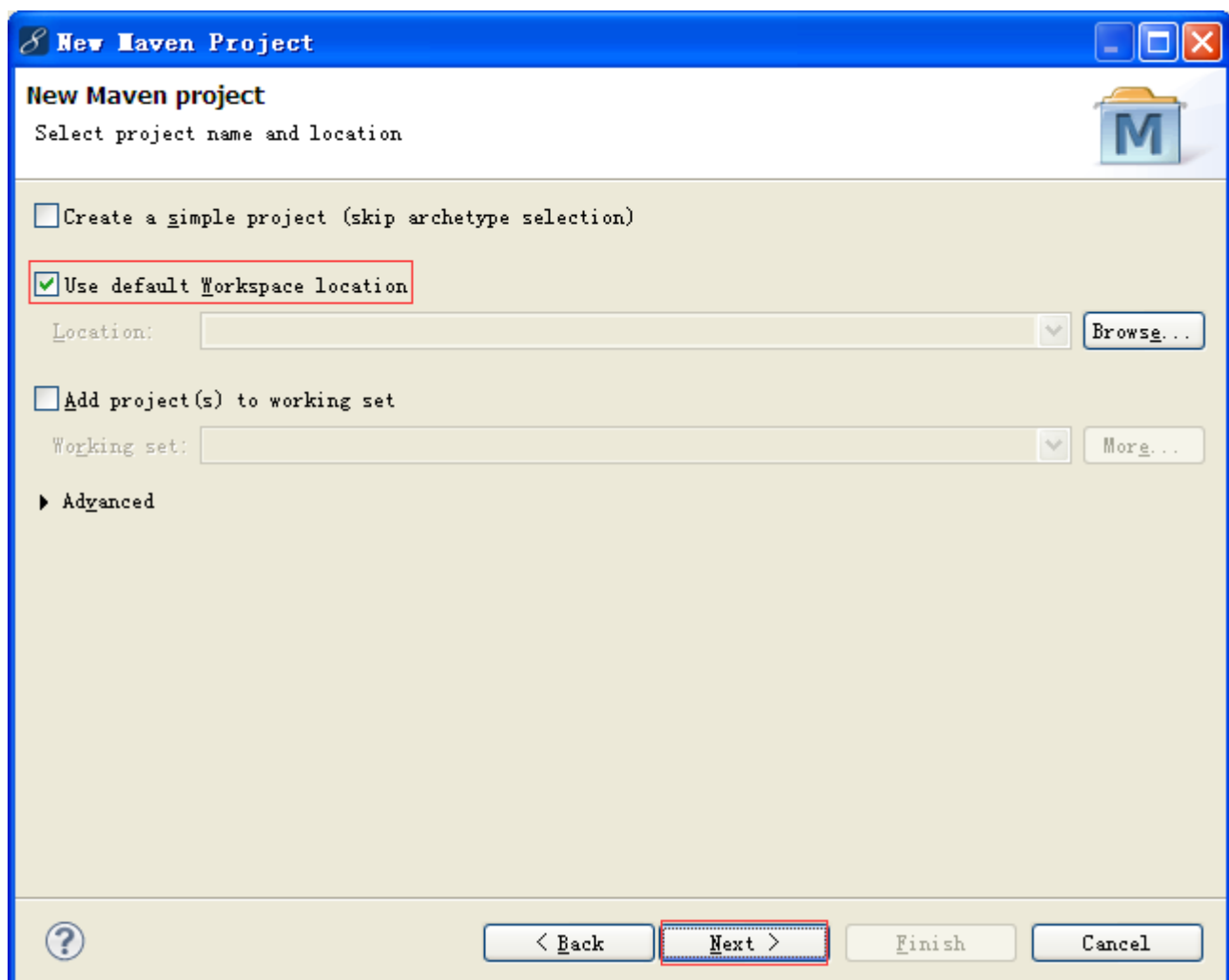
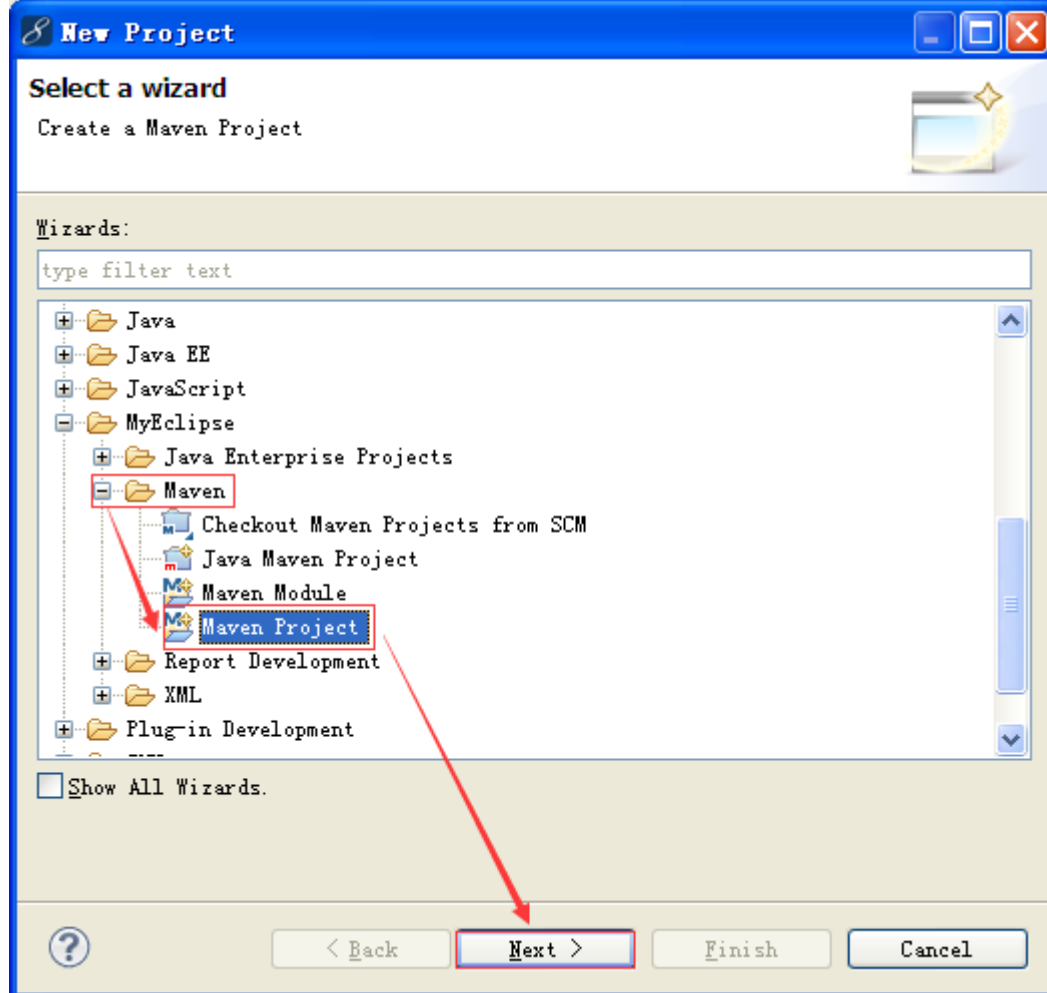


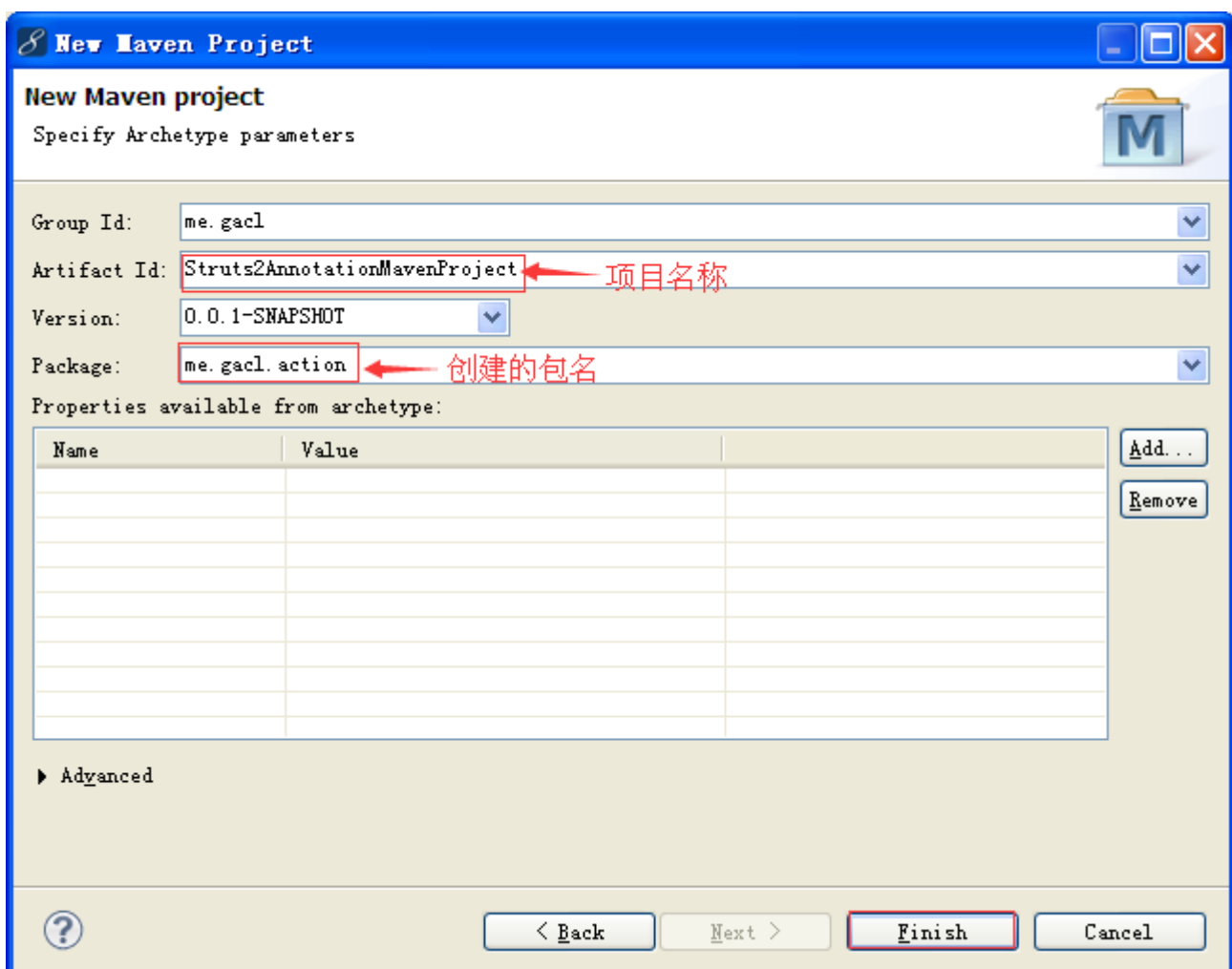
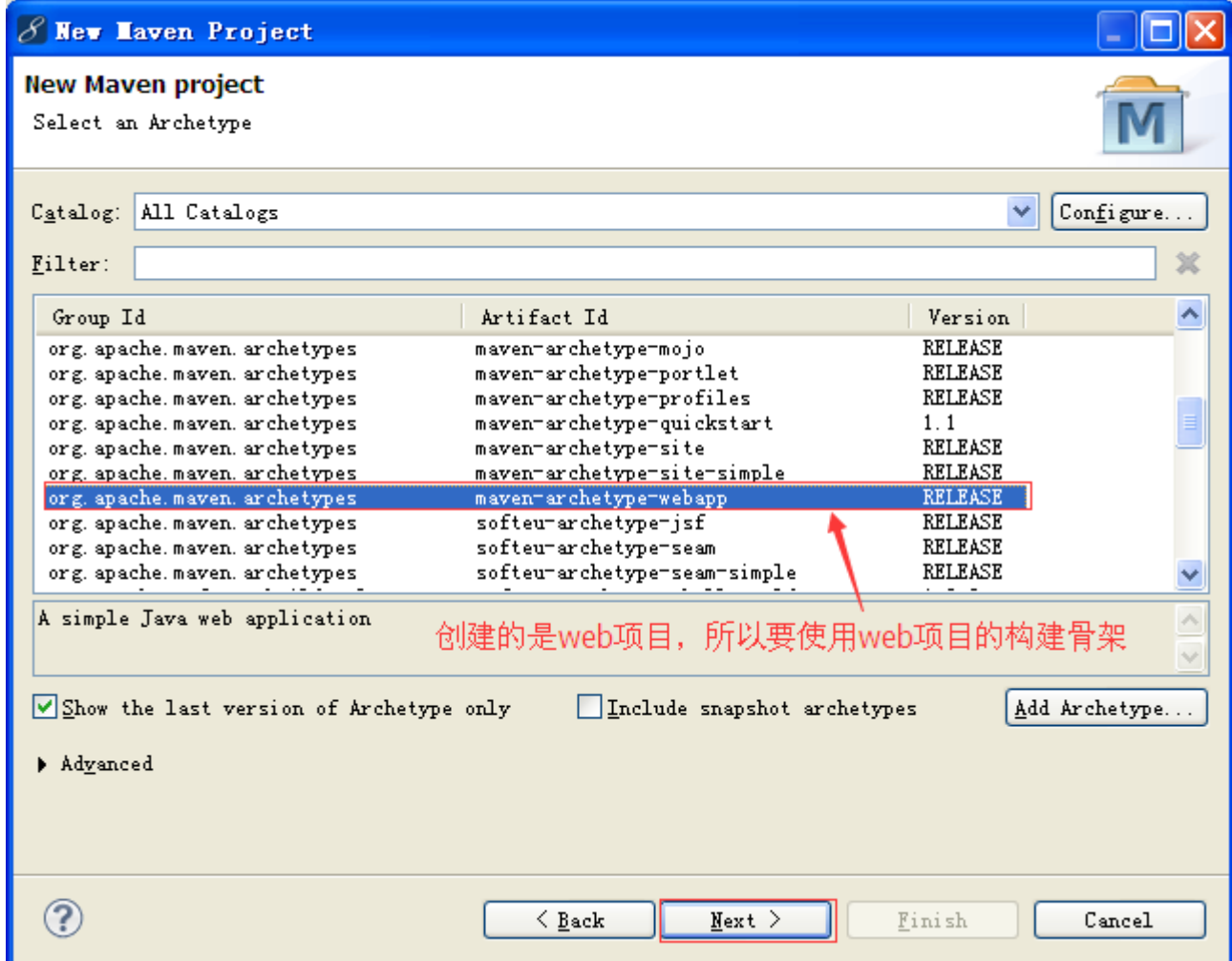
这两种方案都可以解决上述的问题，可以根据个人习惯来设置，如下图所示方式是采用配置<project.build.sourceEncoding>属性来指明编译采用的字符编码。

```
ssh/pom.xml X
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>me.gacl</groupId>
5   <artifactId>ssh</artifactId>
6   <packaging>war</packaging>
7   <version>0.0.1-SNAPSHOT</version>
8   <name>ssh Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10
11   <!-- 指明编译源代码时使用的字符编码， maven编译的时候默认使用的GBK编码，
12     通过project.build.sourceEncoding属性设置字符编码，告诉maven这个项目使用UTF-8来编译 -->
13   <properties>
14     <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15   </properties>
16   <dependencies>
17
18   <build>
19     <finalName>ssh</finalName>
20     <plugins>
21       <plugin>
22         <artifactId>maven-war-plugin</artifactId>
23       </plugin>
24       <plugin>
25         <artifactId>maven-compiler-plugin</artifactId>
26         <configuration>
27           <source>1.6</source>
28           <target>1.6</target>
29           <!-- 指明编译源代码时使用的字符编码，
30             maven编译的时候默认使用的GBK编码，
31             通过encoding属性设置字符编码，
32             告诉maven这个项目使用UTF-8来编译 -->
33           <!-- <encoding>utf8</encoding> -->
34         </configuration>
35       </plugin>
36     </plugins>
37   </build>
38 </project>
```

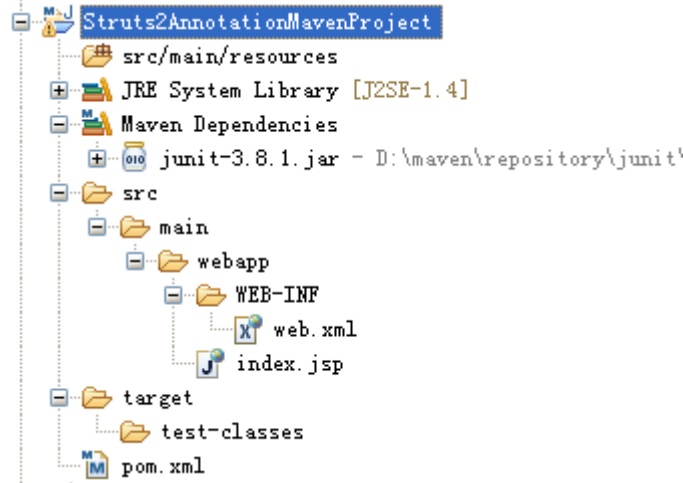
今天被这个问题折腾够久了，还好及时找到了解决办法，在此记录一下解决方案。

一、创建基于Maven的Web项目

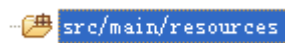




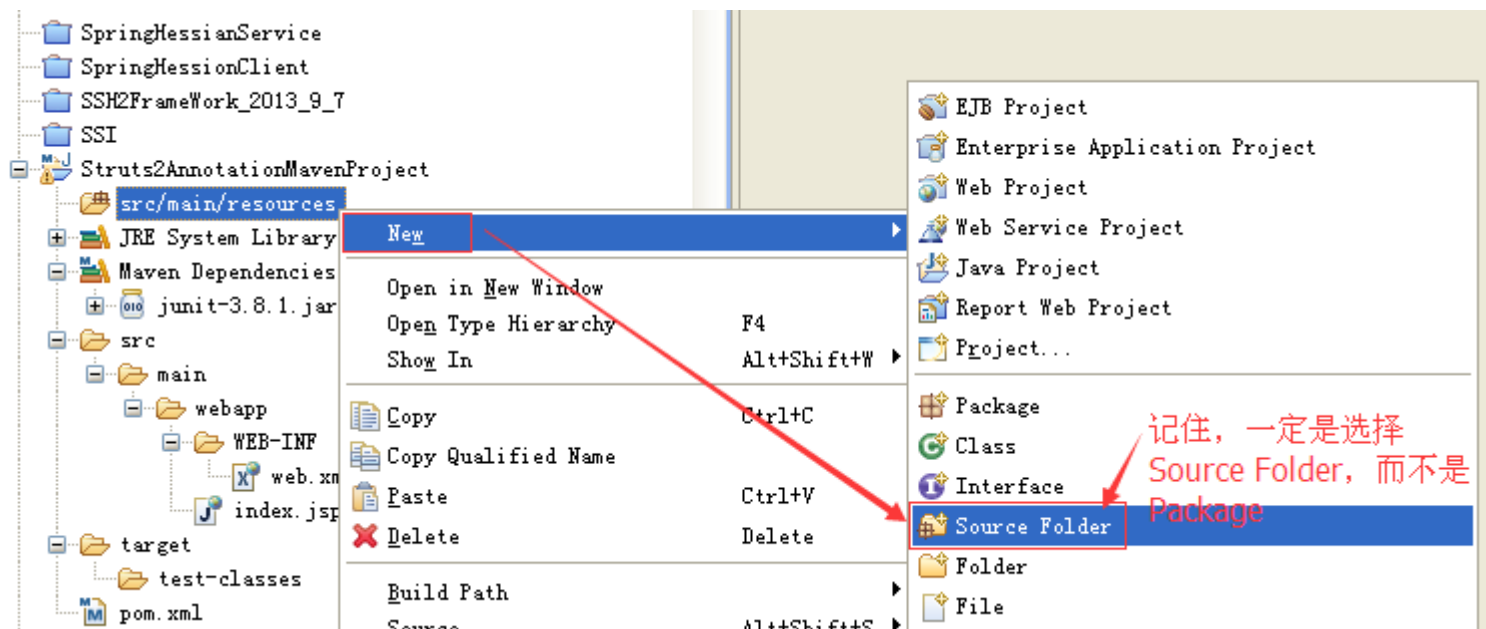
我使用的是MyEclipse8.5的版本，创建好的Web项目如下所示：

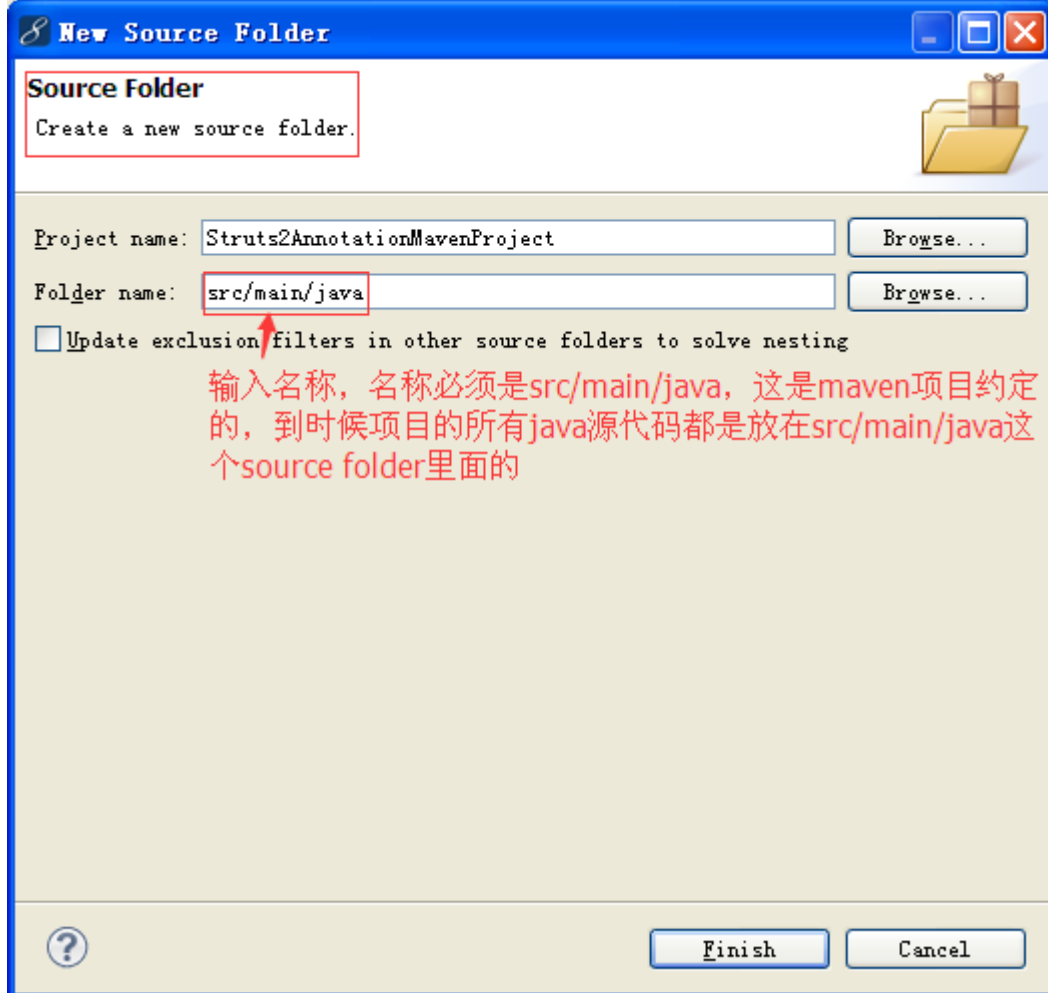


我们知道，一个标准的Maven项目是必须包括【src/main/java】，【src/main/resources】，【src/test/java】，【src/test/resources】这四个Source Folder的，而创建好的项目当中只有一个

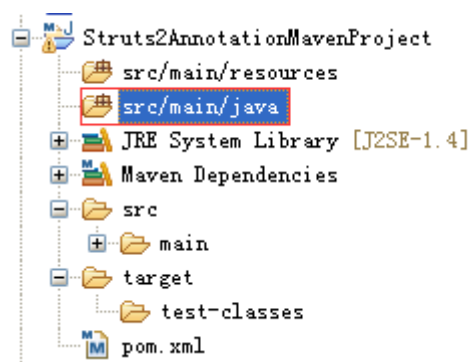


(不懂为啥MyEclipse8.5没有帮我生成【src/main/java】)，所以我们还需要手动创建剩下的【src/main/java】，【src/test/java】，【src/test/resources】这三个Source Folder，以创建【src/main/java】为例，具体步骤如下：

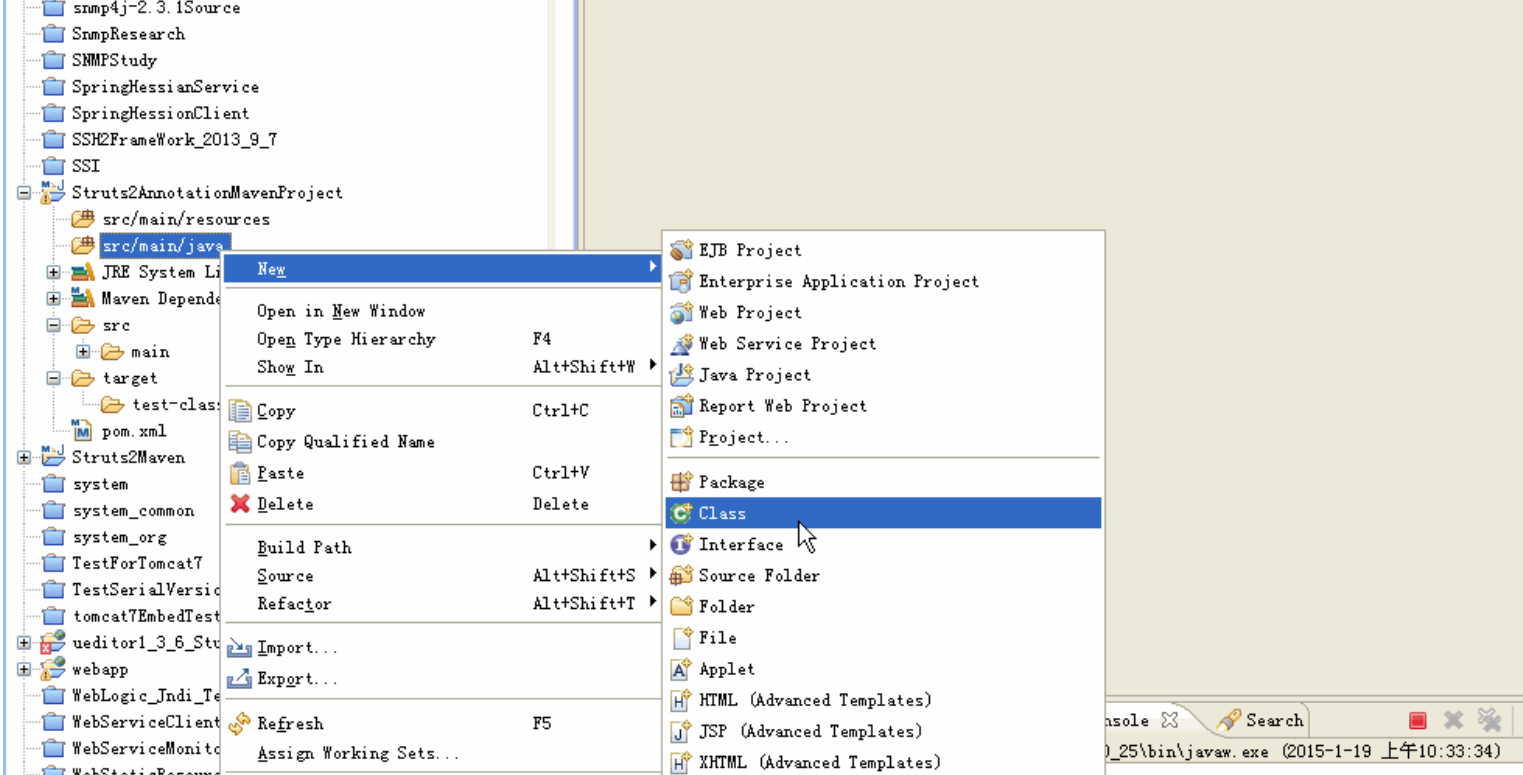




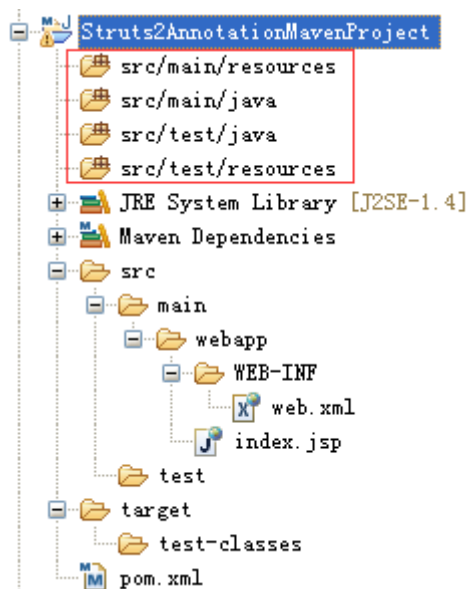
点击【Finish】按钮完成创建，如下图所示：



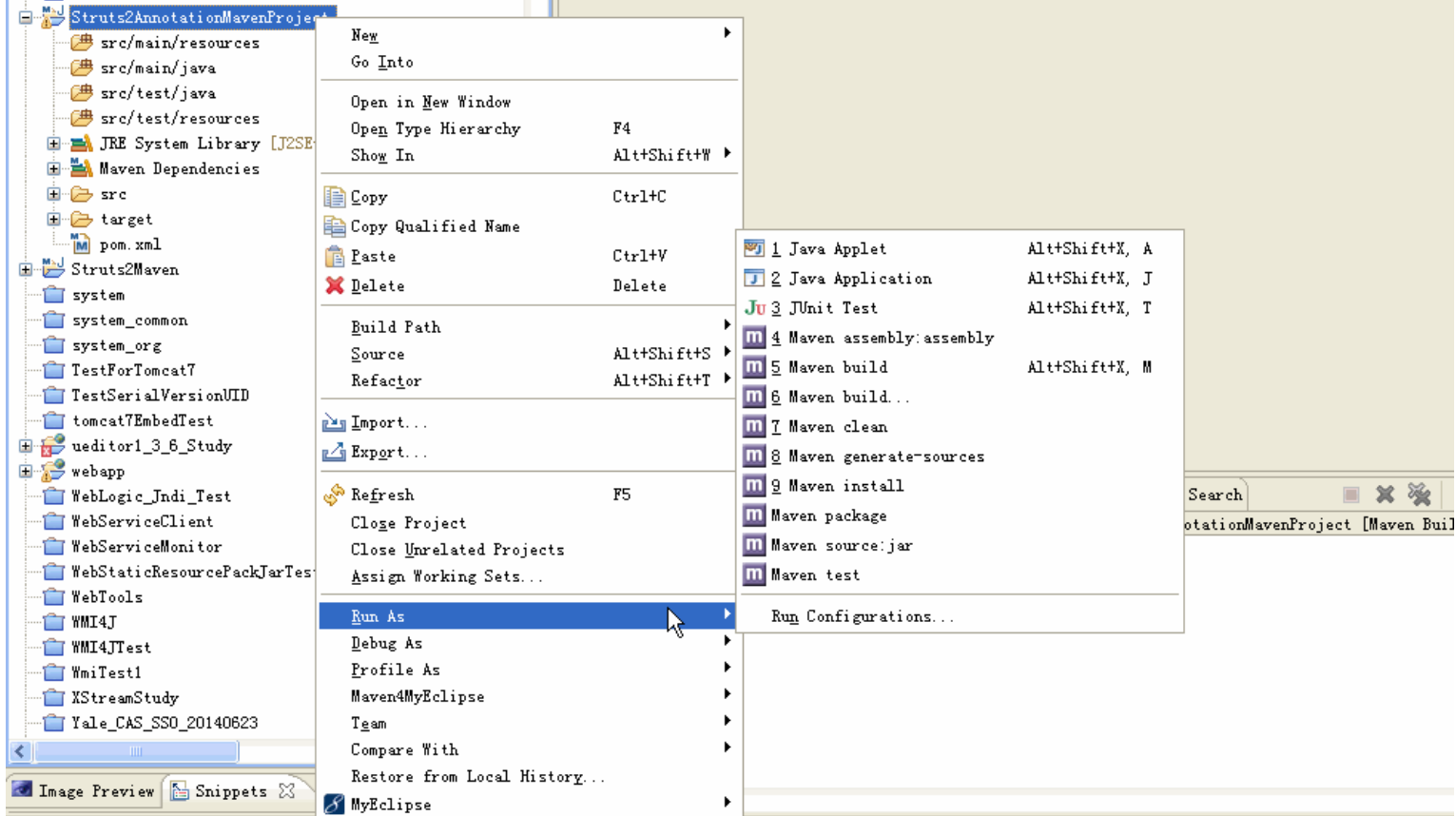
【src/test/java】，【src/test/resources】也是采用相同的方式进行创建，这里用一张动态的gif动画演示一下创建过程，如下图所示：



最终效果如下：



这样我们的【Struts2AnnotationMavenProject】项目才是一个标准的Maven项目，我们可以使用Maven来构建一下【Struts2AnnotationMavenProject】项目，看看能否正常构建成功，如下图所示：



从运行结果显示，项目可以正常构建成功。

二、搭建Struts2的开发环境

2.1、添加Struts2框架的核心jar包

由于我们是使用Maven管理项目中的jar包的，所以我们需要在pom.xml文件中添加Struts2框架的核心jar包的描述。

编辑pom.xml文件，添加添加Struts2框架的核心jar包的描述，如下：



```

1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
2   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
3   <modelVersion>4.0.0</modelVersion>
4   <groupId>me.gacl</groupId>
5   <artifactId>Struts2AnnotationMavenProject</artifactId>
6   <packaging>war</packaging>
7   <version>0.0.1-SNAPSHOT</version>
8   <name>Struts2AnnotationMavenProject Maven Webapp</name>
9   <url>http://maven.apache.org</url>
10  <dependencies>
11    <dependency>
12      <groupId>junit</groupId>
13      <artifactId>junit</artifactId>
14      <version>3.8.1</version>
15      <scope>test</scope>
16    </dependency>
17    <!-- Struts2的核心包 -->
18    <dependency>
19      <groupId>org.apache.struts</groupId>

```



```

20     <artifactId>struts2-core</artifactId>
21     <version>2.3.16</version>
22 </dependency>
23 </dependencies>
24 <build>
25     <finalName>Struts2AnnotationMavenProject</finalName>
26 </build>
27 </project>

```



pom.xml文件中标红的部分就是我们添加的Struts2框架的核心jar包的描述，保存pom.xml文件，此时Maven就会自动帮我们吧struts2-core这个jar包依赖的其他相关jar包导入到我们的Web项目当中，如下图所示：

我们只在pom.xml文件中添加对struts2-core.2.3.16这个jar包的描述，但是Maven却自动帮我们把它的相关的依赖jar包导入到项目当中，这就是使用Maven管理jar包的好处

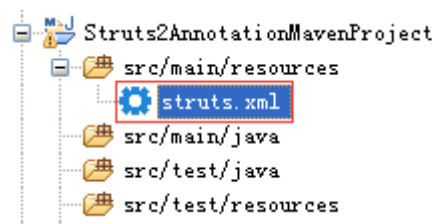
```

10 <dependencies>
11 <dependency>
12     <groupId>junit</groupId>
13     <artifactId>junit</artifactId>
14     <version>3.8.1</version>
15     <scope>test</scope>
16 </dependency>
17 <!-- Struts2的核心包 -->
18 <dependency>
19     <groupId>org.apache.struts</groupId>
20     <artifactId>struts2-core</artifactId>
21     <version>2.3.16</version>
22 </dependency>
23 </dependencies>
24 <build>
25     <finalName>Struts2AnnotationMavenProject</finalName>
26 </build>
27 </project>

```

2.2、添加Struts2框架的配置文件struts.xml

Maven约定，web项目开发中的使用到的配置文件都要放到【src/main/resources】这个Source Folder下，如下图所示：



编辑struts.xml文件，添加常用的配置项，配置信息如下：



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
"http://struts.apache.org/dtds/struts-2.3.dtd">
3 <struts>
4     <!-- 所有匹配*.action的请求都由struts2处理 -->
5     <constant name="struts.action.extension" value="action" />
6     <!-- 是否启用开发模式 -->
7     <constant name="struts.devMode" value="true" />
8     <!-- struts配置文件改动后，是否重新加载 -->
9     <constant name="struts.configuration.xml.reload" value="true" />
10    <!-- 设置浏览器是否缓存静态内容 -->

```

```

11 <constant name="struts.serve.static.browserCache" value="false" />
12 <!-- 请求参数的编码方式 -->
13 <constant name="struts.i18n.encoding" value="utf-8" />
14 <!-- 每次HTTP请求系统都重新加载资源文件，有助于开发 -->
15 <constant name="struts.i18n.reload" value="true" />
16 <!-- 文件上传最大值 -->
17 <constant name="struts.multipart.maxSize" value="104857600" />
18 <!-- 让struts2支持动态方法调用 -->
19 <constant name="struts.enable.DynamicMethodInvocation" value="true" />
20 <!-- Action名称中是否还是用斜线 -->
21 <constant name="struts.enable.SlashesInActionNames" value="false" />
22 <!-- 允许标签中使用表达式语法 -->
23 <constant name="struts.tag.altSyntax" value="true" />
24 <!-- 对于WebLogic, Orion, OC4J此属性应该设置成true -->
25 <constant name="struts.dispatcher.parametersWorkaround" value="false" />
26
27 <package name="basePackage" extends="struts-default">
28
29
30 </package>
31
32 </struts>

```



2.3、convention-plugin和config-browser-plugin插件介绍

以前用struts2框架开发项目时，每次编写好一个Action，就需要在struts.xml文件中配置Action，而convention-plugin这个插件的出现出现后，就不再需要在struts.xml文件中配置Action了，convention-plugin提供了一种非常方便的注解方式来配置Action类。

convention-plugin采用“约定大于配置”的思想，只要我们遵守约定，完全可以少写配置甚至不写配置；config-browser-plugin插件则用于方便的浏览项目中的所有action及其与 jsp view的映射。这二个插件结合起来学习，能很方便的搞定struts2中各种复杂的action-view映射需求，所以现在使用Struts2框架开发Web应用时，一般都会配合这两个插件一起使用。

2.3.1、convention-plugin和config-browser-plugin插件使用 为了方便使用Struts2框架，我们最好在项目中配合convention-plugin、 config-browser-plugin这两个插件一起使用，在maven项目的pom.xml中加上这两个插件的jar包描述，如下：



```

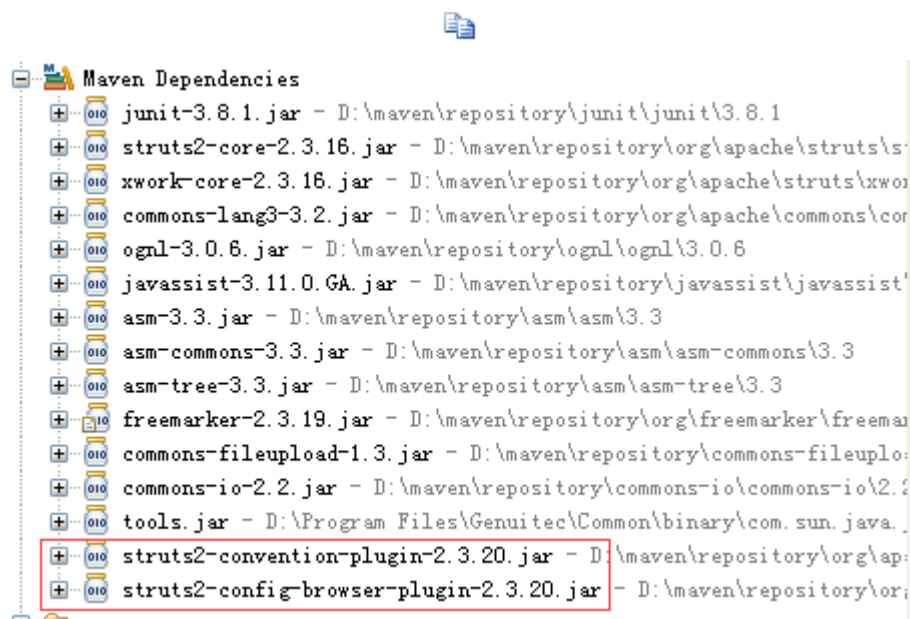
1 <!-- convention-plugin插件，使用了这个插件之后，就可以采用注解的方式配置Action -->
2 <dependency>
3     <groupId>org.apache.struts</groupId>
4     <artifactId>struts2-convention-plugin</artifactId>
5     <version>2.3.20</version>
6 </dependency>
7 <!--config-browser-plugin插件，使用了这个插件之后，就可以很方便的浏览项目中的所有action及其与 jsp
view的映射-->
8 <dependency>
9     <groupId>org.apache.struts</groupId>

```

```

10     <artifactId>struts2-config-browser-plugin</artifactId>
11     <version>2.3.20</version>
12 </dependency>

```



2.4、在web.xml文件中配置Struts2的核心过滤器

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="2.5"
3     xmlns="http://java.sun.com/xml/ns/javaee"
4     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5     xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
6     http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
7
8     <display-name>Archetype Created Web Application</display-name>
9
10    <!-- add struts2 configuation -->
11    <filter>
12        <description>配置struts2的核心过滤器</description>
13        <filter-name>struts2</filter-name>
14        <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-
15        class>
16    </filter>
17
18    <filter-mapping>
19        <filter-name>struts2</filter-name>
20        <url-pattern>*.action</url-pattern>
21    </filter-mapping>
22    <!-- end add struts2 configuration-->
23 </web-app>

```

2.5、测试Struts2框架是否搭建成功

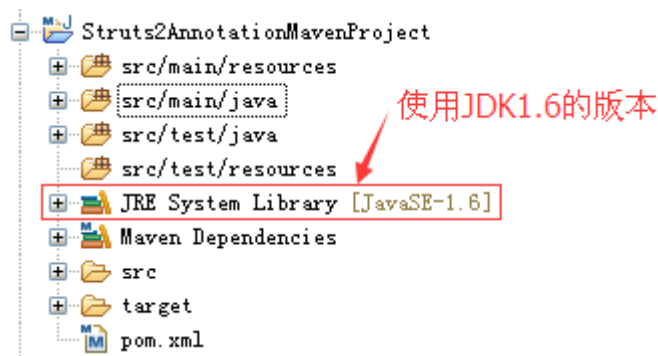
编写一个TestAction类，用于测试



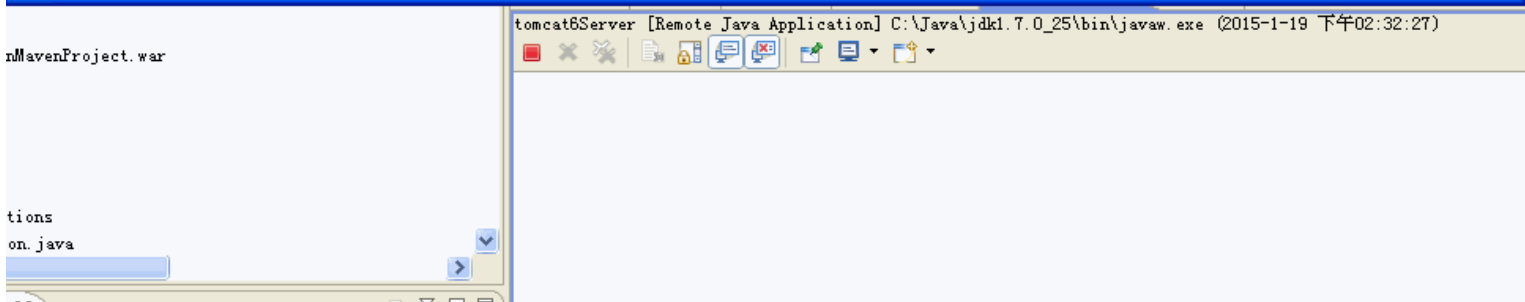
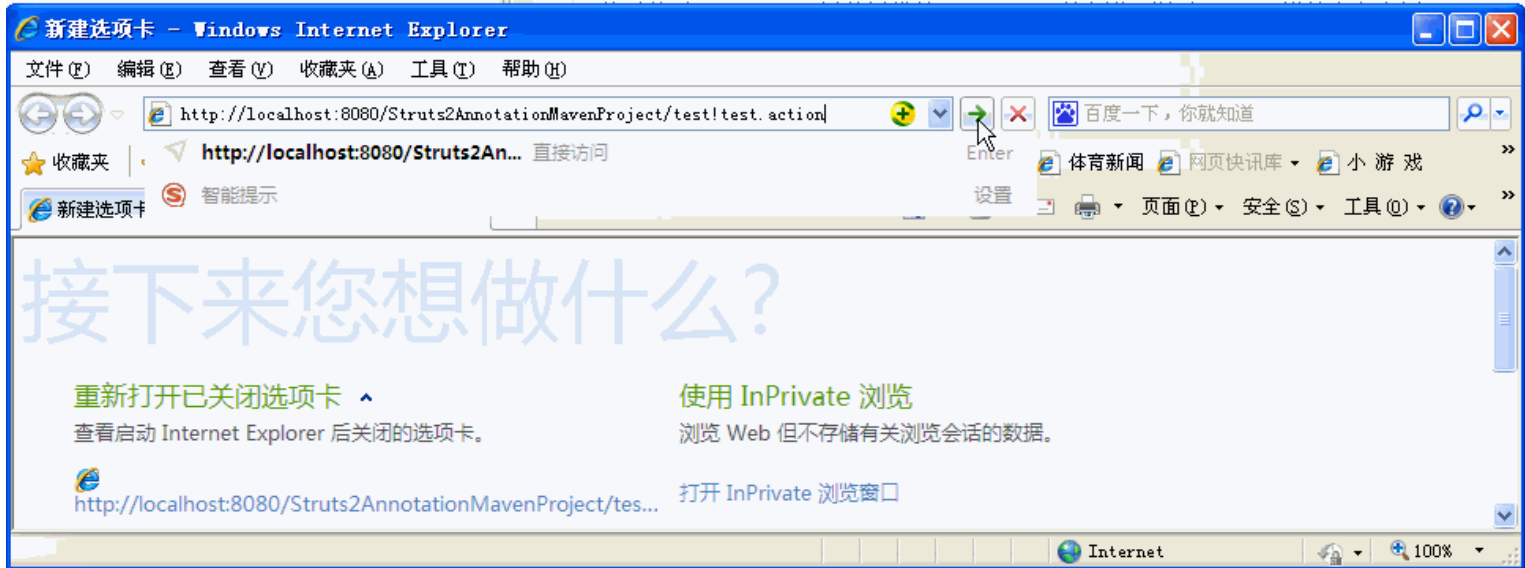
```
1 package me.gacl.action;
2
3 import org.apache.struts2.convention.annotation.Action;
4 import org.apache.struts2.convention.annotation.Namespace;
5
6 @ParentPackage("basePackage")
7 //使用convention插件提供的@Action注解将一个普通Java类标识为可以处理用户请求的Action类
8 @Action
9 //使用使用convention插件提供的@Namespace注解指明这个Action类的命名空间
10 @Namespace("/")
11 public class TestAction {
12     /**
13      * test方法的访问方式: http://localhost:8080/Struts2AnnotationMavenProject/test!test
14      * MethodName: test
15      * Description:
16      * @author xudp
17      */
18     public void test() {
19         System.out.println("调用了TestAction里面的test方法");
20     }
21 }
```



在将类使用@Action注解标识时发现@Action注解必须使用JDK1.6才行，所以我修改了JDK的版本，改成使用JDK1.6的，如下图所示：



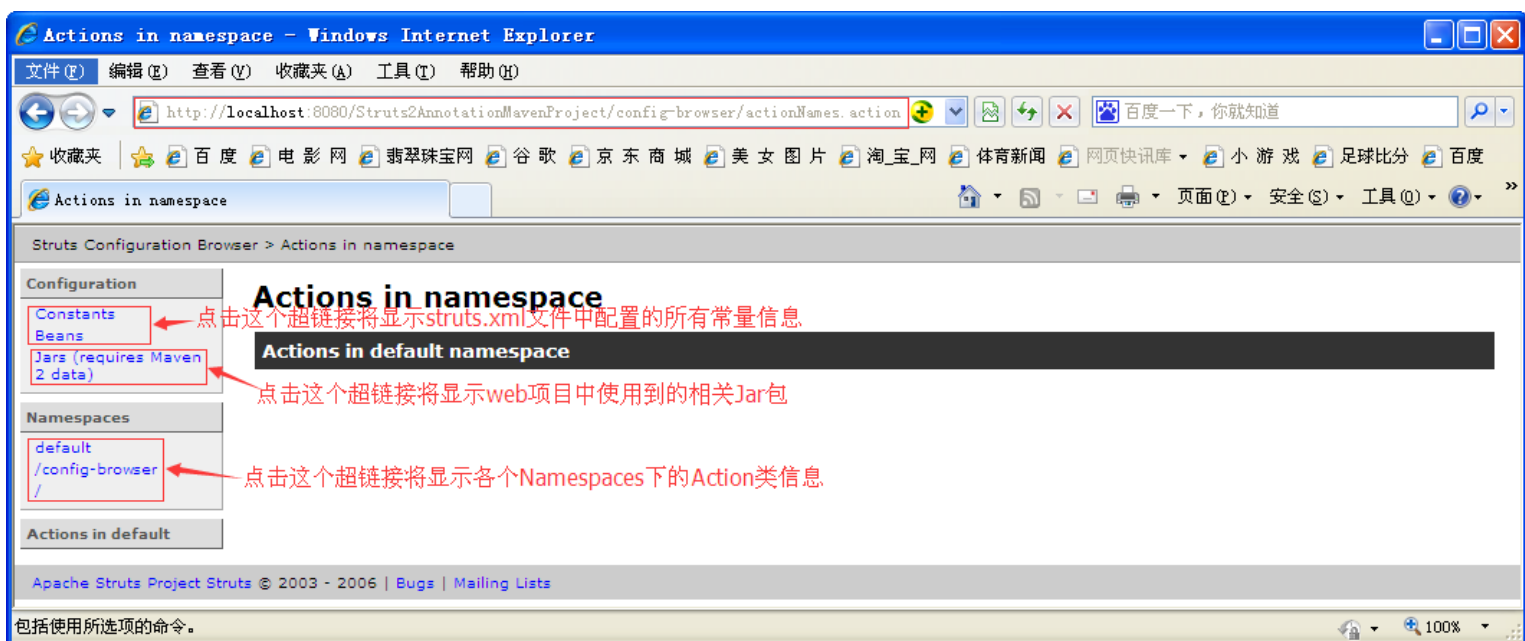
由于使用了convention-plugin插件的提供的注解方式映射Action的访问路径，所以我们不再需要像以前那样在Struts.xml文件中配置Action的访问路径了，测试结果如下：



可以看到，我们的TestAction里面的test方法已经成功调用了，这说明我们的Struts2框架的开发环境搭建成功。并且我们也感受到了使用convention-plugin插件开发基于注解的Struts2程序是非常方便和快捷的。

我们的项目中还使用到了config-browser-plugin插件，下面我们来感受一下这个config-browser-plugin插件带来的便利之处

输入访问URL: `http://localhost:8080/项目名/config-browser/index.action`来访问config-browser-plugin插件提供的视图页面，例如: `http://localhost:8080/Struts2AnnotationMavenProject/config-browser/index.action`，这样就可以进入config-browser-plugin插件提供的视图界面，如下图所示：



以上就是在MyEclipse中使用Maven搭建Struts2框架的开发环境的相关介绍，使用了Maven之后，大大提高了框架开发环境的搭建速度，起码我们不再需要关心Struts2框架的开发环境需要哪些jar包，我们只需要引入Struts2的核心jar包struts2-core，然后Maven就会自动帮我们吧struts2-core这个jar包的相关依赖jar全部加入到项目中，将web应用中的jar包全部交给Maven进行管理是一种非常好的做法，现在主流的项目都是采用Maven进行构建的。