

用Eclipse建spring mvc项目（图解） - fulai0_0的专栏 - 博客频道 - CSDN

最近工作需要快速了解spring mvc架构，于是在网上找了很多相关教程。说真的，百度出来的基本都一样，但是我却遇到各种奇奇怪怪的问题。当然，做完之后回头看，发现那些问题都不是问题，哈哈哈哈哈~主要还是不懂Java这些东西，遇到问题的时候就会觉得很难，各种查，但是百度的结果又是让人很无奈，不同的网址，相同的内容！



抓狂

我参考了一个博客来建项目，网址是：<http://limingnihao.iteye.com/blog/830409>。按照这个博客的流程创建了项目，但是由于实际情况的不同，中间还是有一些不一样的。

下面我把我用到的相关软件版本列一下，虽然我不知道这些有什么关系，但是，还是列一下吧，



害羞

哈哈哈哈哈~~

```
C:\Users\Administrator>mvn -v
Apache Maven 3.2.1 (ea8b2b07643dbb1b84b6d16e1f08391b666bc1e9; 2014-02-15T01:37:52+08:00)
Maven home: D:\apache-maven-3.2.1-bin\apache-maven-3.2.1
Java version: 1.7.0_17, vendor: Oracle Corporation
Java home: D:\Java\jdk1.7.0_17\jre
Default locale: zh_CN, platform encoding: GBK
OS name: "windows 7", version: "6.1", arch: "amd64", family: "windows"
C:\Users\Administrator>

Using CATALINA_BASE:   "D:\tomcat\apache-tomcat-6.0.41"
Using CATALINA_HOME:   "D:\tomcat\apache-tomcat-6.0.41"
Using CATALINA_TMPDIR: "D:\tomcat\apache-tomcat-6.0.41\tmp"
Using JRE_HOME:        "D:\Java\jdk1.7.0_17"
Using CLASSPATH:       "D:\tomcat\apache-tomcat-6.0.41\bin\bootstrap.jar"
Server version: Apache Tomcat/6.0.41
Server built:   May 19 2014 11:49:25
Server number:  6.0.41.0
OS Name:        Windows 7
OS Version:     6.1
Architecture:   amd64
JVM Version:    1.7.0_17-b02
JVM Vendor:     Oracle Corporation
```

因为不知道列哪些，所以就把这些都截了个图



害羞

，只知道要用jdk, maven, tomcat, 和java编译器（如eclipse）

现在开始建项目

1、创建一个maven项目

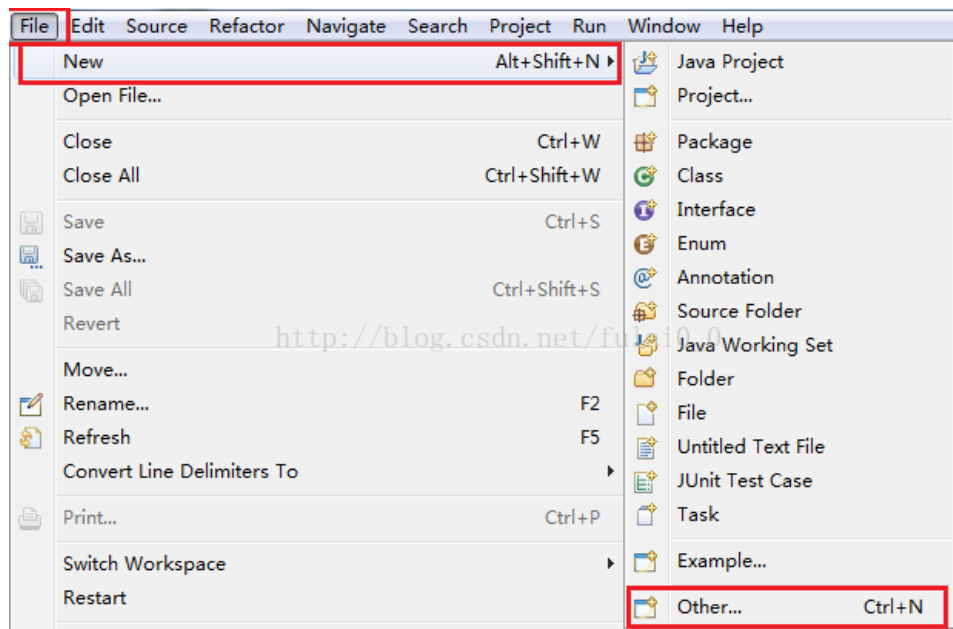


图1

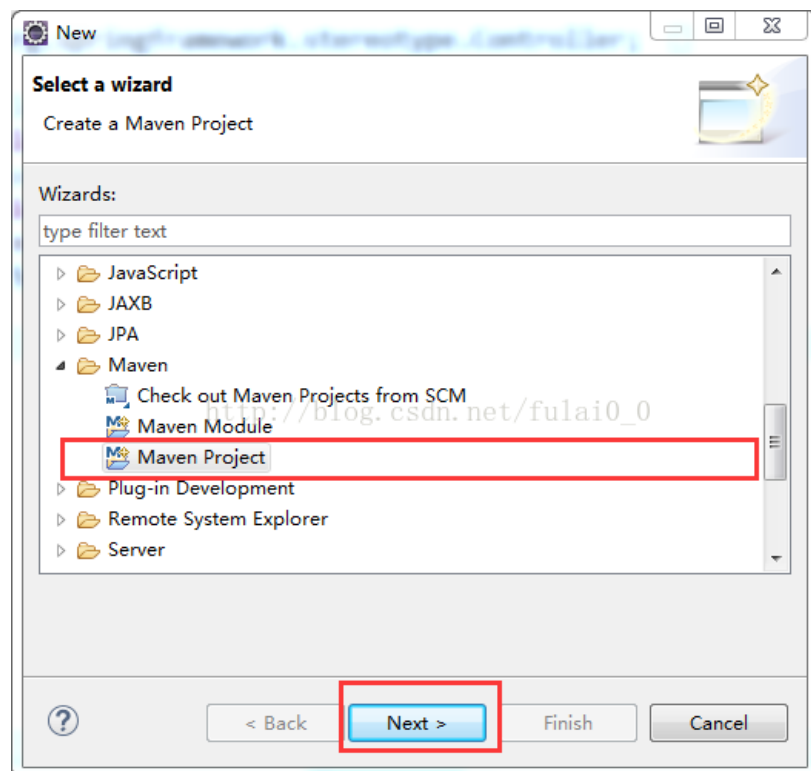


图2

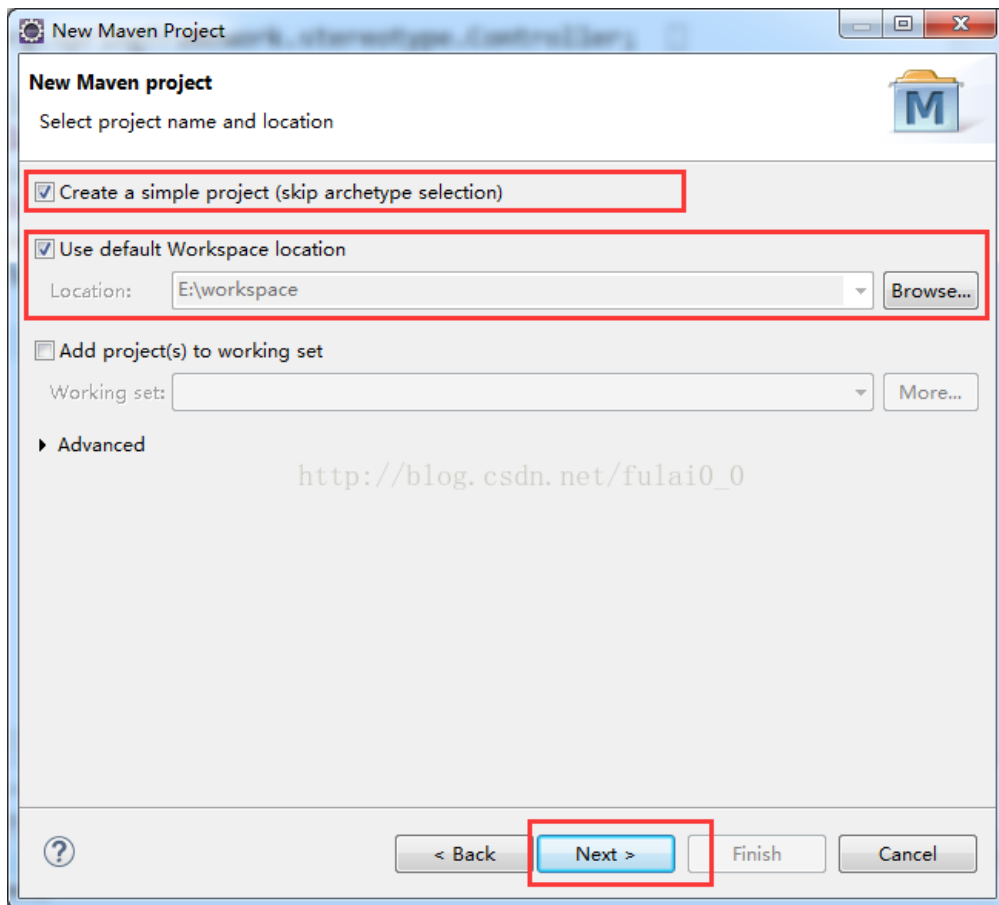


图3（第一个复选框一定要框，反正我这里不选就是错~）

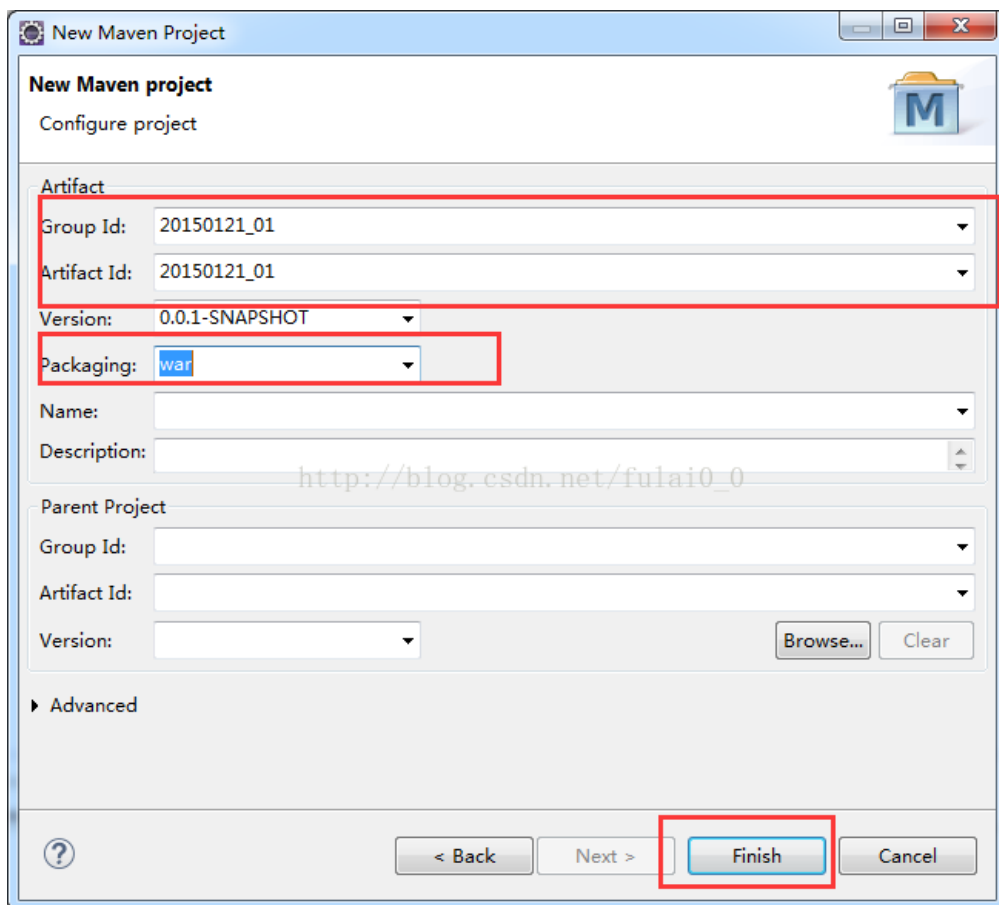


图4（Packaging选择war。反正我之前选jar就有问题了）

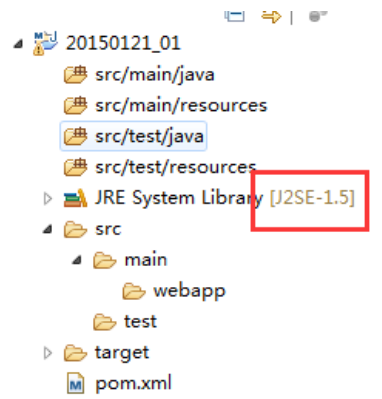


图5（红框中显示的不是我想要的，需要修改，右击它，选择“Build Path” ==> “ConfigureBuild Path”）

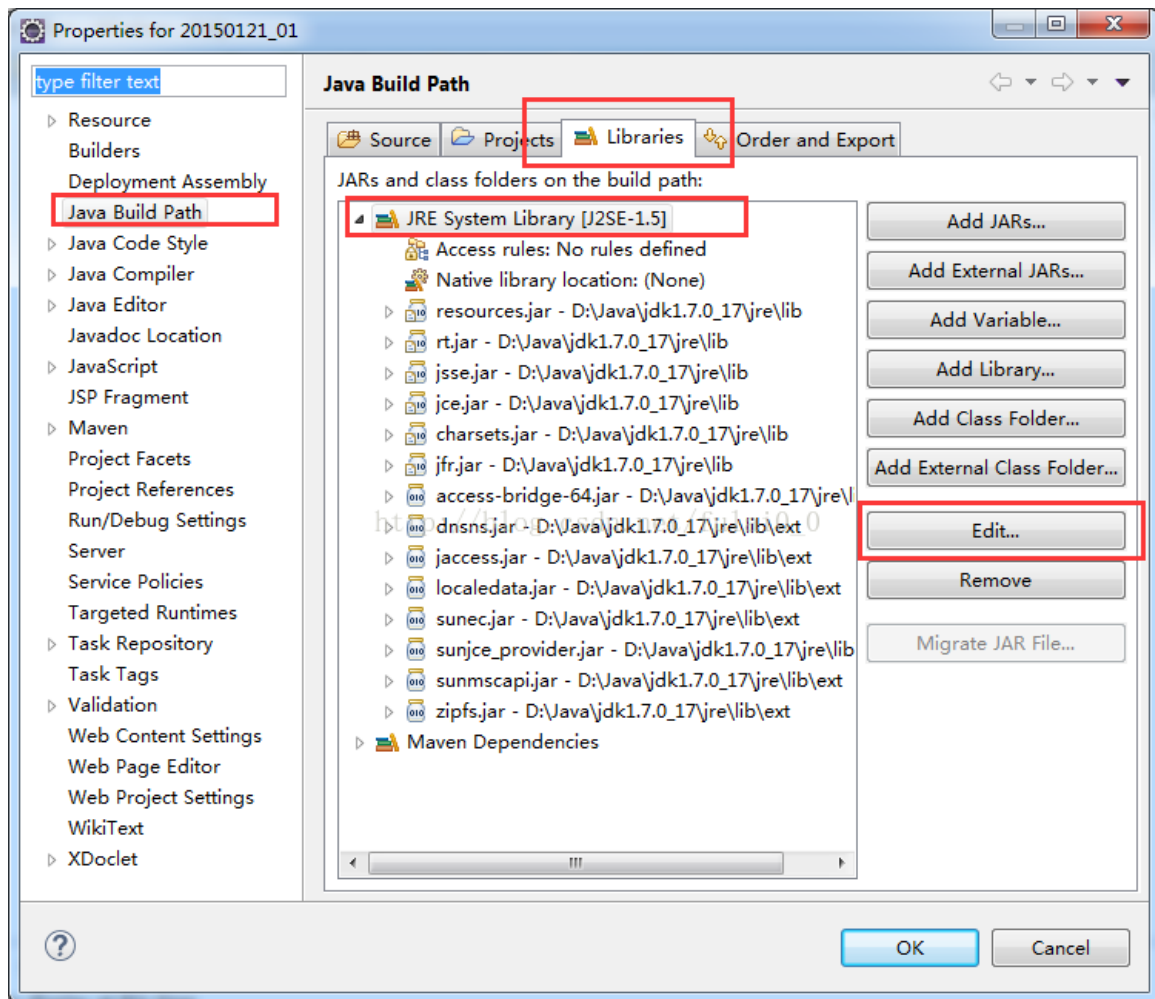


图6

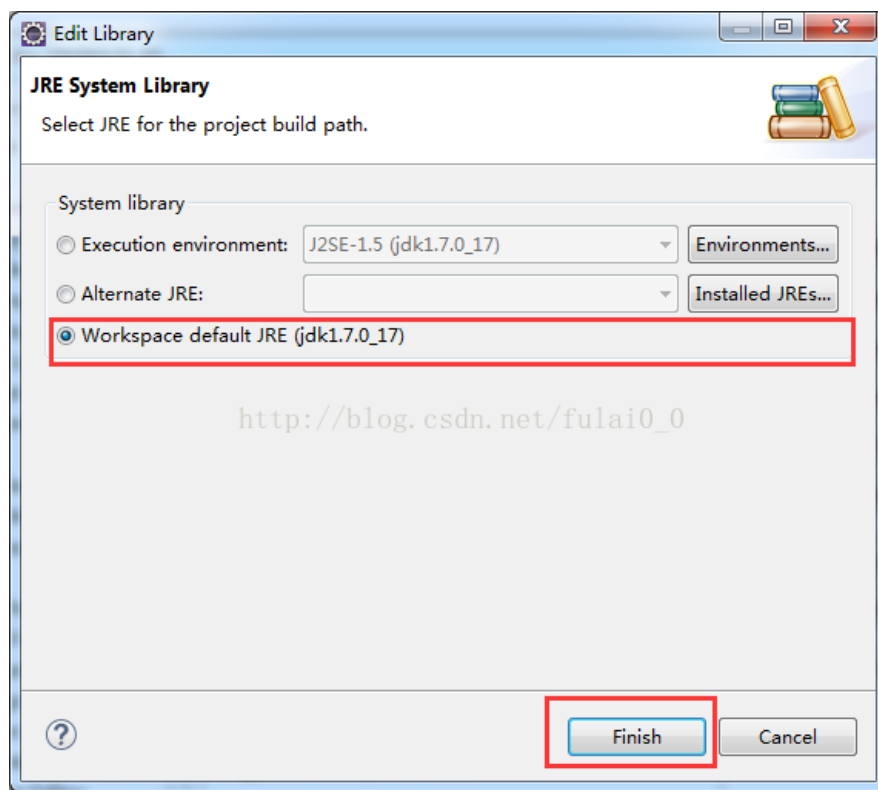


图7（不知道为什么选这个，只是之前看同事选了这个。）

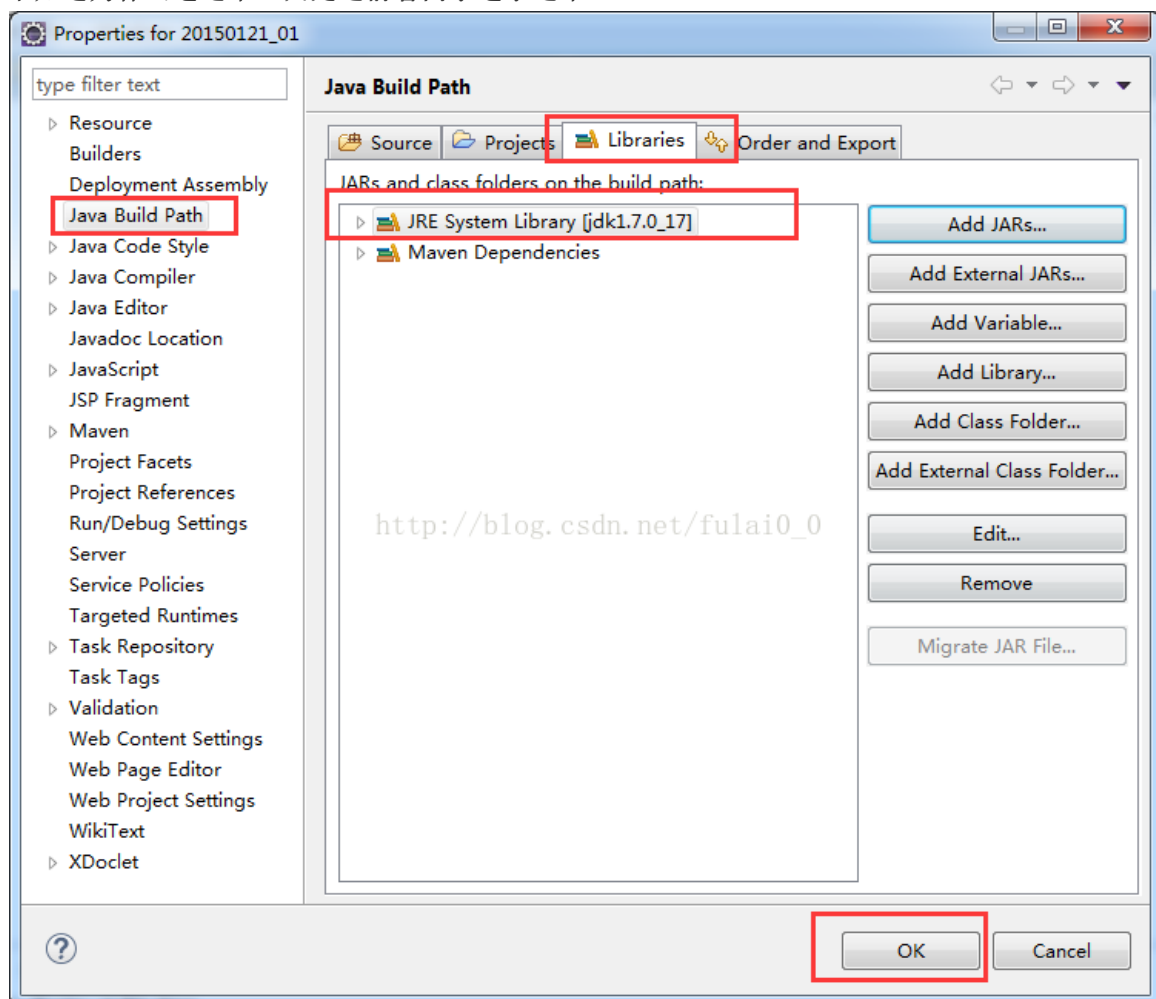


图8

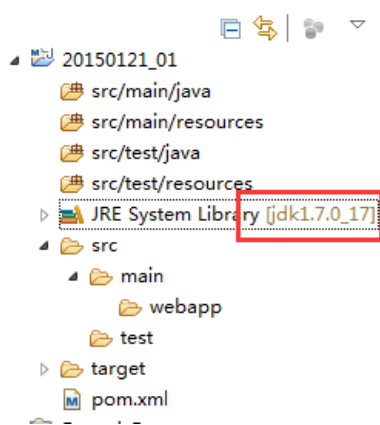


图9

到这一步，已经算是建好了一个mave项目。我们现在来配置一些东西。

2、配置pom.xml（可以直接添加到pom.xml里面，配置一些七七八八的东西。junit、jstl、spring包）

[html]view plaincopy

在CODE上查看代码片
派生到我的代码片

1. <dependencies>
2. <dependency>
3. <groupId>junitgroupId>
4. <artifactId>junitartifactId>
5. <version>4.12-beta-3version>
6. <scope>testscope>
7. dependency>
8. <dependency>
9. <groupId>log4jgroupId>
10. <artifactId>log4jartifactId>
11. <version>1.2.17version>
12. dependency>
13. <dependency>
14. <groupId>jstlgroupId>
15. <artifactId>jstlartifactId>
16. <version>1.2version>
17. dependency>
18. <dependency>
19. <groupId>org.springframeworkgroupId>
20. <artifactId>spring-coreartifactId>
21. <version>4.1.2.RELEASEversion>
22. dependency>
23. <dependency>
24. <groupId>org.springframeworkgroupId>
25. <artifactId>spring-expressionartifactId>
26. <version>4.1.2.RELEASEversion>

27. dependency>
28. <dependency>
29. <groupId>org.springframeworkgroupId>
30. <artifactId>spring-beansartifactId>
31. <version>4.1.2.RELEASEversion>
32. dependency>
33. <dependency>
34. <groupId>org.springframeworkgroupId>
35. <artifactId>spring-aopartifactId>
36. <version>4.1.2.RELEASEversion>
37. dependency>
38. <dependency>
39. <groupId>org.springframeworkgroupId>
40. <artifactId>spring-contextartifactId>
41. <version>4.1.2.RELEASEversion>
42. dependency>
43. <dependency>
44. <groupId>org.springframeworkgroupId>
45. <artifactId>spring-context-supportartifactId>
46. <version>4.1.2.RELEASEversion>
47. dependency>
48. <dependency>
49. <groupId>org.springframeworkgroupId>
50. <artifactId>spring-txartifactId>
51. <version>4.1.2.RELEASEversion>
52. dependency>
53. <dependency>
54. <groupId>org.springframeworkgroupId>
55. <artifactId>spring-webartifactId>
56. <version>4.1.2.RELEASEversion>
57. dependency>
58. <dependency>
59. <groupId>org.springframeworkgroupId>
60. <artifactId>spring-jdbcartifactId>
61. <version>4.1.2.RELEASEversion>
62. dependency>
63. <dependency>
64. <groupId>org.springframeworkgroupId>
65. <artifactId>spring-webmvcartifactId>
66. <version>4.1.2.RELEASEversion>
67. dependency>
68. <dependency>
69. <groupId>org.springframeworkgroupId>
70. <artifactId>spring-aspectsartifactId>
71. <version>4.1.2.RELEASEversion>
72. dependency>
73. <dependency>

- 74. <groupId>org.springframeworkgroupId>
- 75. <artifactId>spring-testartifactId>
- 76. <version>4.1.2.RELEASEversion>
- 77. dependency>
- 78. dependencies>

3、在webapp下添加一个文件夹“WEB-INF”，在文件夹下添加一个文件“web.xml”（下面的内容可以全部复制过去）

[html]view plaincopy

 [在CODE上查看代码片](#)
 [派生到我的代码片](#)

- 1. xmlversion="1.0"encoding="UTF-8"?>
- 2. <web-appxmlns="http://java.sun.com/xml/ns/javaee"xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
- 3. xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
- 4. version="2.5">
- 5. <context-param>
- 6. <param-name>webAppRootKeyparam-name>
- 7. <param-value>maven.example.rootparam-value>
- 8. context-param>
- 9. <listener>
- 10. <listener-class>org.springframework.web.util.Log4jConfigListenerlistener-class>
- 11. listener>
- 12. <filter>
- 13. <filter-name>CharacterEncodingFilterfilter-name>
- 14. <filter-class>org.springframework.web.filter.CharacterEncodingFilterfilter-class>
- 15. <init-param>
- 16. <param-name>encodingparam-name>
- 17. <param-value>UTF-8param-value>
- 18. init-param>
- 19. <init-param>
- 20. <param-name>forceEncodingparam-name>
- 21. <param-value>>trueparam-value>
- 22. init-param>
- 23. filter>
- 24. <filter-mapping>
- 25. <filter-name>CharacterEncodingFilterfilter-name>
- 26. <url-pattern>/*url-pattern>
- 27. filter-mapping>
- 28. <servlet>
- 29. <servlet-name>dispatcher0121servlet-name>
- 30. <servlet-class>org.springframework.web.servlet.DispatcherServletservlet-class>
- 31. <init-param>
- 32. <param-name>contextConfigLocationparam-name>
- 33. <param-value>/WEB-INF/dispatcher0121-servlet.xmlparam-value>

- 34. **init-param**>
- 35. **<load-on-startup>1load-on-startup**>
- 36. **servlet**>
- 37. **<servlet-mapping>**
- 38. **<servlet-name>dispatcher0121servlet-name**>
- 39. **<url-pattern>*.dourl-pattern**>
- 40. **servlet-mapping**>
- 41. **web-app**>

web.xml文件里有几个值得注意的地方，上面标红的“dispatcher0121”可以修改为你想要的字符串，但是这个三个地方一定要一样。[servlet-name]+“-servlet.xml”这个是spring配置文件的文件名的命名规则。

4、在“webapp/WEB-INF/”下再添加一个文件“dispatcher0121-servlet.xml”，这个文件就是上面说的东西（可以全部复制到新建的文件中）

[html]view plaincopy


在CODE上查看代码片

派生到我的代码片

- 1. **xmlversion="1.0"encoding="UTF-8"?>**
- 2. **<beansxmlns="http://www.springframework.org/schema/beans"**
- 3. **xmlns:aop="http://www.springframework.org/schema/aop"**
- 4. **xmlns:context="http://www.springframework.org/schema/context"**
- 5. **xmlns:mvc="http://www.springframework.org/schema/mvc"**
- 6. **xmlns:tx="http://www.springframework.org/schema/tx"**
- 7. **xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**
- 8. **xsi:schemaLocation="http://www.springframework.org/schema/aop**
- 9. **http://www.springframework.org/schema/aop/spring-aop-3.0.xsd**
- 10. **http://www.springframework.org/schema/beans**
- 11. **http://www.springframework.org/schema/beans/spring-beans-3.0.xsd**
- 12. **http://www.springframework.org/schema/context**
- 13. **http://www.springframework.org/schema/context/spring-context-3.0.xsd**
- 14. **http://www.springframework.org/schema/mvc**
- 15. **http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd**
- 16. **http://www.springframework.org/schema/tx**
- 17. **http://www.springframework.org/schema/tx/spring-tx-3.0.xsd">**
- 18. **<mvc:annotation-driven/>**
- 19. **<context:component-scanbase-package="example0121"/>**
- 20. **<beanclass="org.springframework.web.servlet.view.InternalResourceViewResolver">**
- 21. **<propertyname="prefix"value="/WEB-INF/views/">**
- 22. **<propertyname="suffix"value=".jsp"/>**
- 23. **bean**>
- 24. **beans**>

dispatcher0121-servlet.xml文件中也有要注意的地方，“example0121”这个是接下来会用的包，名字可以随意，但是不能全部是数字。我不想说，我就是偷懒用数字，导致之前建的项目一直有问题。



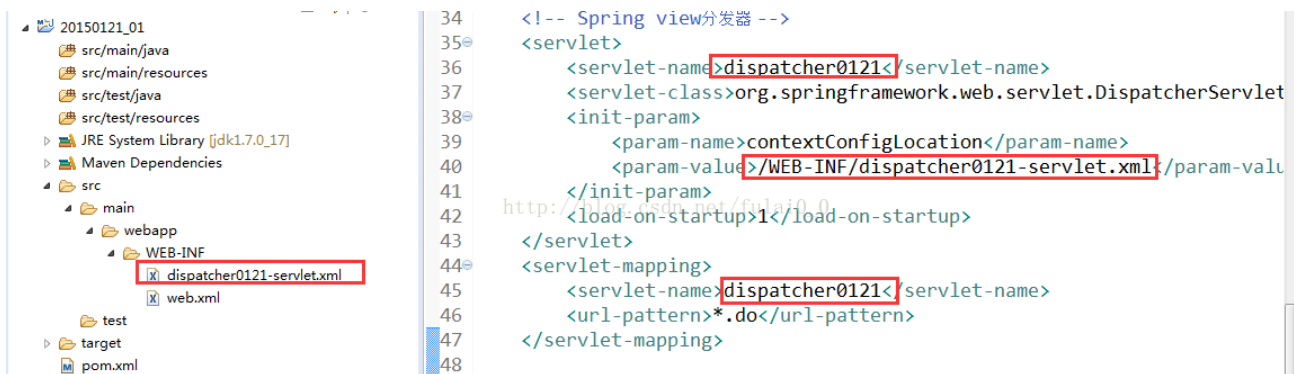


图10

5、创建一个Controller层测试类（这个是什么东西，我也不知道。只知道怎么操作）

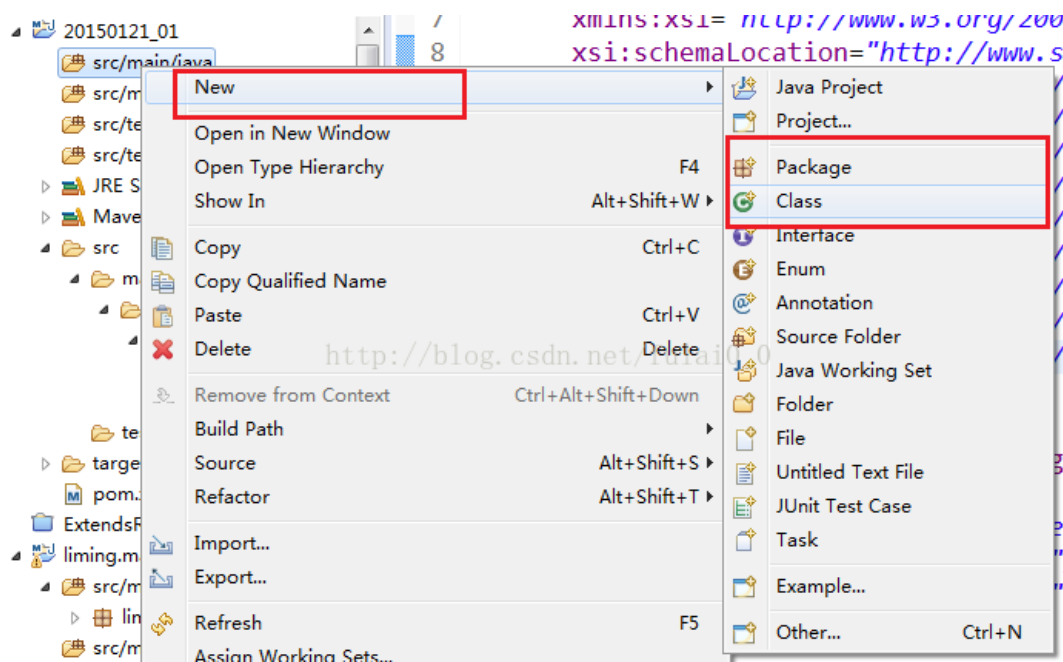


图11

按照这个操作，可以添加包也可以添加类，先添加类，要是发现在包那里没有写“example0121”，那就取消，然后添加一个包，包的名字是“example0121”。然后再创建一个叫“GeneralController”的类。

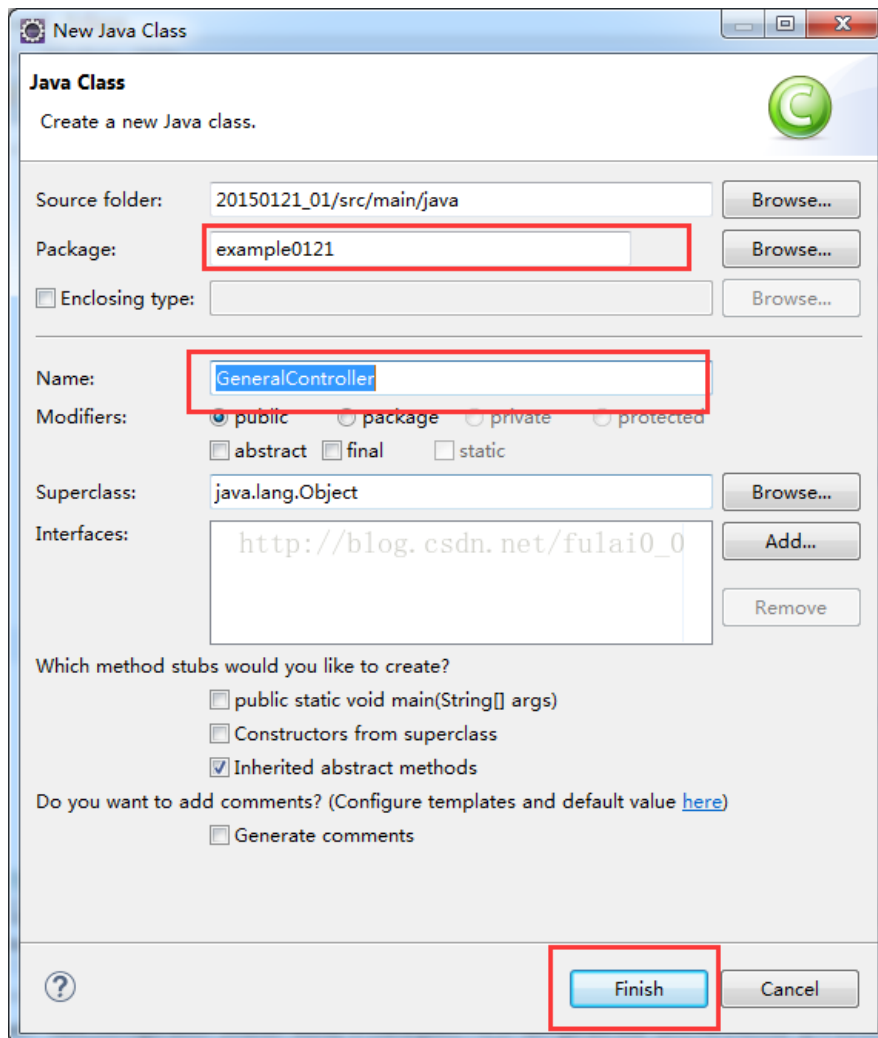


图12

然后将代码写好，像下面一样

[html]view plaincopy


在CODE上查看代码片
派生到我的代码片

```
1. package example0121;
2. import org.springframework.stereotype.Controller;
3. import org.springframework.ui.Model;
4. import org.springframework.web.bind.annotation.RequestMapping;
5. @Controller
6. public class GeneralController {
7.     @RequestMapping(value="index.do")
8.     public void index_jsp(Model model){
9.         model.addAttribute("str0121", "Hellow world");
10.         System.out.println("index.jsp");
11.     }
12. }
```

6、编写jsp页面。在“webapp/WEB-INF/”下添加一个文件夹“views”，再添加一个文件“index.jsp”。这个路径是在“dispatcher0121-servlet.xml”里面定下来的。

[html]view plaincopy

1. <%@ page language="java"contentType="text/html; charset=UTF-8"pageEncoding="UTF-8"%>
2. <%@ taglib prefix="c"uri="http://java.sun.com/jsp/jstl/core"%>
3. >
4. <html>
5. <head>
6. <metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">
7. <title>Insert title heretitle>
8. head>
9. <body>
10. <c:outvalue="\${str0121}">c:out>
11. body>
12. html>

这个时候你可以看到你的工程出现红色的叉叉，报了个错，如图13。这个时候你要淡定，两种方法。一是无视它，二是删掉这个错。我选择了无视它，我因为这个错查了好久资料，莫名烦躁，试了网上说的方法，还是不行。最后问了同事，同事说，你运行一下可不可以用。我运行了项目，发现可以用。然后同事说，那就无视它。似乎好像是校验的问题，反正现在不影响。one or two? !

1 error, 0 warnings, 0 others			
Description	Resource	Path	Location
Errors (1 item)	http://blog.csdn.net/fulai00		
The superclass "javax.servlet.http.HttpServlet" was not found on the Java Build Path	index.jsp	/20150121_01/src...	line 1

图13

7、运行

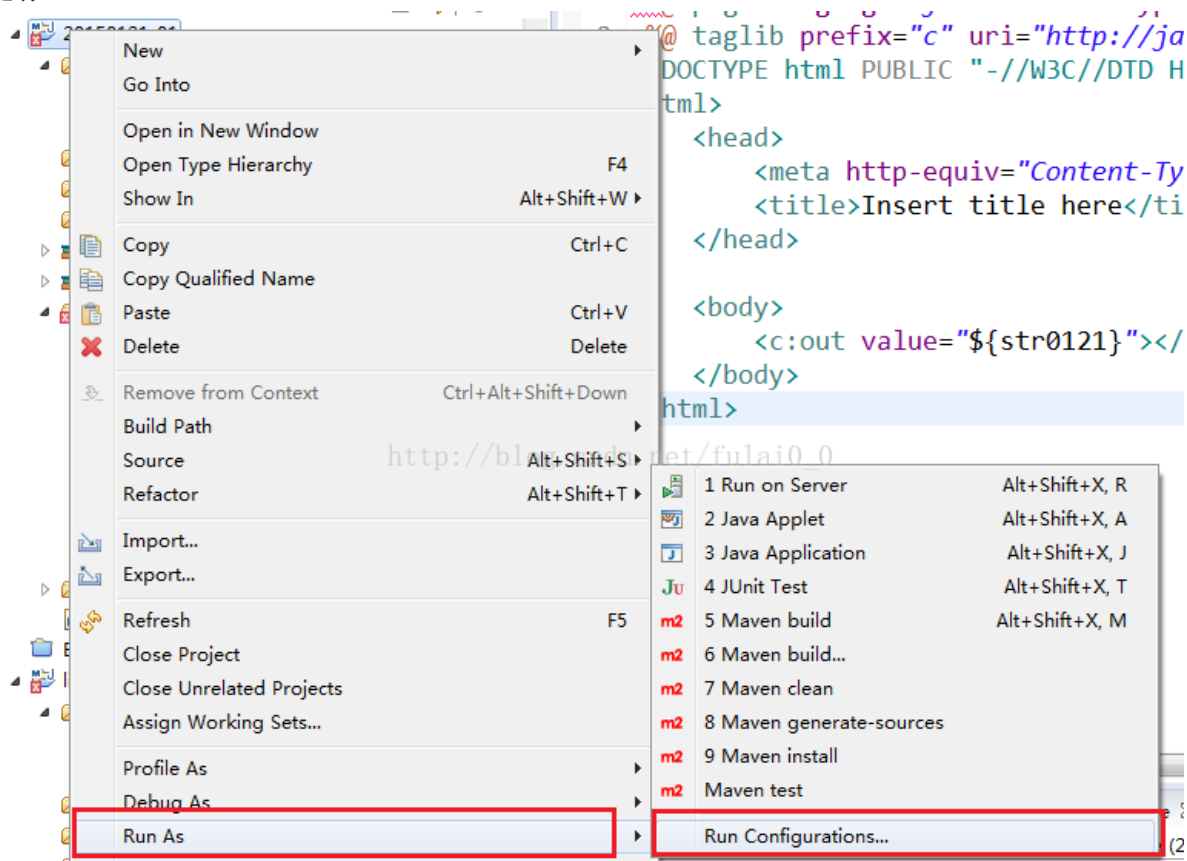


图14

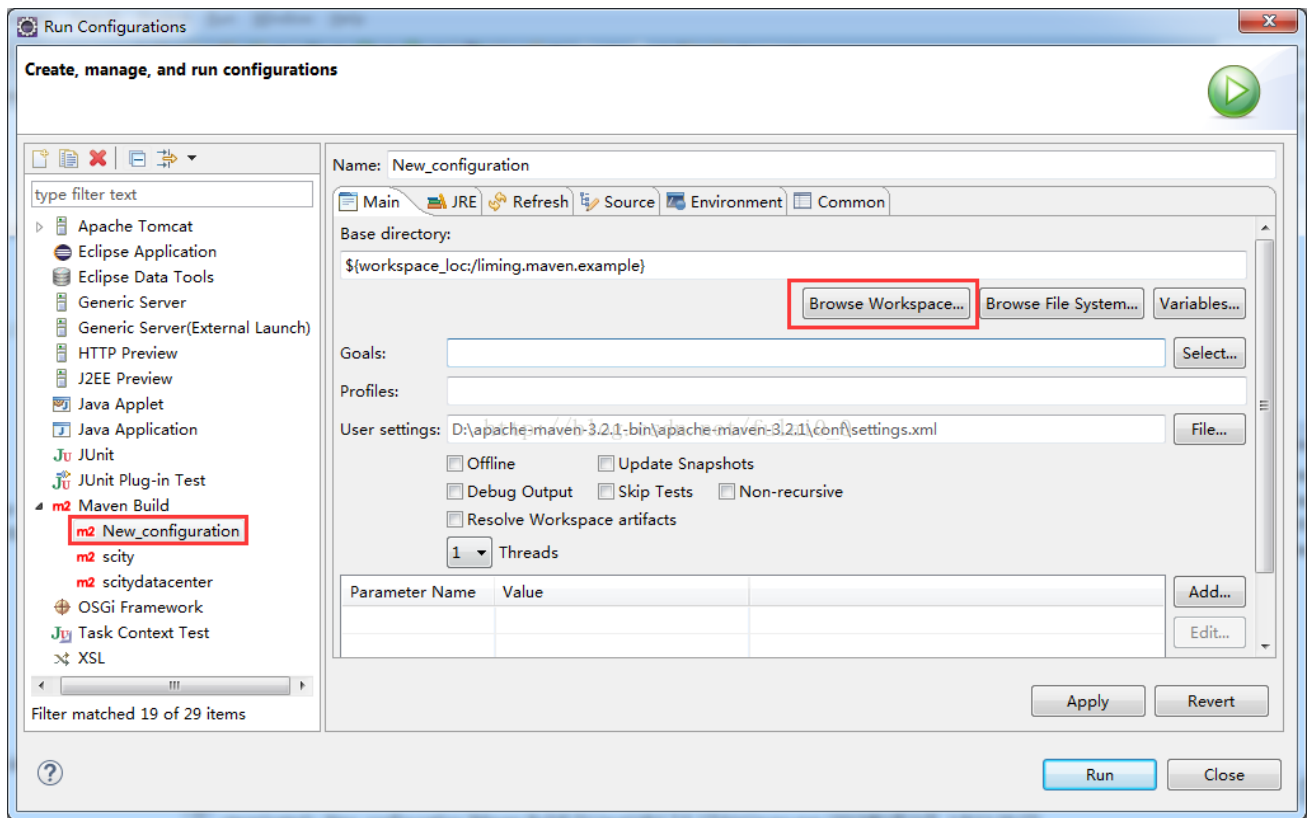


图15

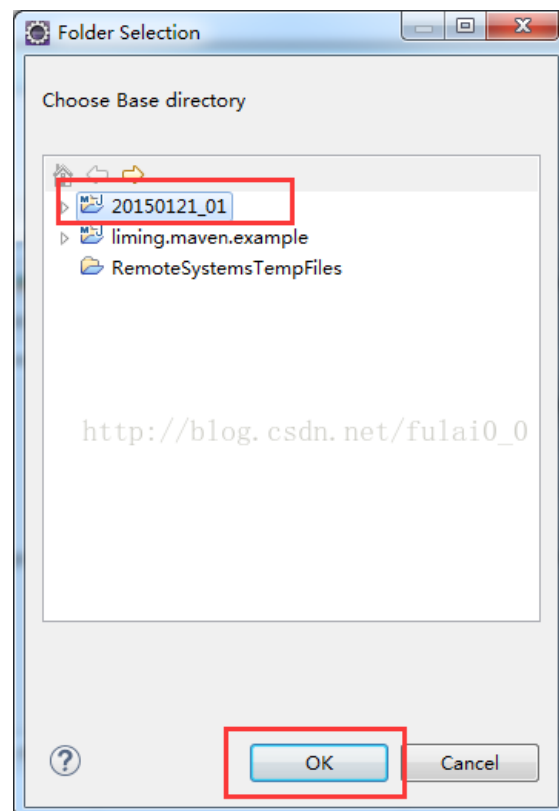


图16

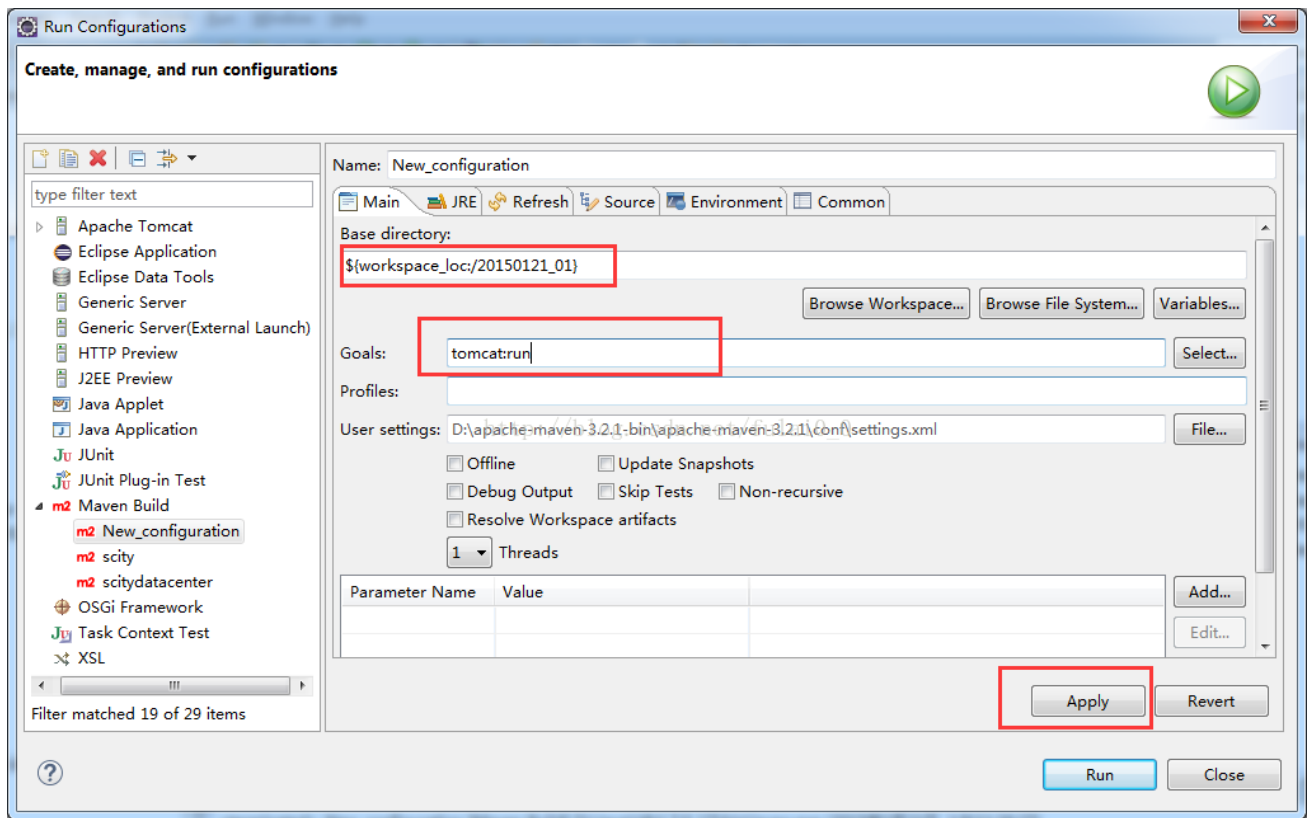


图17

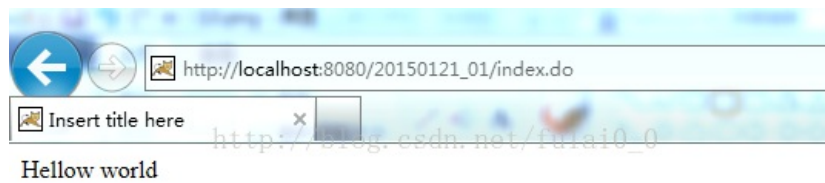


图18

现在算是创建好了项目。要开始深层次的探讨了

大笑
大笑
大笑
大笑