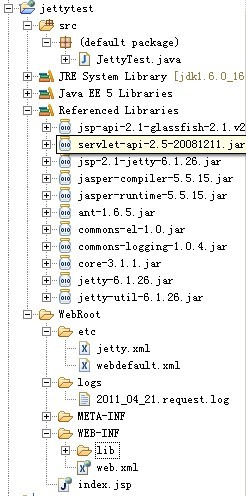
[**jetty的使用**](http://gzg0141.iteye.com/blog/1721750)

之前都用tomcat的童鞋是不是觉得tomcat很方便啊？但是还有个更方便的web服务器叫jetty，而且它的体积很小，只有几百KB，所以在项目中使用是很方便的。   
  下面来介绍下jetty所必需的一些东西：请注意看他的包结构



这些包都是必需的。你可以去网上下个jetty-7.2版本的类似于tomecat的东西。

其实运用jetty的方法比较多。我们一一来分析。

方法1：

 看到jetty.xml和webdefault.xml这2玩意了吗？

 webdefault.xml官方的解释是 The default descriptor is a web.xml format file that is applied to the context before the standard WEB-INF/web.xml 你可以理解为在web.xml加载之前先加载的jetty所需的一个容器的配置文件。

你可以去你下载的jetty里面拿出来 比如说你的路径D:\jetty-hightide-7.2.1.v20101111\etc 去这里拽个出来放在你项目里面。内容基本上不要改。

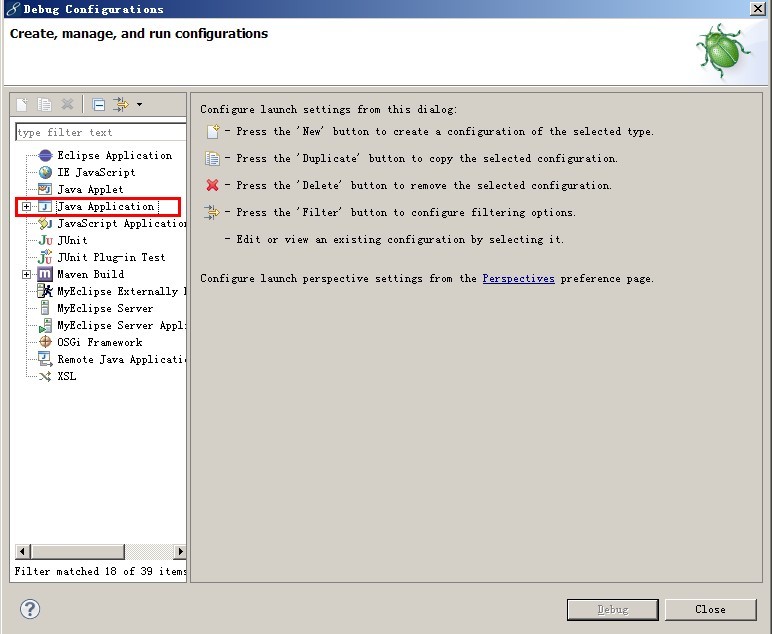
然后我们来看下jetty.xml 这是一个负责告诉jetty容器要运行的你的工程是什么。

咱们可以先看下它的内容：

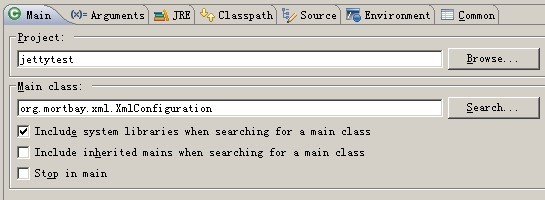
**Xml代码  收藏代码**

1. **<?xml** version="1.0"**?>**
2. <!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN" "http://jetty.mortbay.org/configure.dtd"**>**
3. **<Configure** id="Server" class="org.mortbay.jetty.Server"**>**
4. **<Set** name="ThreadPool"**>**
5. **<New** class="org.mortbay.thread.QueuedThreadPool"**>**
6. **<Set** name="minThreads"**>**10**</Set>**
7. **<Set** name="maxThreads"**>**200**</Set>**
8. **<Set** name="lowThreads"**>**20**</Set>**
9. **<Set** name="SpawnOrShrinkAt"**>**2**</Set>**
10. **</New>**
11. **</Set>**
12. **<Call** name="addConnector"**>**
13. **<Arg>**
14. **<New** class="org.mortbay.jetty.nio.SelectChannelConnector"**>**
15. <!--你的IP-->
16. **<Set** name="host"**><SystemProperty** name="jetty.host" default="localhost"**/></Set>**
17. <!--你的端口-->
18. **<Set** name="port"**><SystemProperty** name="jetty.port" default="9090"**/></Set>**
19. **<Set** name="maxIdleTime"**>**30000**</Set>**
20. **<Set** name="Acceptors"**>**2**</Set>**
21. **<Set** name="statsOn"**>**false**</Set>**
22. **<Set** name="confidentialPort"**>**8443**</Set>**
23. **<Set** name="lowResourcesConnections"**>**5000**</Set>**
24. **<Set** name="lowResourcesMaxIdleTime"**>**5000**</Set>**
25. **</New>**
26. **</Arg>**
27. **</Call>**
29. **<Set** name="handler"**>**
30. **<New** id="Handlers" class="org.mortbay.jetty.handler.HandlerCollection"**>**
31. **<Set** name="handlers"**>**
32. **<Array** type="org.mortbay.jetty.Handler"**>**
33. **<Item>**
34. **<New** id="RequestLog" class="org.mortbay.jetty.handler.RequestLogHandler"**/>**
35. **</Item>**
36. **<Item>**
37. **<New** class="org.mortbay.jetty.webapp.WebAppContext"**>**
38. **<Set** name="contextPath"**>**/jettytest**</Set>**
39. **<Set** name="war"**>**./WebRoot**</Set>**  <!--你的web应用根目录-->
40. **</New>**
41. **</Item>**
42. **</Array>**
43. **</Set>**
44. **</New>**
45. **</Set>**
46. **<Ref** id="RequestLog"**>**
47. **<Set** name="requestLog"**>**
48. **<New** id="RequestLogImpl" class="org.mortbay.jetty.NCSARequestLog"**>**
49. **<Set** name="filename"**><SystemProperty** name="jetty.logs"   default="./WebRoot/logs"
50. **/>**/yyyy\_mm\_dd.request.log**</Set>**
51. **<Set** name="filenameDateFormat"**>**yyyy\_MM\_dd**</Set>**
52. **<Set** name="retainDays"**>**90**</Set>**
53. **<Set** name="append"**>**true**</Set>**
54. **<Set** name="extended"**>**true**</Set>**
55. **<Set** name="logCookies"**>**false**</Set>**
56. **<Set** name="LogTimeZone"**>**GMT**</Set>**
57. **</New>**
58. **</Set>**
59. **</Ref>**
60. **<Set** name="stopAtShutdown"**>**true**</Set>**
61. **<Set** name="sendServerVersion"**>**true**</Set>**
62. **<Set** name="sendDateHeader"**>**true**</Set>**
63. **<Set** name="gracefulShutdown"**>**1000**</Set>**
64. **</Configure>**

字面就很好理解吧？看完了上面的你啥类都可以先不写，可以进入debug配置里面 如下图



右击new 一个如下图来配置





到这里方法1就基本完成了,保存配置 run debug！http://localhost:9090/jettytest 直接去访问吧。

方法2和方法3 都是用java代码来call jetty。

方法2先来：

**Java代码  收藏代码**

1. **public** **static** **void** startJettyMethodO() {
2. String jetty\_home = "jettytest";//这个就是你的项目发布时候的名字
3. **try** {
4. Server server = **new** Server();
5. Connector connector = **new** SelectChannelConnector();
6. connector.setPort(9999);
7. server.setConnectors(**new** Connector[] { connector });
8. WebAppContext webapp = **new** WebAppContext();
9. webapp.setContextPath("/"+jetty\_home);//上下文路径 比如说/jettytest
10. webapp.setResourceBase("./WebRoot");//你的资源文件所在的路径 一般都在这下面
11. webapp.setDefaultsDescriptor("./WebRoot/etc/webdefault.xml");
12. server.setHandler(webapp);
13. server.start();
14. server.join();
15. } **catch** (Exception e) {
16. e.printStackTrace();
17. }
18. }

  方法3：

**Java代码  收藏代码**

1. FileNotFoundException, Exception {
2. Server server = **new** Server();
3. XmlConfiguration configuration = **new** XmlConfiguration(
4. **new** FileInputStream(
5. "D:\\teamwork\\jettytest\\WebRoot\\etc\\jetty.xml")); //指定自定义的jetty.xml路径
6. configuration.configure(server);
7. server.start();

附：官方api<http://jetty.codehaus.org/jetty/jetty-6/apidocs/index.html>

其他我也没啥好说的了。如果还有其他方法的话我会更新的。多谢支持~！

**spiritfrog 写道**

总结：   
本文作为jetty的初体验体会，着重于怎样移植webapp到jetty，以及如何快速在jetty上部署webapp，也尝试了嵌入方式启动webapp，这些工作在找到方法后很轻松（实际上我可找了很久）。主要是完成你的webapp对jetty的配置，然后start！   
本文并没有重点讨论jetty.xml的配置，这部分内容可以根据官方文档和看jetty自带的examples来学习，以后有时间看了再说。   
  
注意：   
jetty6是基于servlet2.5和jsp2.1设计的，   
几个核心包是jetty-6.1.11.jar、jetty-util-6.1.11.jar、servlet-api-2.5-6.1.11.jar；   
如果以嵌入式启动，需要引用jsp2.1包，我一开始引用的是jsp2.0包，结果造成了一些方法调用对不上参数的问题，明显是class版本有差别，换用jsp2.1就可以了。   
另外对于嵌入式，/lib/naming和/lib/plus下的包最好也全部引入。   
有时候jetty配置文件中会有如下引用jetty.home等变量   
<Set name="config"><SystemProperty name="jetty.home" default="."/>/etc/realm.properties</Set>   
需要在启动程序配置中加上vm参数，如果是jetty.home,则加上   
-Djetty.home=D:/devworkspace/jetty-6.1.11