

Tomcat windows 下的下载及其配置

一、Tomcat 的下载

下载地址: <http://tomcat.apache.org/download-80.cgi>

8.0.30

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [zip \(pgp, md5, sha1\)](#)
 - [tar.gz \(pgp, md5, sha1\)](#)
 - [32-bit Windows zip \(pgp, md5, sha1\)](#)
 - [64-bit Windows zip \(pgp, md5, sha1\)](#)
 - [64-bit Itanium Windows zip \(pgp, md5, sha1\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, md5, sha1\)](#)

选中最后一项 32-bit/64-bit Windows Service Installer 即可下载

注: 特别注意不要追求新版下载 9.0 的 alpha 版本, 这只是个测试版, 不稳定;

另外这个只是下载的 Tomcat 8.0.30 版本, 要求的 Java 最低版本是 Java 7.如果是 Java6 版本需要下载 Tomcat 7 版本。具体说明见下图

Servlet Spec	JSP Spec	EL Spec	WebSocket Spec	Apache Tomcat version	Actual release revision	Support Java Versions
4.0	TBD (2.4?)	TBD (3.1?)	TBD (1.2?)	9.0.x	9.0.0.M1 (alpha)	8 and later
3.1	2.3	3.0	1.1	8.0.x	8.0.30	7 and later
3.0	2.2	2.2	1.1	7.0.x	7.0.67	6 and later (WebSocket 1.1 requires 7 or later)
2.5	2.1	2.1	N/A	6.0.x	6.0.44	5 and later
2.4	2.0	N/A	N/A	5.5.x (archived)	5.5.36 (archived)	1.4 and later
2.3	1.2	N/A	N/A	4.1.x (archived)	4.1.40 (archived)	1.3 and later
2.2	1.1	N/A	N/A	3.3.x (archived)	3.3.2 (archived)	1.1 and later

然后对应的下载连接在 <http://tomcat.apache.org/index.html> 的对应版本下点击

[Download](#) 就可以了

Tomcat 7.0.67 Released

2015-12-10

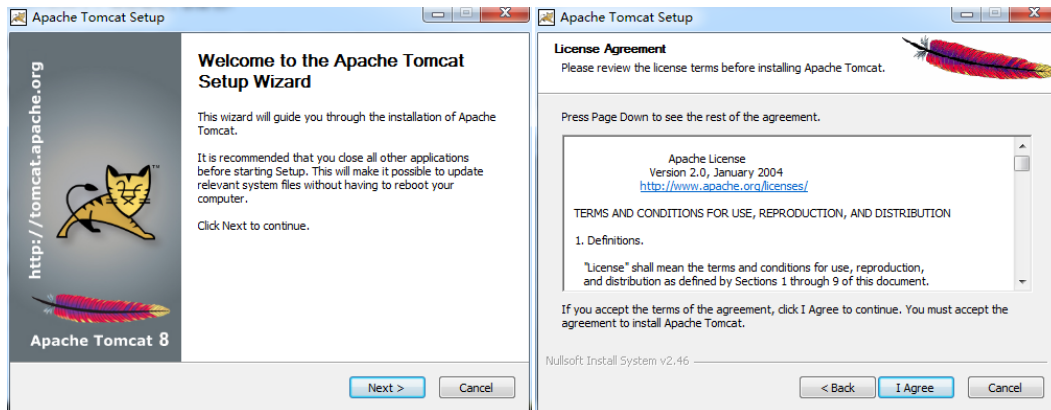
The Apache Tomcat Project is proud to announce the release of version 7.0.67 of Apache Tomcat. This release contains a number of bug fixes and improvements compared to version 7.0.65. The notable changes since 7.0.65 include:

- Allow file based configuration resources (e.g. key stores) to be configured using URLs.
- Add an option to control (per context) quoting of EL expressions in JSP attributes. Restore the default behavior in Jasper to align with 7.0.64 and earlier as well as other JSP implementations.
- Add a new RestCsrfPreventionFilter that provides basic CSRF protection for REST APIs.
- Use instance manager for WebSocket server endpoint instances.
- Location headers for redirects now use relative URIs. This can be controlled per Context with the useRelativeRedirects attribute.

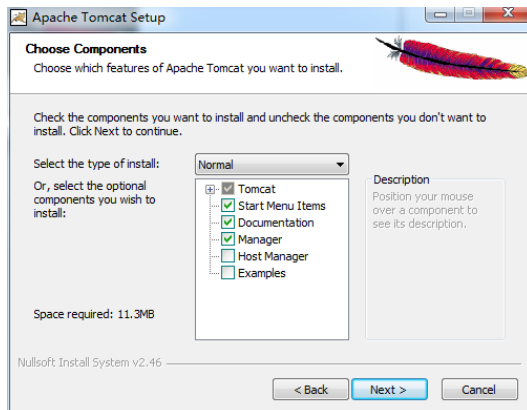
Full details of these changes, and all the other changes, are available in the [Tomcat 7 changelog](#).

[Download](#) | [ChangeLog for 7.0.67](#)

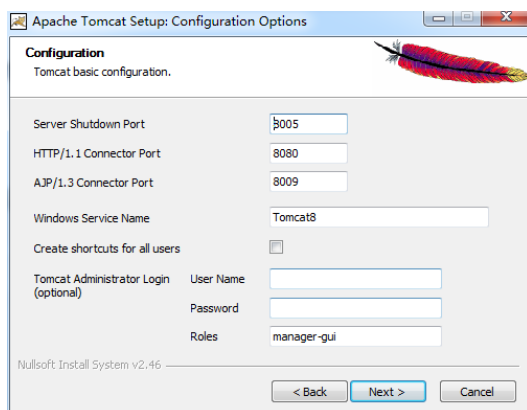
二、 下载到的这个 exe 可执行文件，双击即可进入安装。安装时一路 next 好像就可以了



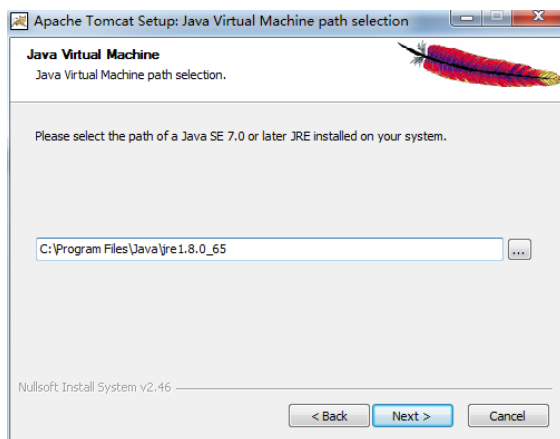
这里我全选了，不过网上教程说 normal 就可以了



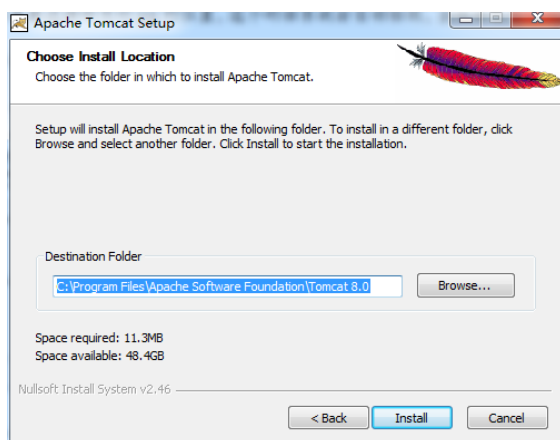
用户名和密码貌似没什么用，空着就好了



下面需要选择所用的 jre 的位置，这个的话系统会自动检测，当然你也可以改(装了多个 java，有多个 jre 的同学建议只保留一个这样就不会混淆现在用的是哪个了)

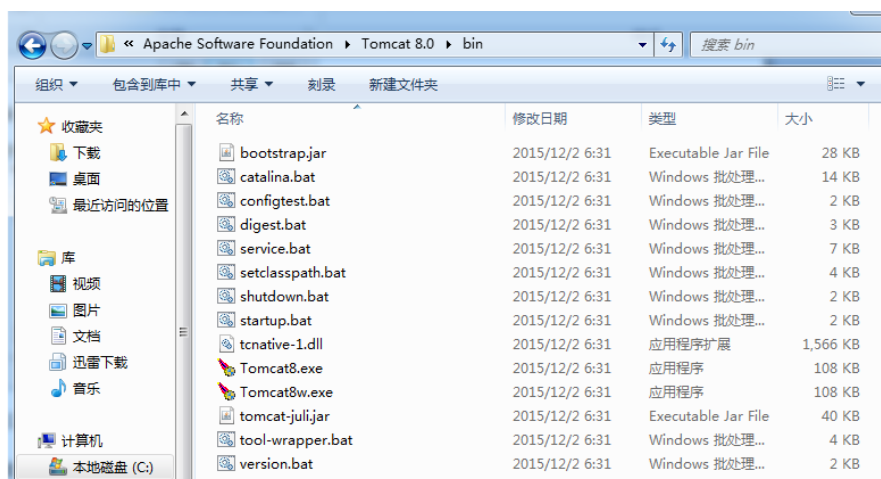


接下来选择安装路径，你们想放哪里都可以，但是要记住

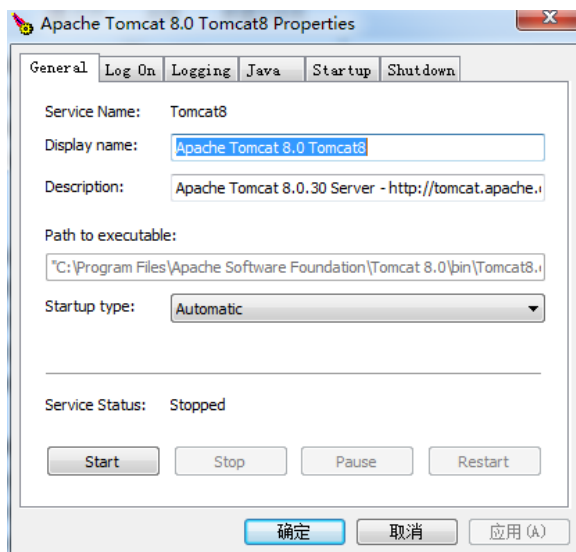


三、 检测 Tomcat 安装是否成功

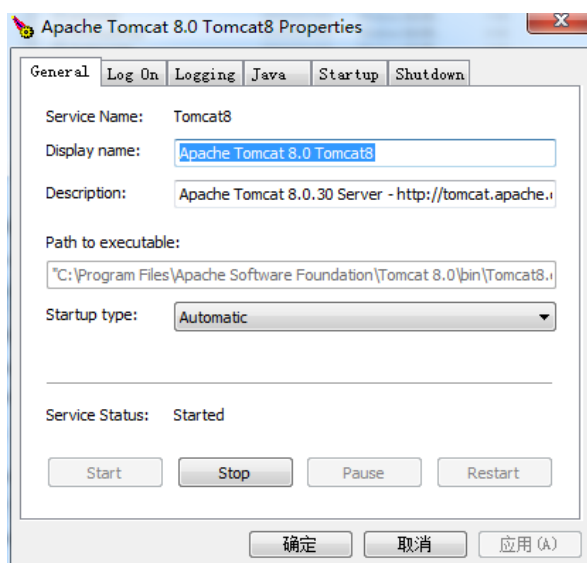
进到安装目录的 bin 文件夹下，然后双击里面的 Tomcat8w.exe 运行



然后点击 start 运行即可



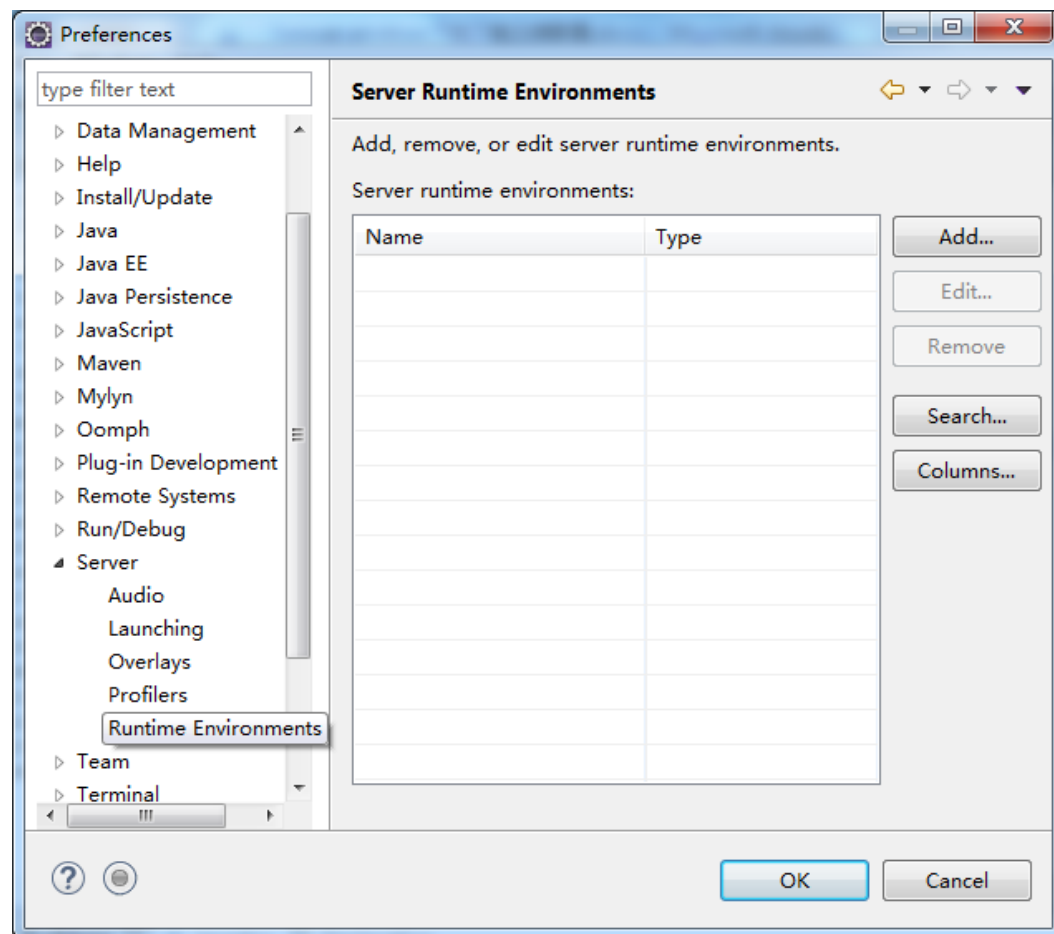
这样就算处于运行状态了，这时可以通过 netstat 查看 8080 端口（web 后台返回 http 页面的端口）是否打开



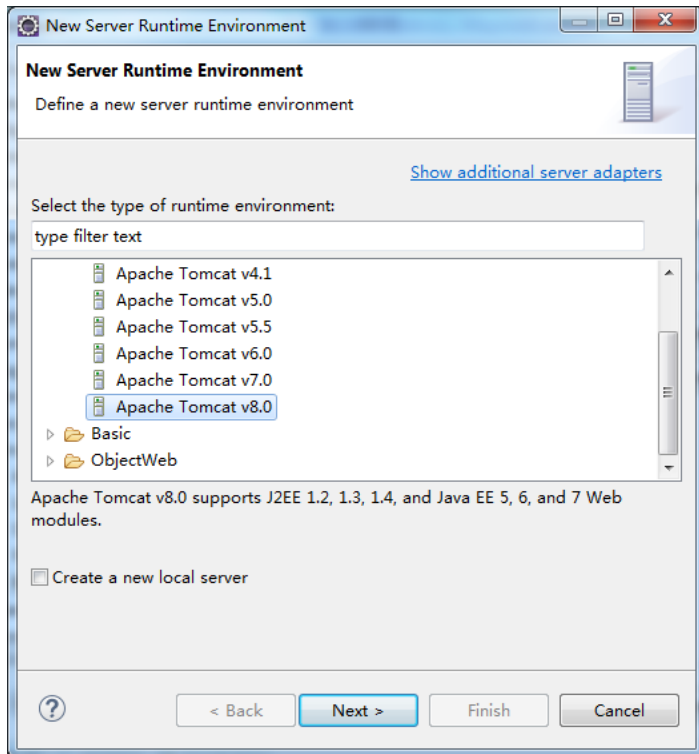
活动连接

协议	本地地址	外部地址	状态
TCP	0.0.0.0:7	0.0.0.0:0	LISTENING
TCP	0.0.0.0:9	0.0.0.0:0	LISTENING
TCP	0.0.0.0:13	0.0.0.0:0	LISTENING
TCP	0.0.0.0:17	0.0.0.0:0	LISTENING
TCP	0.0.0.0:19	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1026	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1027	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1028	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1030	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1031	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3306	0.0.0.0:0	LISTENING
TCP	0.0.0.0:6001	0.0.0.0:0	LISTENING
TCP	0.0.0.0:8009	0.0.0.0:0	LISTENING
TCP	0.0.0.0:8080	0.0.0.0:0	LISTENING

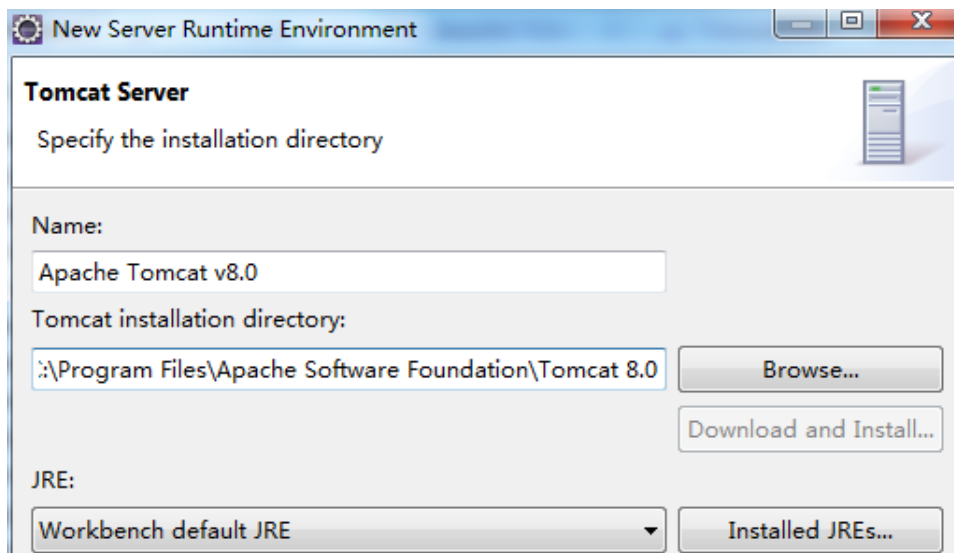
选择 Windows 菜单下的 preference 选项，选中 server 下的 runtime environment



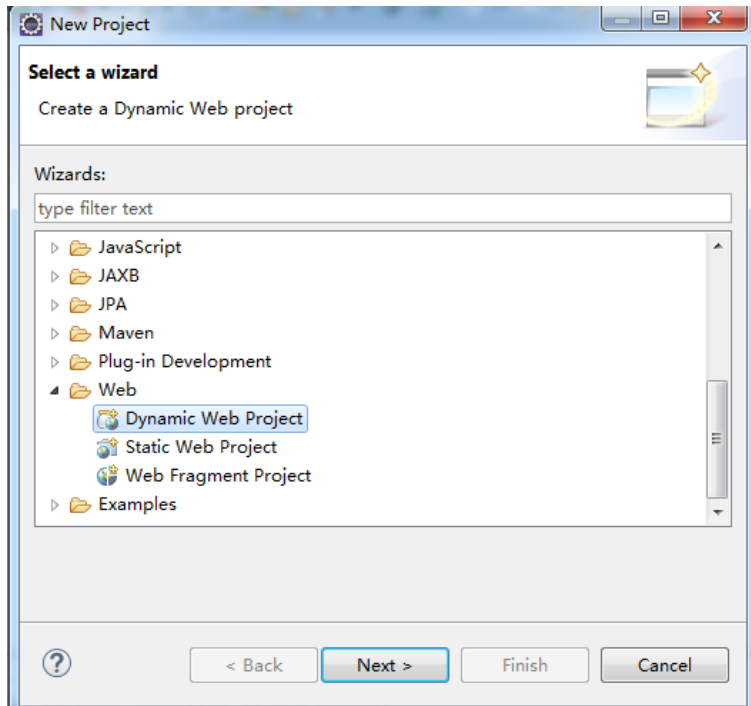
点击 Add 进行添加，选择对应的 Tomcat 安装版本



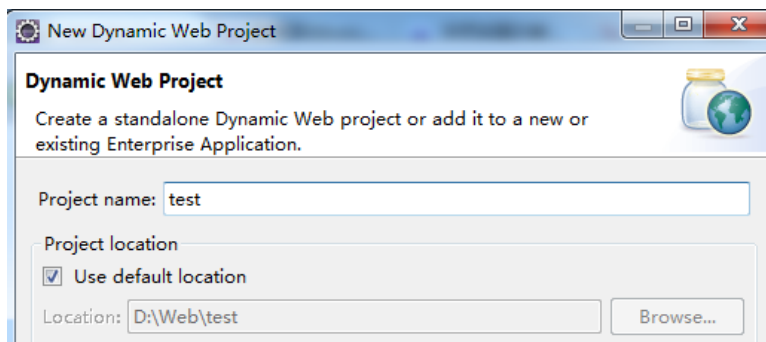
点击 Browse 选择 Tomcat 的安装路径(具体到 Tomcat x.x 这个文件夹就可以了)



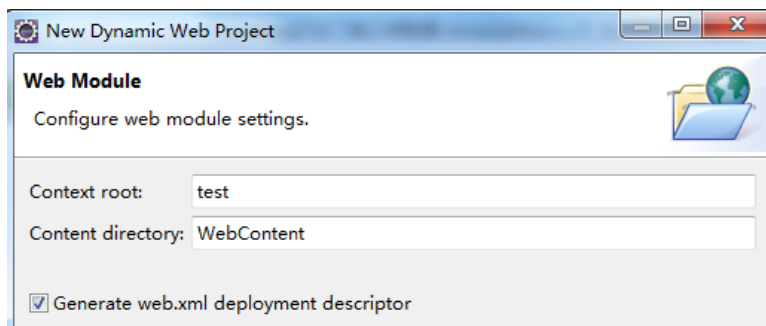
新建一个动态页面的工程



然后 next 选择工程名，然后继续 next



再 next 以后会有一个是否生成 web.xml 的选项(这个文件相当于注册表，所有的后台程序或动态页面需要在这上面进行注册)，勾选上就可以 finish 了。

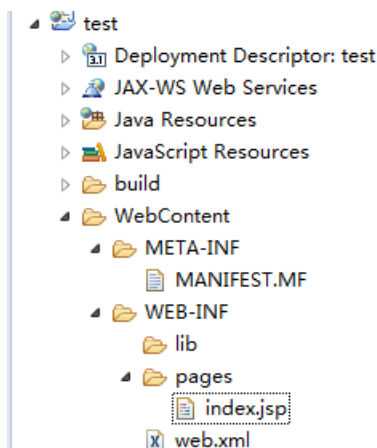


然后进入那个注册文件，发现原来有很多项的起始页(按照默认的行为会根据这个列表去根目录找(特别注意：这个根目录指的是工程下的 WebContent 这个文件

夹)有没有对应的文件(先看 index.html 是否存在),如果没有就跳到下一个文件(如果 index.html 没有就会去找 index.htm),如果全部都没有那么等到后面运行时就会出现 404 找不到文件错误了)。我们也可以只把一项作为欢迎页

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/
  <display-name>test</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

这时候我们就可以在指定的路径下新建一个 jsp 文件了,由于根目录是在工程下的 WebContent 这个文件夹,因此我们可以直接把文件新建在这个文件夹下面。如果不新建在这个下面也可以,你们稍微改一下路径就好了。比如说我在 WebContent 这个文件夹的 WEB-INF 文件夹下新建了一个 pages 文件夹,并把 index.jsp 放在了里面



这时候就需要修改刚刚的 web.xml 为

```
web.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/
3   <display-name>test</display-name>
4   <welcome-file-list>
5     <welcome-file>index.html</welcome-file>
6     <welcome-file>index.htm</welcome-file>
7     <welcome-file>/WEB-INF/pages/index.jsp</welcome-file>
8     <welcome-file>default.html</welcome-file>
9     <welcome-file>default.htm</welcome-file>
10    <welcome-file>default.jsp</welcome-file>
11  </welcome-file-list>
12 </web-app>
```


这时看一下 new 出来的这个 jsp，拿到这个先做一件事情，把 ISO-8859-1 改成 utf-8(ISO-8859-1 是西欧字符的标准编码，也就是 Latin1，在这个编码下中文在页面显示时会出现乱码)



The image shows two sequential screenshots of an IDE editor window titled 'index.jsp'. The first screenshot shows the initial code with the following lines:

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 </body>
11 </html>
```

The second screenshot shows the same code after modification, where the charset has been changed to 'utf-8' in all three locations:

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 </body>
11 </html>
```

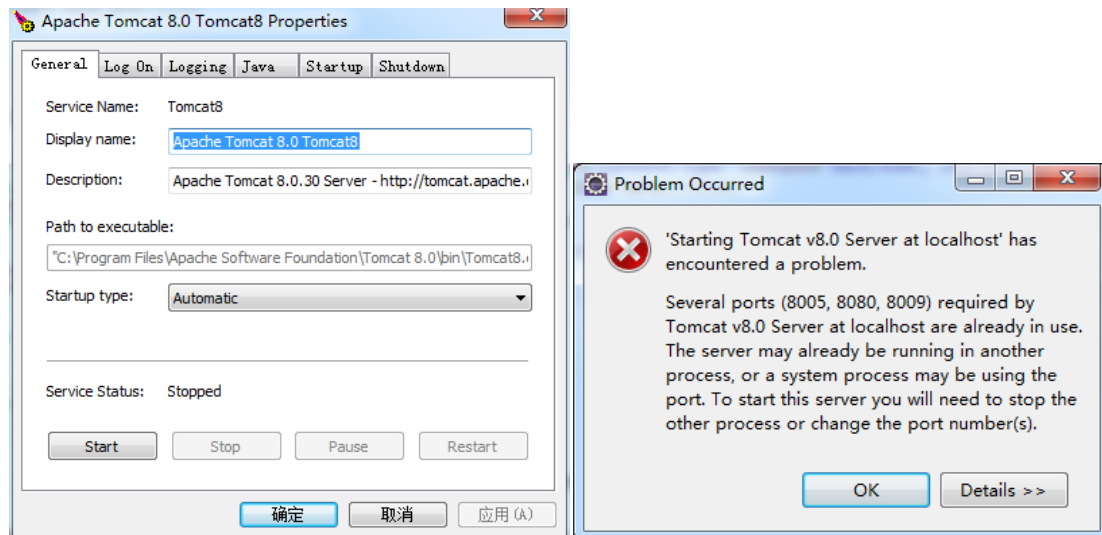
然后就可以在上面写点东西了



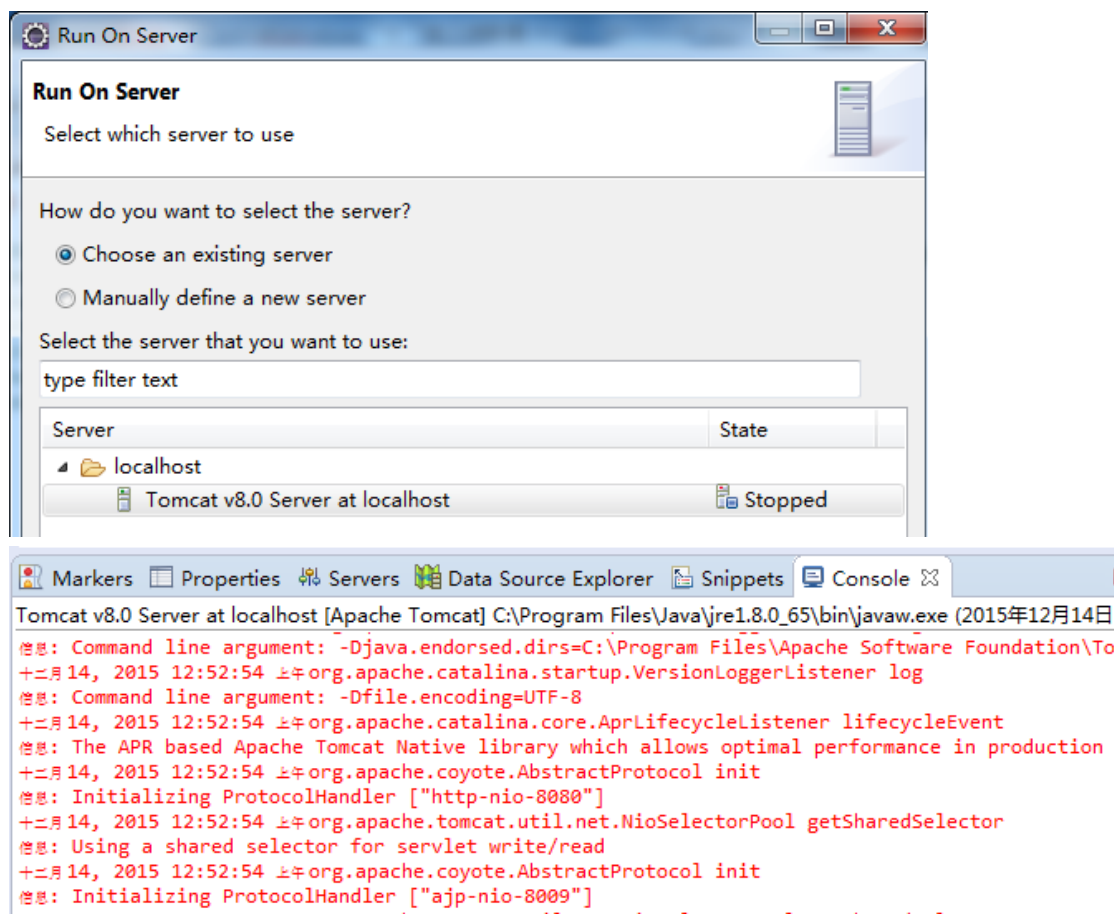
The image shows a screenshot of an IDE editor window titled 'index.jsp'. The code is as follows:

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <h1>
11     <%
12         out.print("hello world");
13     %>
14 </h1>
15 </body>
16 </html>
```

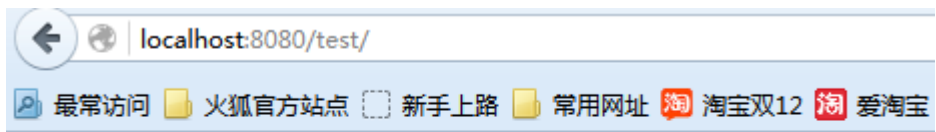
大功告成，可以运行试试了。在运行前需要做一件事情，把刚刚运行的 Tomcat 服务器停掉，以后就通过 eclipse 来启动服务器就可以了。如果不做这一步，等会在运行时 eclipse 也会尝试建立一个服务器占据 8080 端口，但结果发现这个端口上的服务器已经存在了。因此会报出端口已被占用的错误。实际上这也有一种解决办法，改一下 eclipse 的默认端口或者 Tomcat 的端口就可以了。



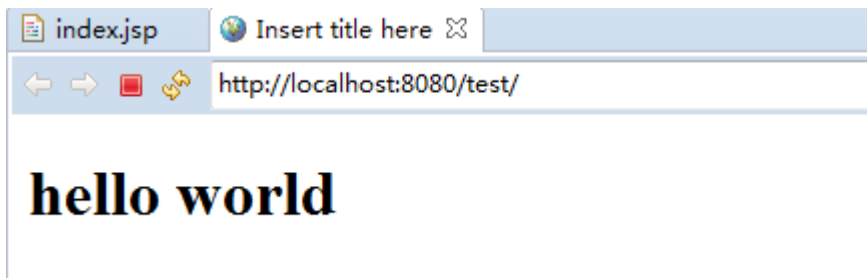
工程右键选到 run as 点击 run on server, 然后就出现了下面这个, 直接 finish 就可以了。然后系统就会在默认的 8080 端口上建立并运行 http 服务器(另外也开了一个 8009 端口用于 ajp 协议, 由于一般上网用的都是 http 协议因此这个用的不多, 具体的你们可以自己百度了解一下)



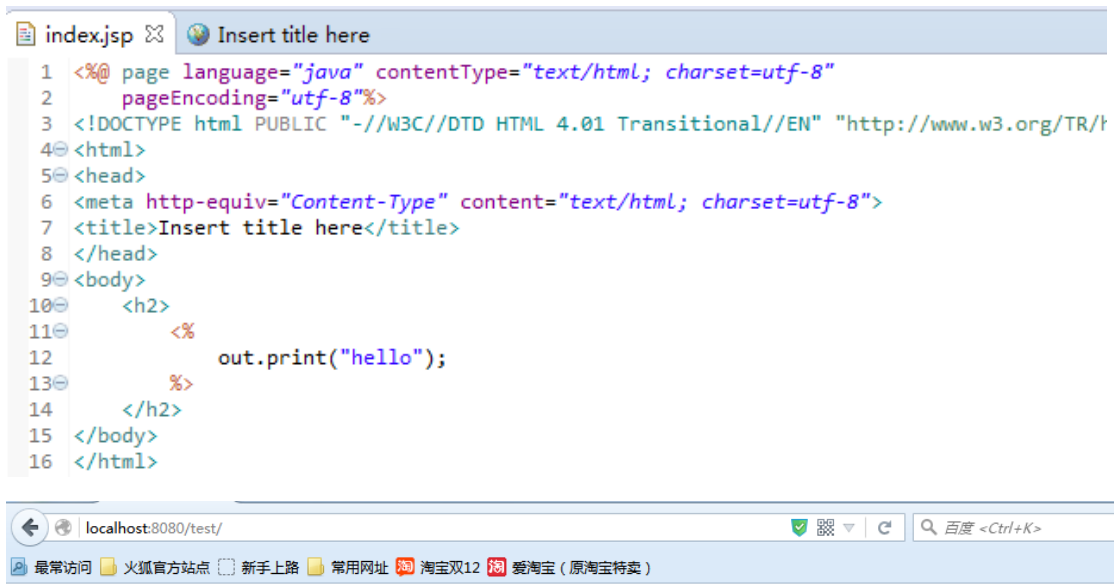
这时候我们在浏览器和 eclipse 内置的浏览器插件上就可以访问到我们刚刚的这个页面了(这里面的这个 test 就是我们刚刚那个工程的名字)



hello world



这时候如果要修改代码不需要停止服务器，改完后保存直接刷新页面就好了，浏览器会自动发起请求重新获取的。如果要在刚刚的端口上重开服务器也行，但前提是要把原来那个占用端口的服务器先停掉(按一下 console 右边的红色方框)



hello

查看器	控制台	调试器	样式编辑器	性能	网络			
方法	文件	域名	类型	已传输	大小	0 毫秒		
200 GET	/test/	localhost:8080	html	0.28 KB	0.28 KB	6 ms		