

[← JSP 标准标签库 \(JSTL \)](#)

Eclipse JSP/Servlet 环境搭建

本文假定你已安装了 JDK 环境，如未安装，可参阅 [Java 开发环境配置](#)。

我们可以使用 Eclipse 来搭建 JSP 开发环境，首先我们分别下载一下软件包：

- **Eclipse J2EE** : <http://www.eclipse.org/downloads/>
- **Tomcat** : <http://tomcat.apache.org/download-70.cgi>

Tomcat 下载安装

你可以根据你的系统下载对应的包(以下以Window系统为例)：

8.0.14

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

Binary Distributions

- Core:
 - [zip \(pgp, md5\)](#)
 - [tar.gz \(pgp, md5\)](#)
 - [32-bit Windows zip \(pgp, md5\)](#)
 - [64-bit Windows zip \(pgp, md5\)](#)
 - [64-bit Itanium Windows zip \(pgp, md5\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, md5\)](#)
- Full documentation:
 - [tar.gz \(pgp, md5\)](#)
- Deployer:
 - [zip \(pgp, md5\)](#)
 - [tar.gz \(pgp, md5\)](#)
- Extras:
 - [JMX Remote jar \(pgp, md5\)](#)
 - [Web services jar \(pgp, md5\)](#)
 - [JULI adapters jar \(pgp, md5\)](#)
 - [JULI log4j jar \(pgp, md5\)](#)
- Embedded:
 - [tar.gz \(pgp, md5\)](#)
 - [zip \(pgp, md5\)](#)

Source Code Distributions

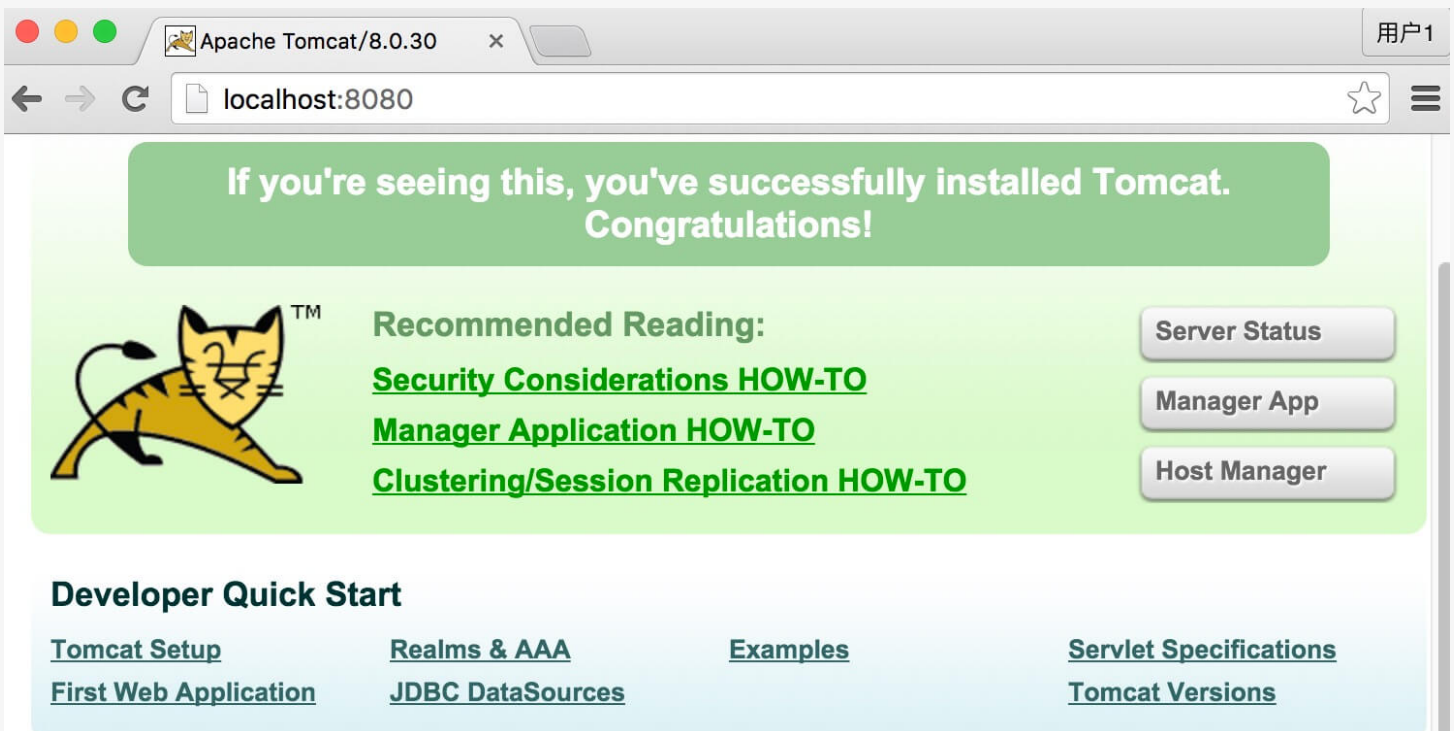
- [tar.gz \(pgp, md5\)](#)
- [zip \(pgp, md5\)](#)

下载之后，将压缩包解压到D盘（你可以自己选择）：


```
Tomcat
03-Oct-2014 18:26:59.787 INFO [localhost-startStop-1] org.apache.catalina.startup
p.HostConfig.deployDirectory Deploying web application directory D:\apache-tomca
t-8.0.14\webapps\host-manager
03-Oct-2014 18:26:59.824 INFO [localhost-startStop-1] org.apache.catalina.startup
p.HostConfig.deployDirectory Deployment of web application directory D:\apache-t
omcat-8.0.14\webapps\host-manager has finished in 37 ms
03-Oct-2014 18:26:59.826 INFO [localhost-startStop-1] org.apache.catalina.startup
p.HostConfig.deployDirectory Deploying web application directory D:\apache-tomca
t-8.0.14\webapps\manager
03-Oct-2014 18:26:59.873 INFO [localhost-startStop-1] org.apache.catalina.startup
p.HostConfig.deployDirectory Deployment of web application directory D:\apache-t
omcat-8.0.14\webapps\manager has finished in 47 ms
03-Oct-2014 18:26:59.873 INFO [localhost-startStop-1] org.apache.catalina.startup
p.HostConfig.deployDirectory Deploying web application directory D:\apache-tomca
t-8.0.14\webapps\ROOT
03-Oct-2014 18:26:59.940 INFO [localhost-startStop-1] org.apache.catalina.startup
p.HostConfig.deployDirectory Deployment of web application directory D:\apache-t
omcat-8.0.14\webapps\ROOT has finished in 67 ms
03-Oct-2014 18:26:59.944 INFO [main] org.apache.coyote.AbstractProtocol.start St
arting ProtocolHandler ["http-apr-8080"]
03-Oct-2014 18:26:59.953 INFO [main] org.apache.coyote.AbstractProtocol.start St
arting ProtocolHandler ["ajp-apr-8009"]
03-Oct-2014 18:26:59.956 INFO [main] org.apache.catalina.startup.Catalina.start
Server startup in 1408 ms
```

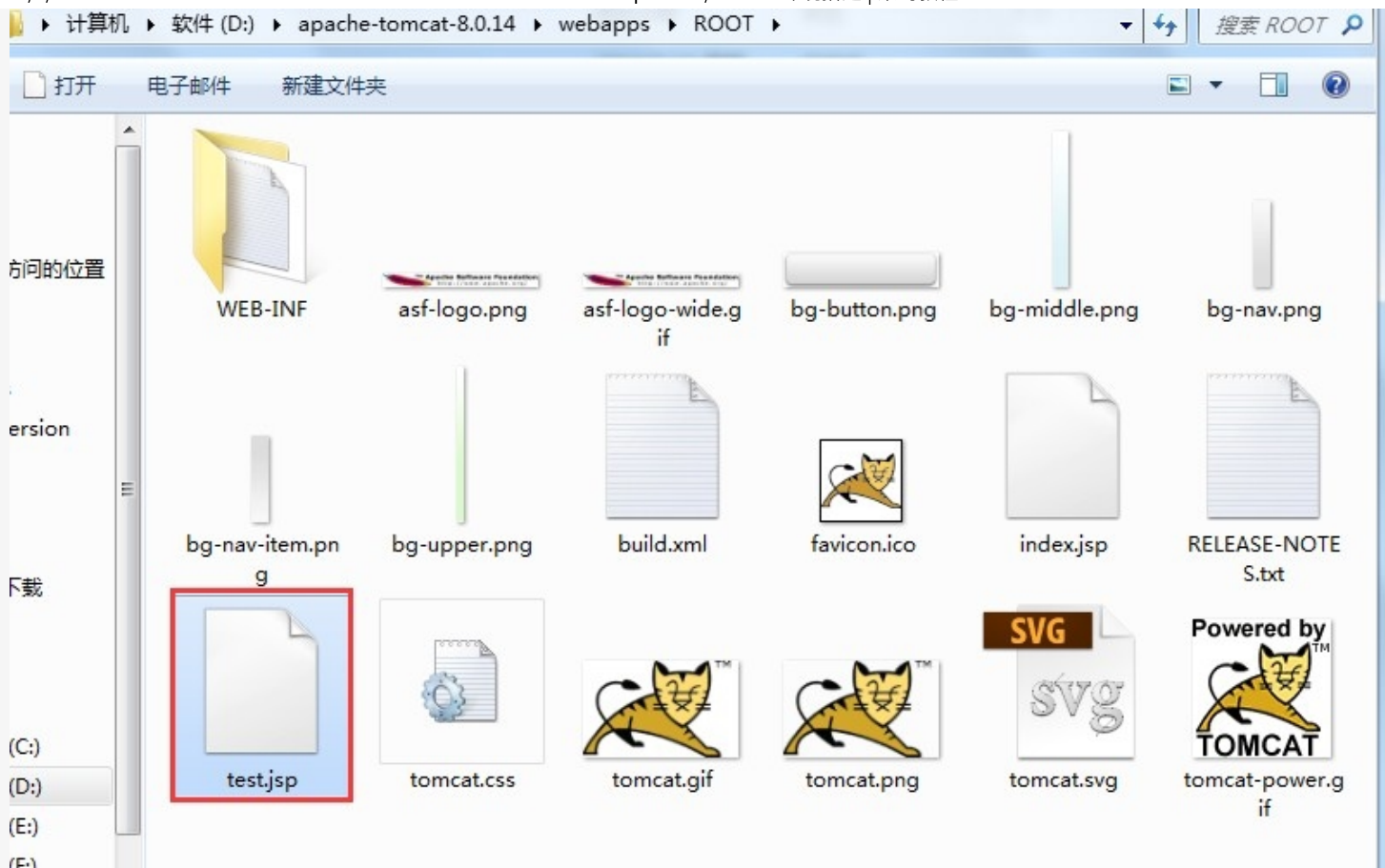
这个时候，本地的服务器就已经搭建起来了。如果想关闭服务器，可以直接关闭上面的窗口，或者在里面输入Ctrl+C禁止服务。

接着我们在浏览器中输入 <http://localhost:8080/>，如果弹出如下界面，表示tomcat安装成功并且启动起来了：



我们现在在浏览器上测试一下它吧：

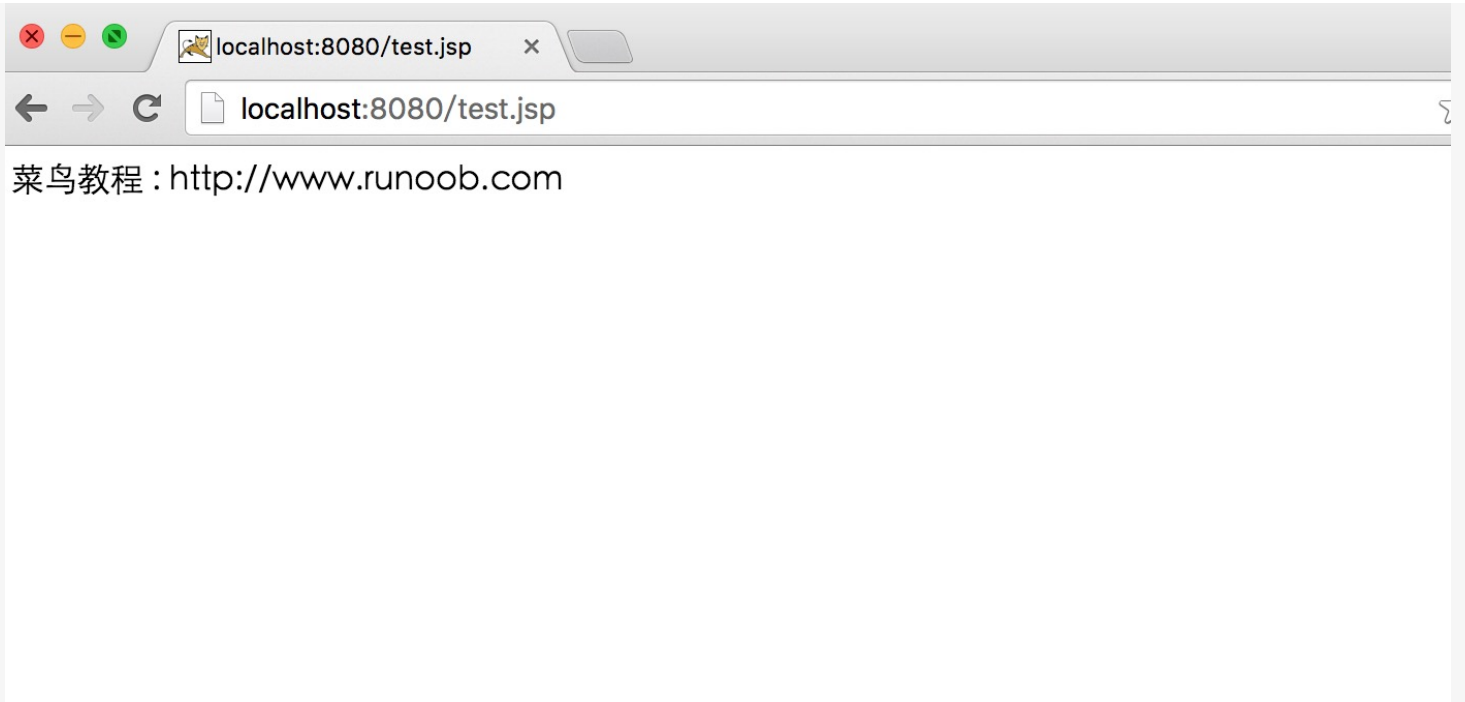
首先在D:\apache-tomcat-8.0.14\webapps\ROOT目录中新建一个jsp文件：



test.jsp 文件代码如下：

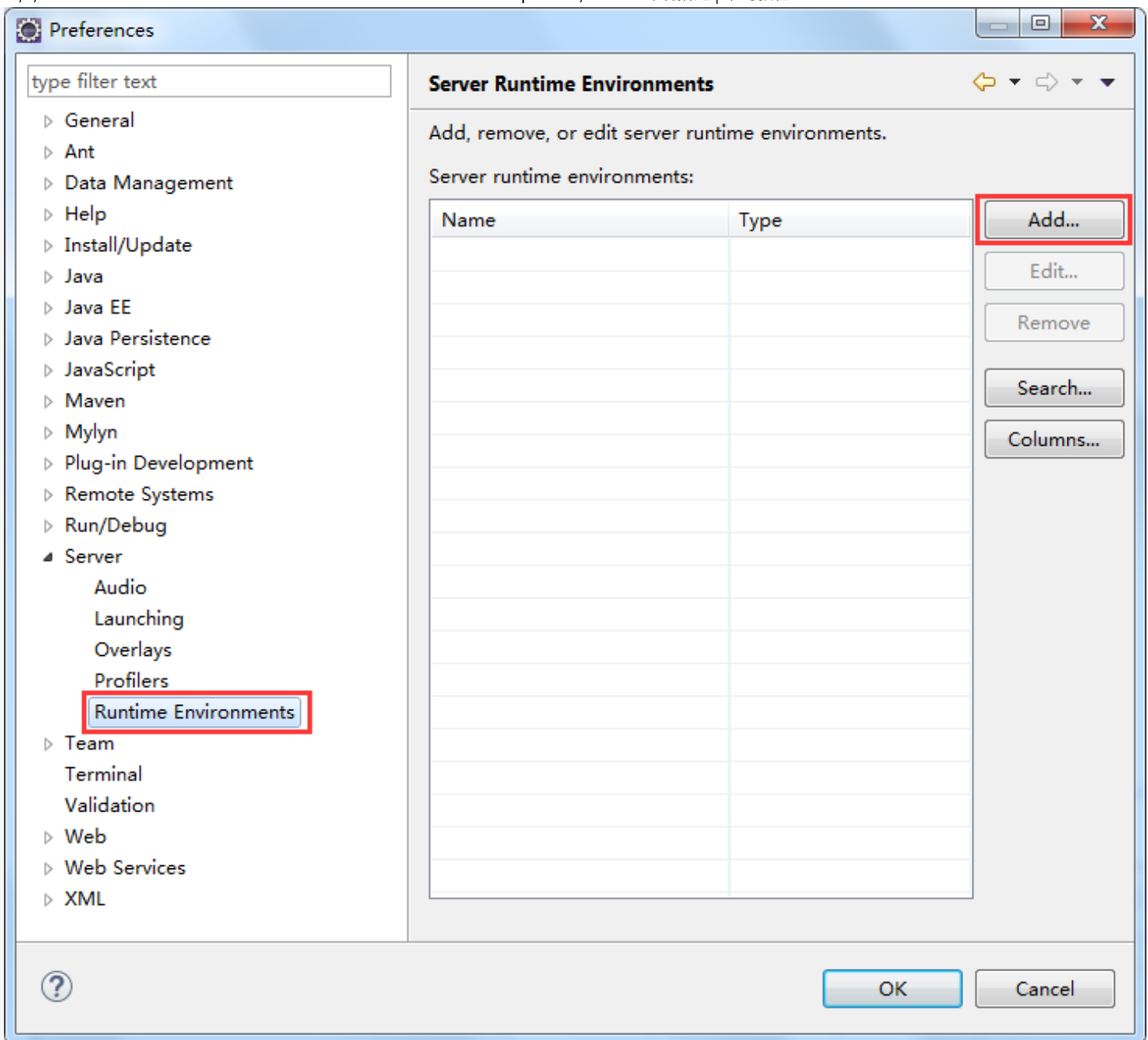
```
<%@ page contentType="text/html; charset=UTF-8" %>
<%
out.print("菜鸟教程 : http://www.runoob.com");
%>
```

接着在浏览器中访问地址 <http://localhost:8080/test.jsp>, 输出结果如下：

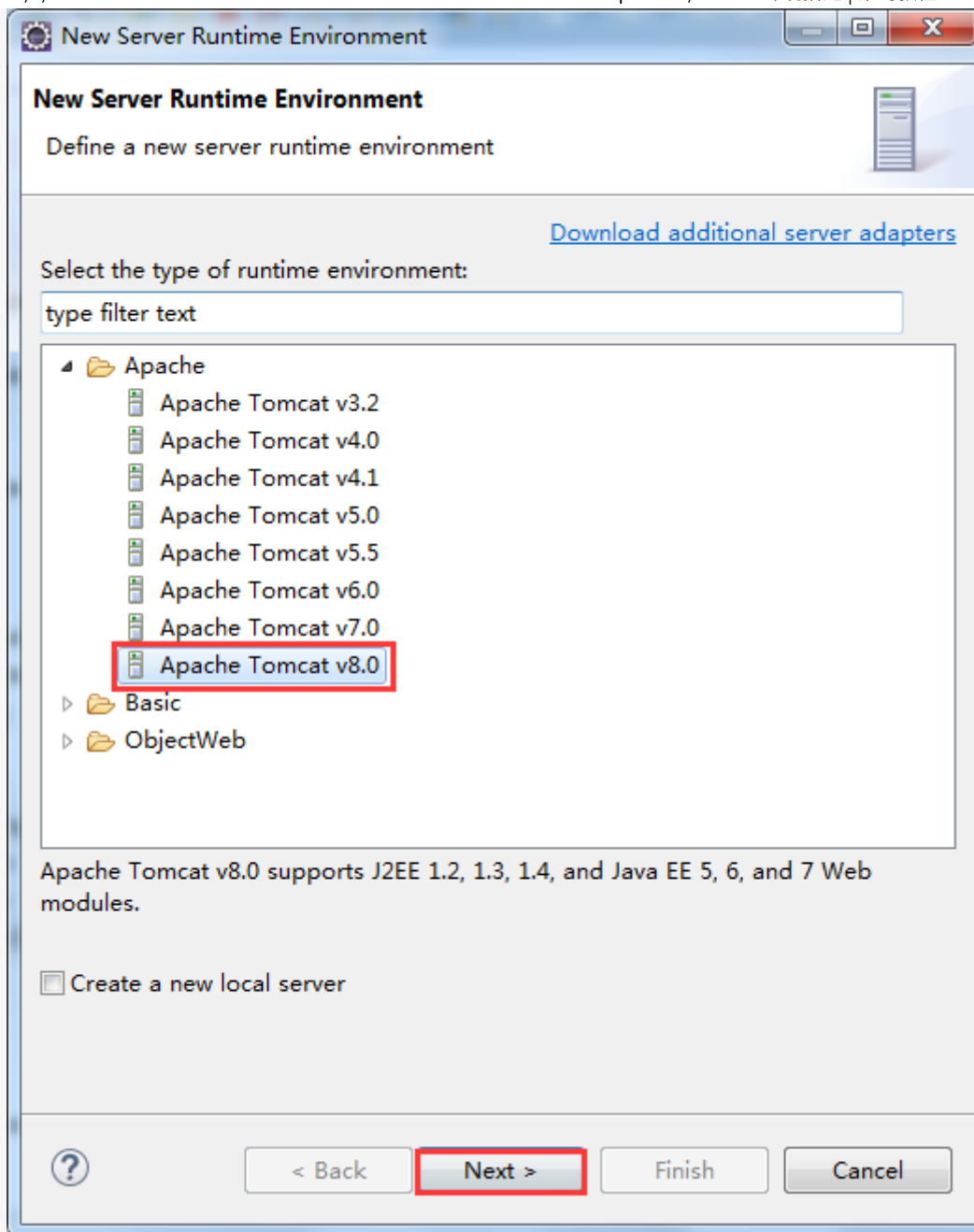


将 Tomcat 和 Eclipse 相关联

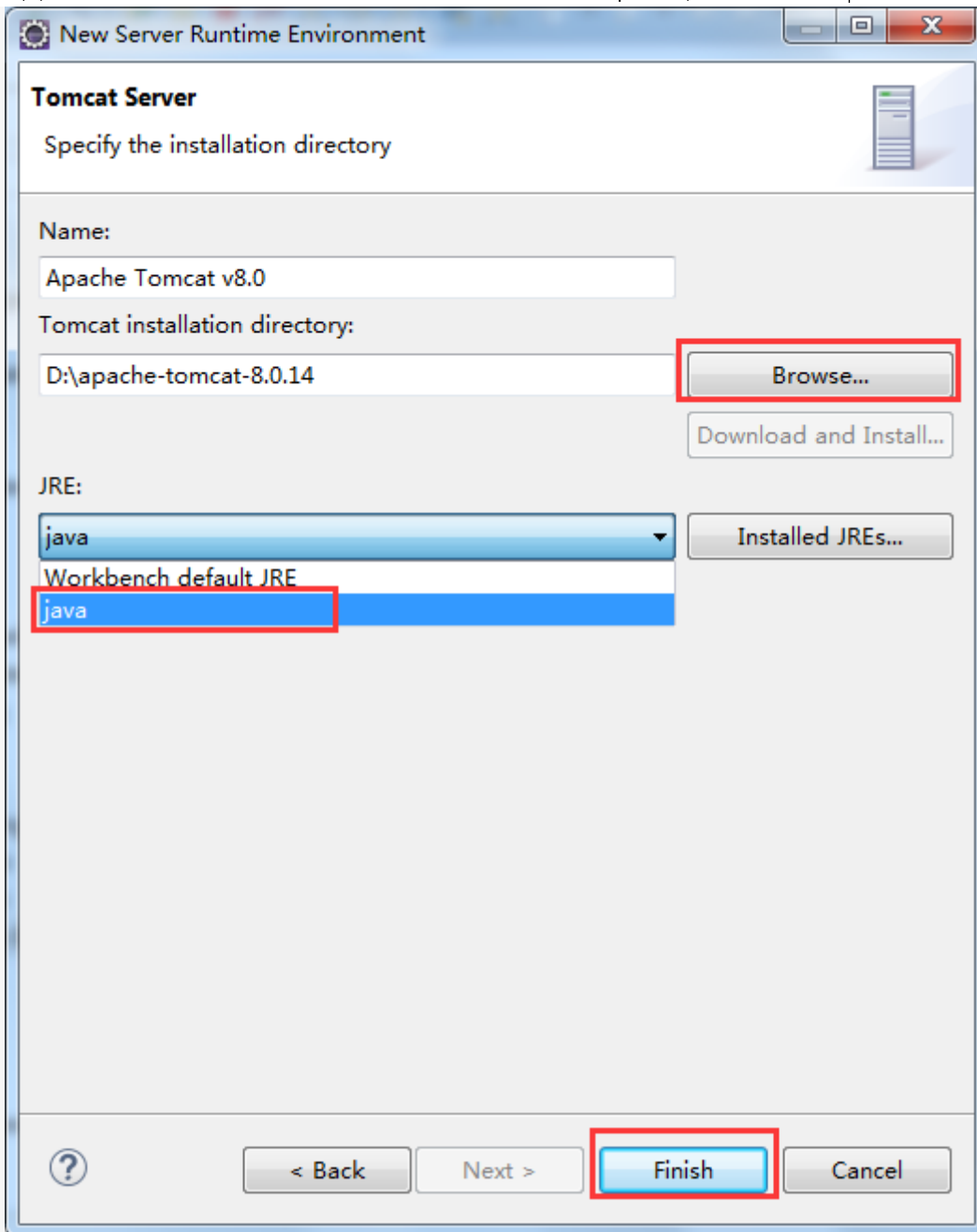
Eclipse J2EE 下载后，解压即可使用，我们打开 Java EE，选择菜单栏 Windows-->preferences (Mac 系统为 Eclipse-->偏好设置)，弹出如下界面：



上图中，点击"add"的添加按钮，弹出如下界面：



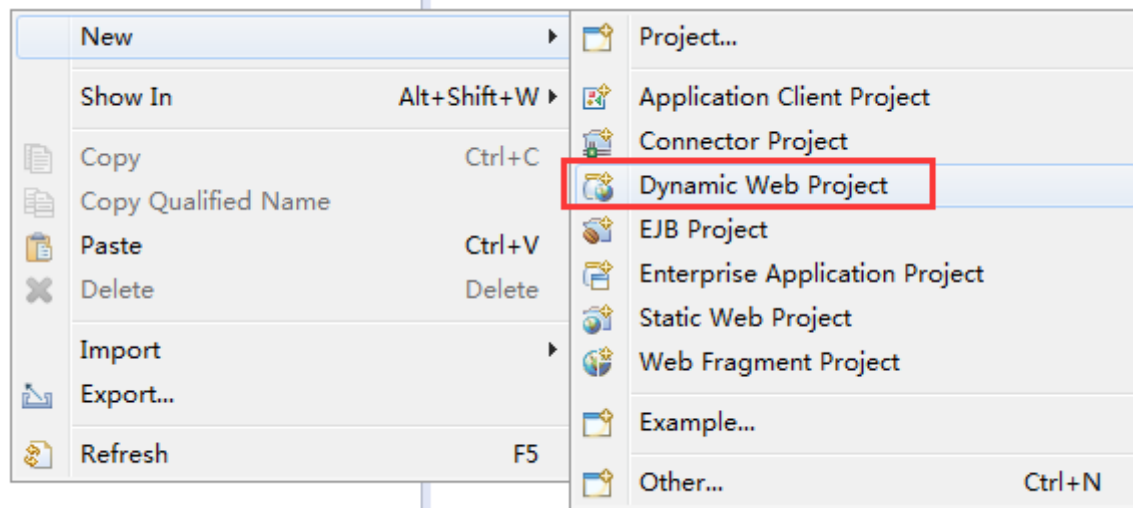
在选项中，我们选择对应的 Tomcat 版本，接着点击 "Next"，选择 Tomcat 的安装目录，并选择我们安装的 Java 环境：

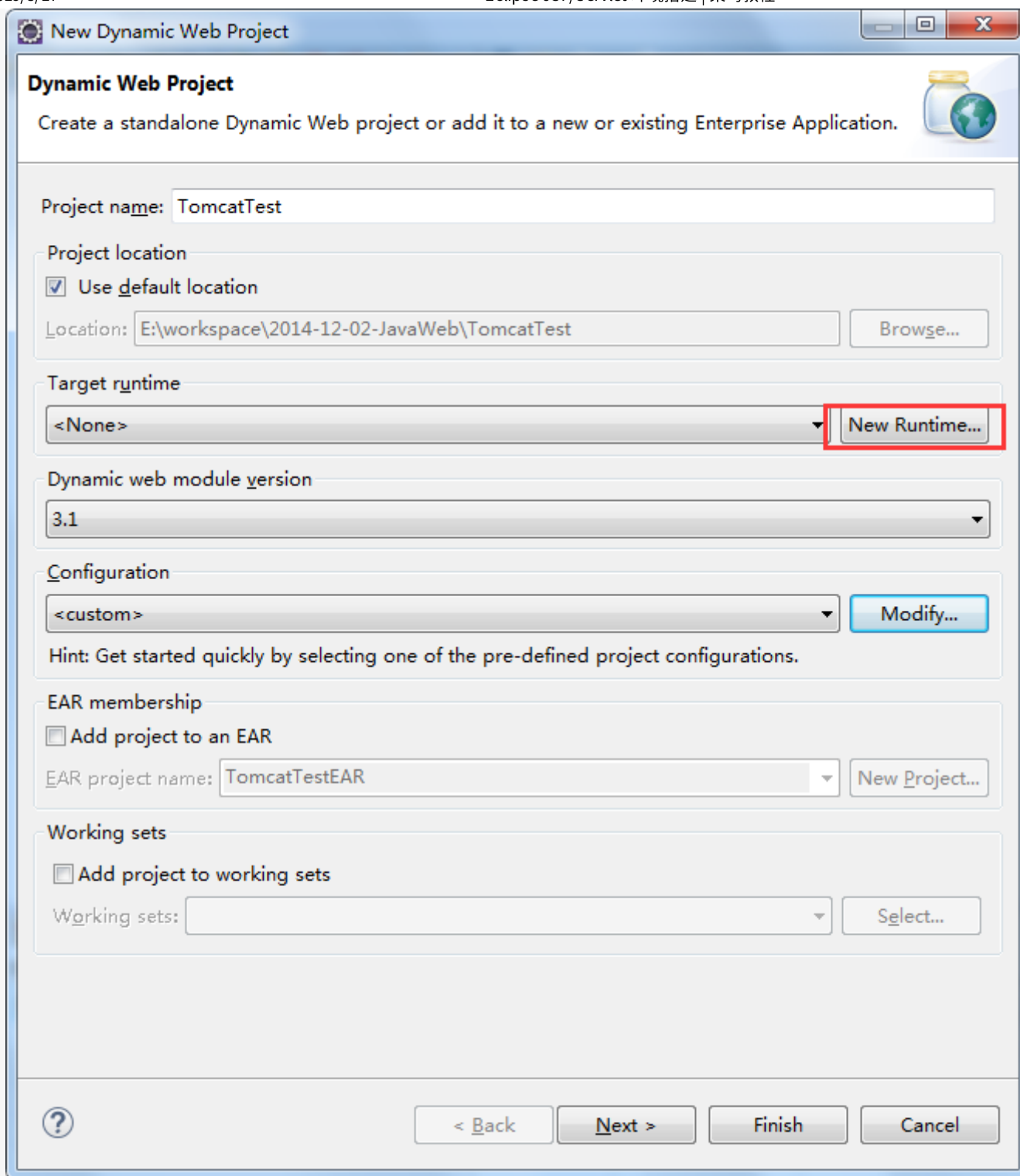


点击 "Finish" , 完成配置。

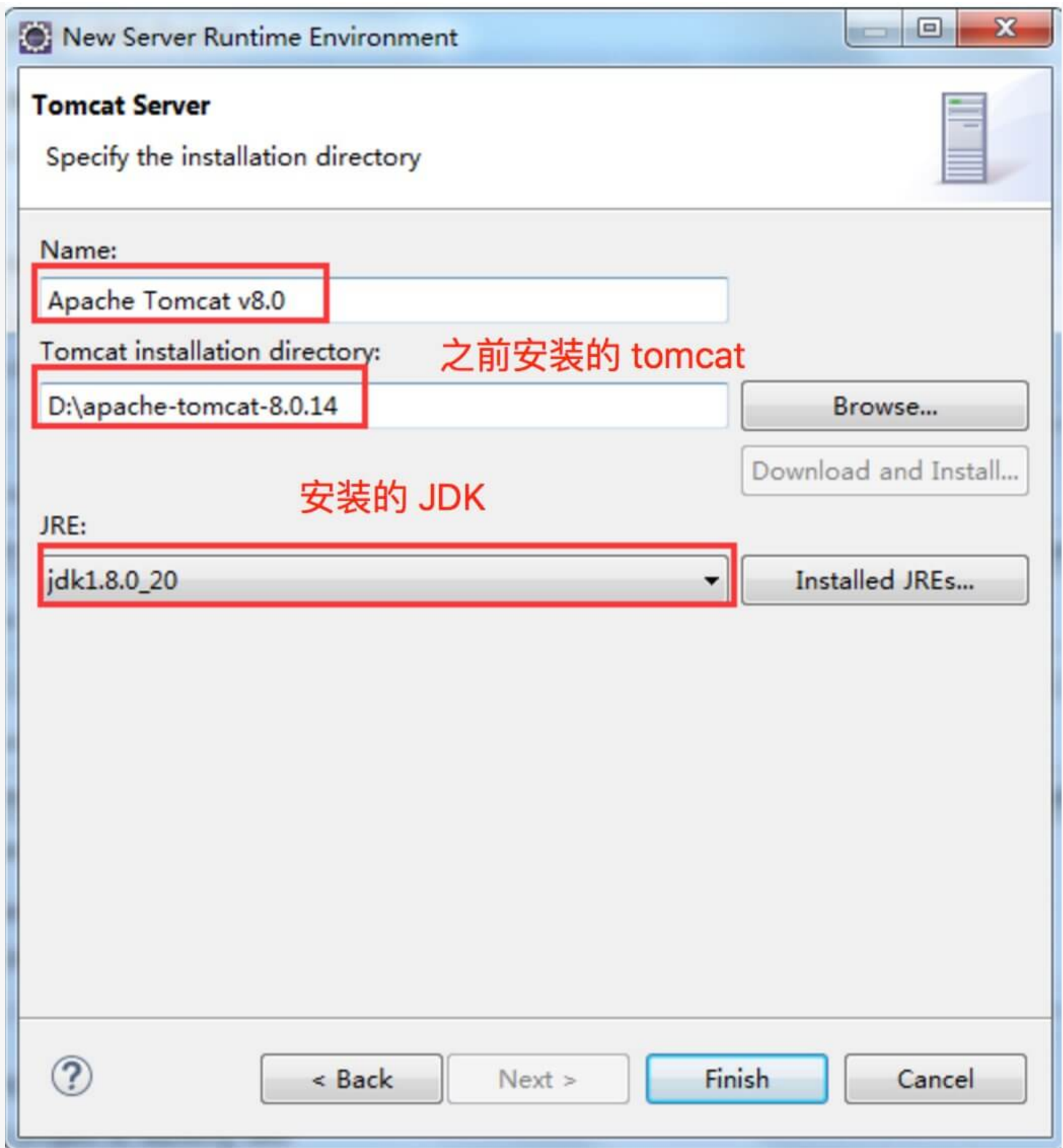
创建实例

选择 "File-->New-->Dynamic Web Project" , 创建 TomcatTest 项目 :







点开上图中的红框部分，弹出如下界面：



注意如果已默认选择了我们之前安装的 Tomcat 和 JDK 则可跳过此步。

然后，单击finish, 继续：

 New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: TomcatTest

Project location

☒ Use default location

Location: E:\workspace\JavaEE\TomcatTest Browse...

Target runtime

Apache Tomcat v8.0 New Runtime...

Dynamic web module version

3.1

Configuration

Default Configuration for Apache Tomcat v8.0 Modify...

A good starting point for working with Apache Tomcat v8.0 runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership


☐ Add project to an EAR

EAR project name: EAR New Project...

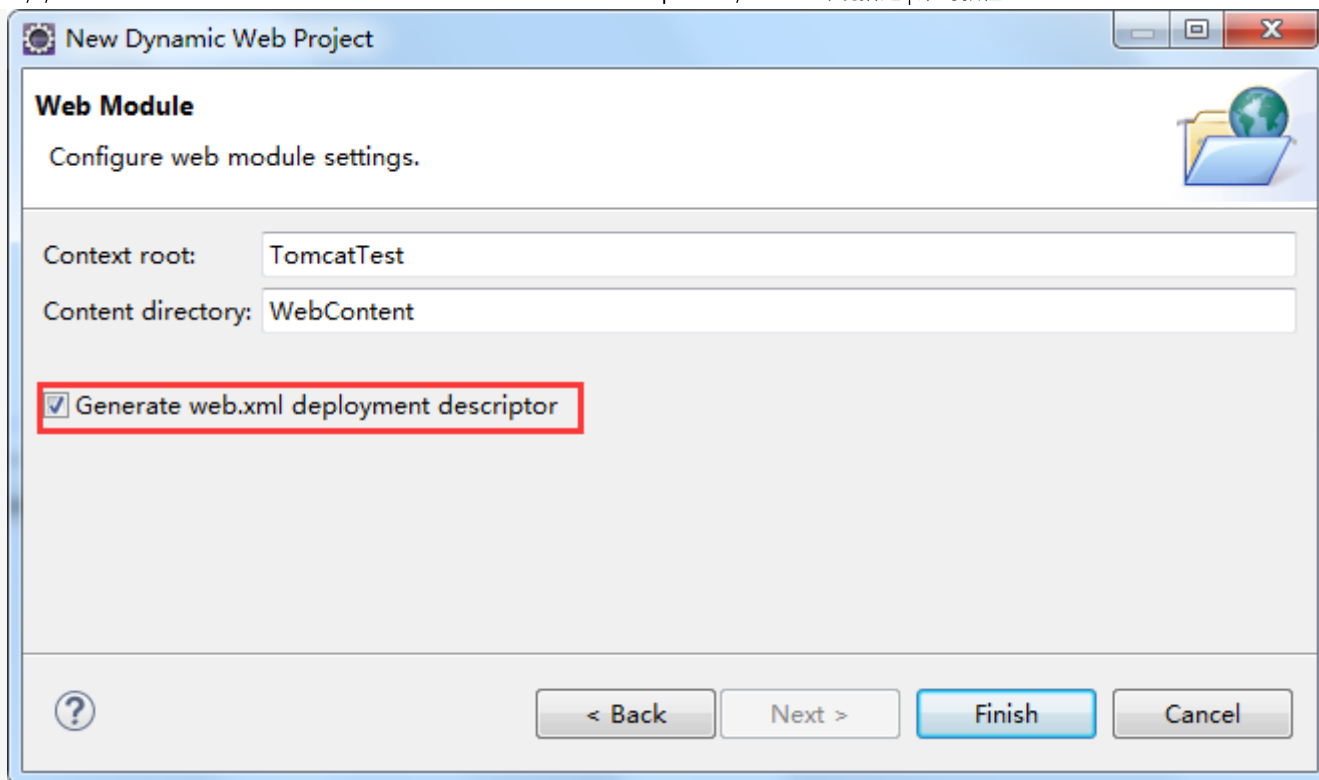
Working sets

☐ Add project to working sets

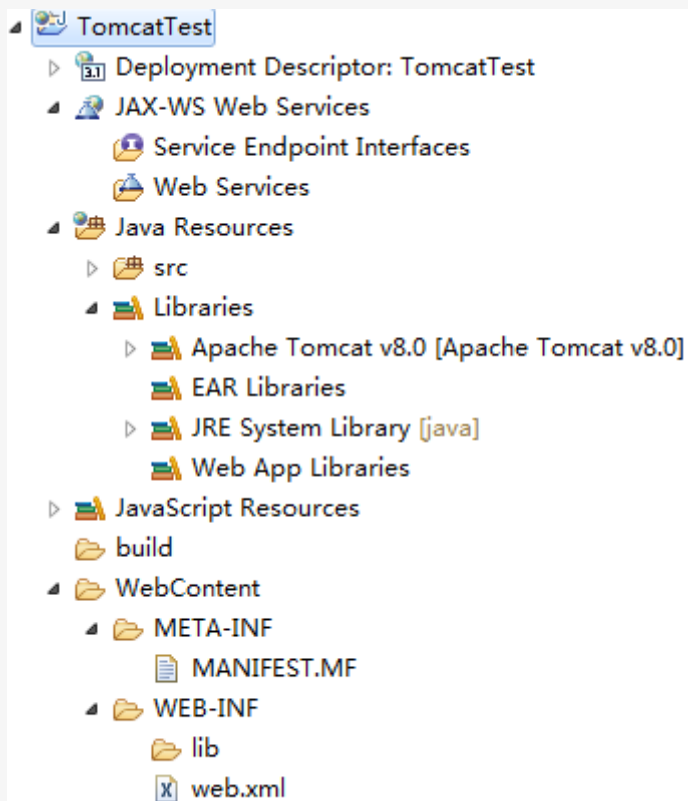
Working sets: Select...



< Back Next > Finish Cancel



工程文件结构：



上图中各个目录解析：

- deployment descriptor：部署的描述。
- Web App Libraries：自己加的包可以放在里面。
- build：放入编译之后的文件。

- WebContent:放进写入的页面。

在WebContent文件夹下新建一个test.jsp文件。在下图中可以看到它的默认代码：

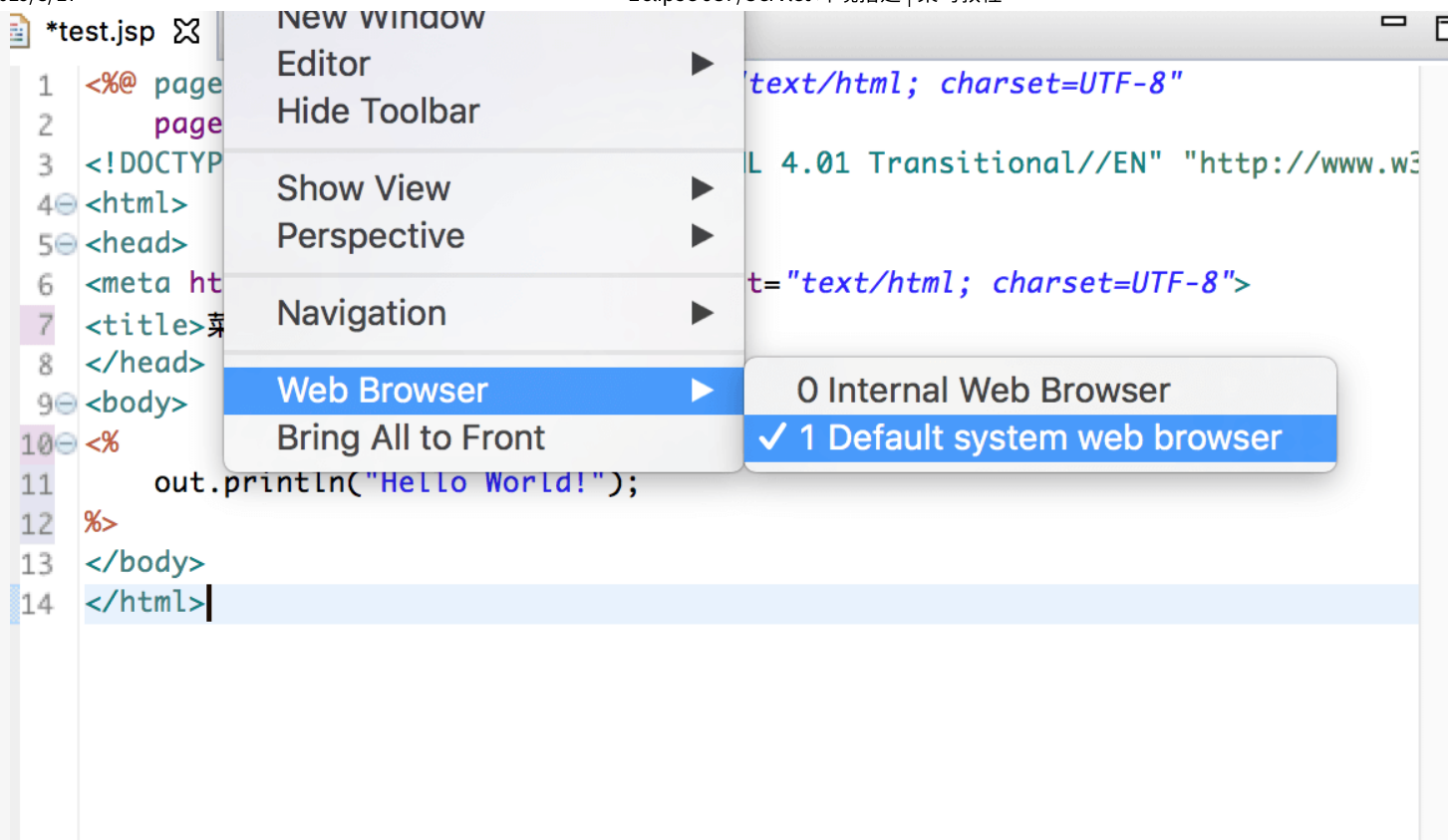
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>

</body>
</html>
```

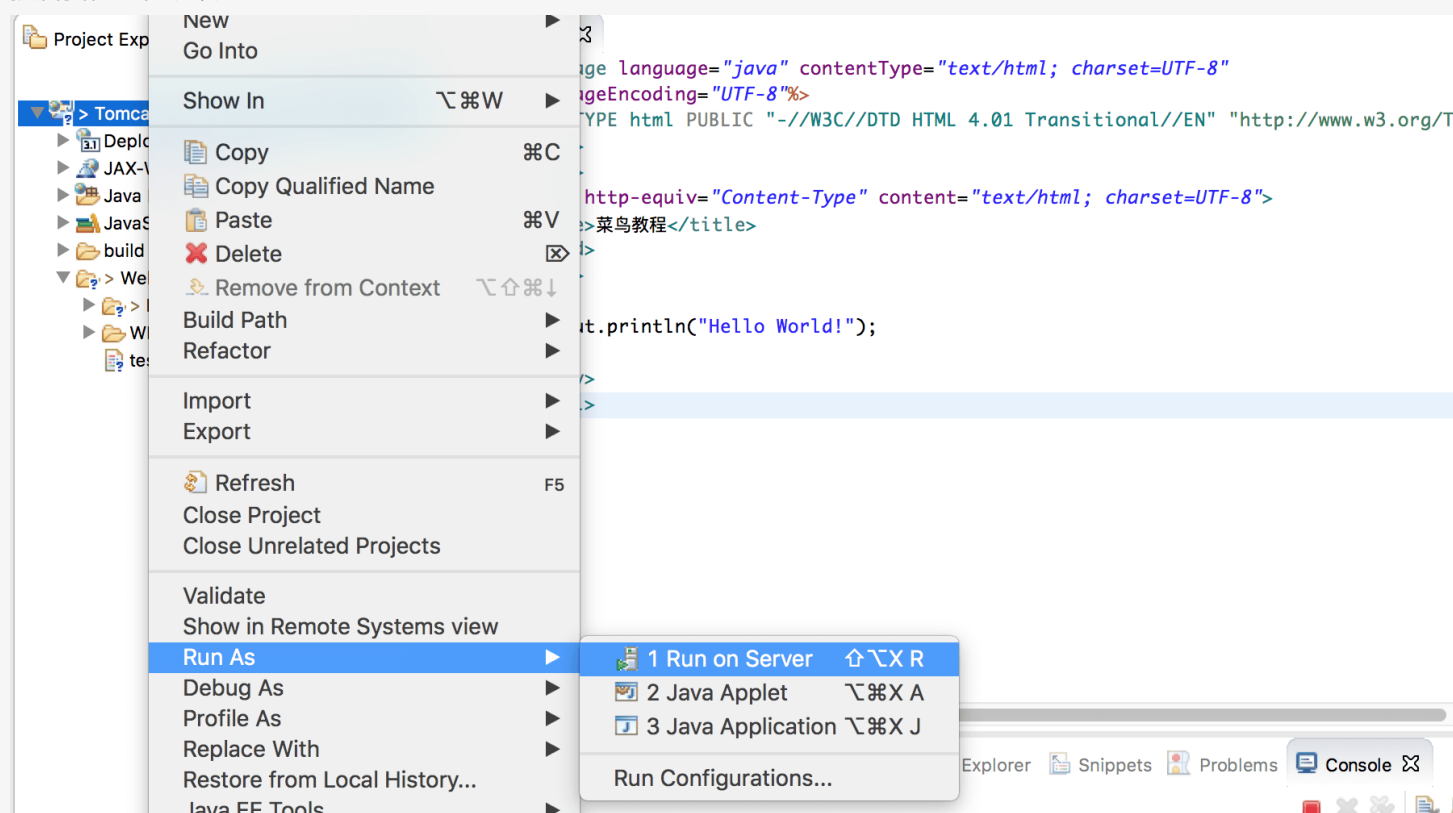
接着我们修改下test.jsp文件代码如下所示：

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>菜鸟教程</title>
</head>
<body>
<%
    out.println("Hello World!");
%>
</body>
</html>
```

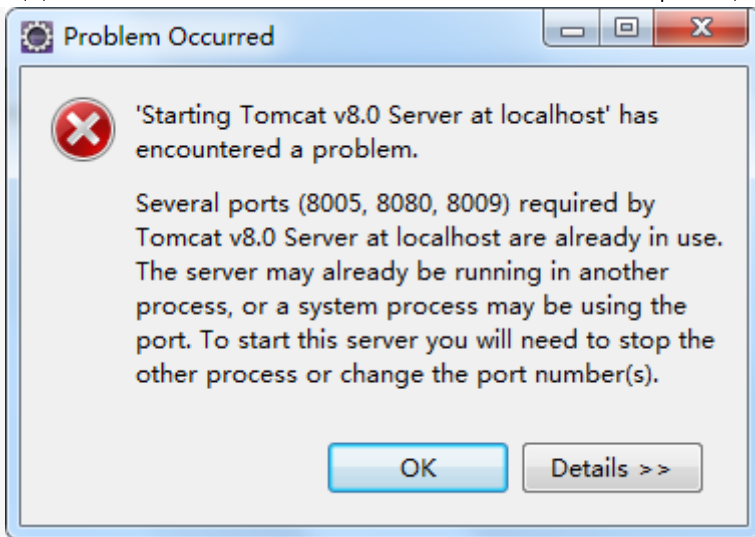
程序运行之前，我们先修改一下浏览器选项：



接着我们运行该项目:



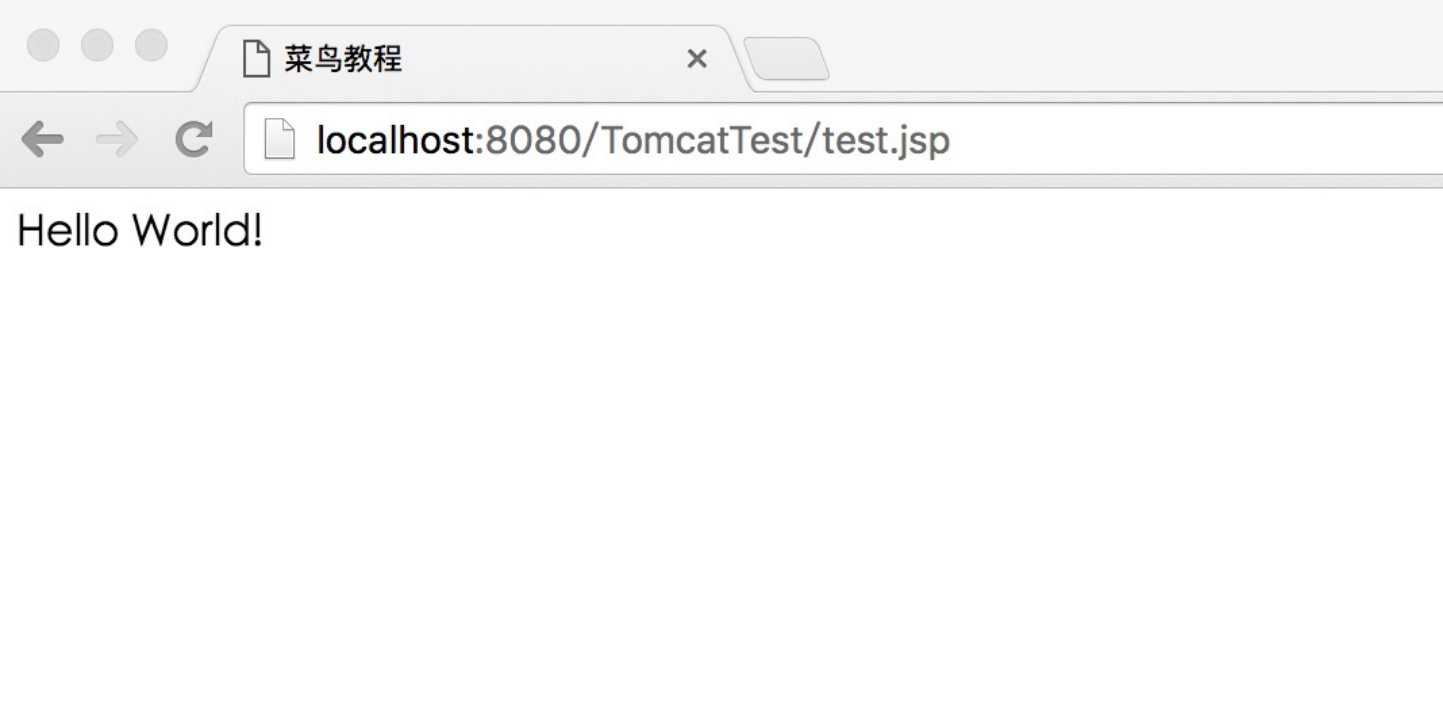
运行时，弹出如下错误：(如果没有此错误，请忽略)



原因是，我们之前点击了Tomcat安装包中的startup.bat，这样一来就手动打开了Tomcat服务器，这明显是多余的，因为程序运行时，eclipse会自动开启Tomcat服务器。所以我们先手动关掉tomcat软件，再次运行程序，就行了。控制台信息如下：

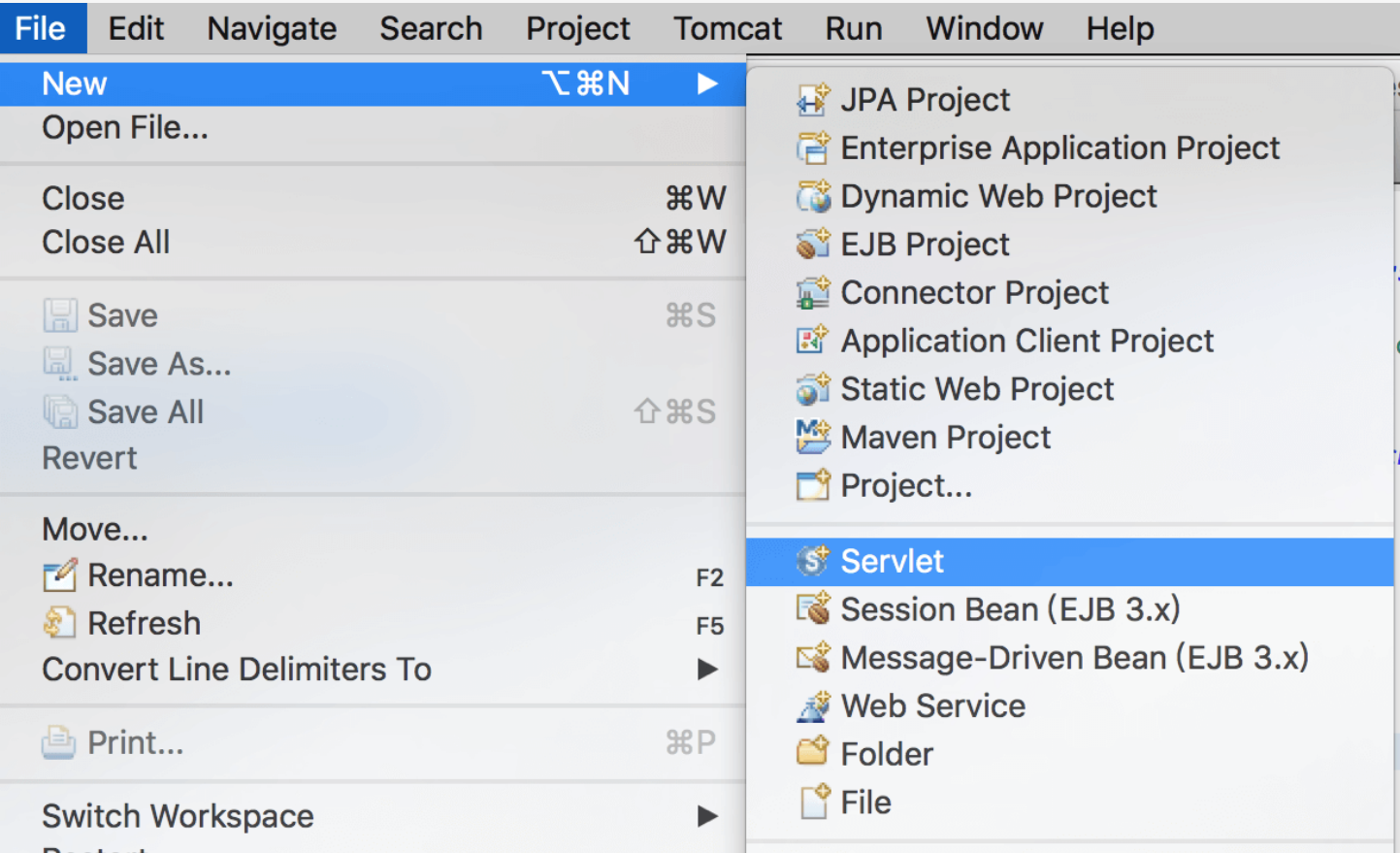
```
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v8.0 Server at localhost [Apache Tomcat] D:\java\bin\javaw.exe (2014年10月23日 下午12:56:51)
信息: Using a shared selector for servlet write/read
十月 23, 2014 12:56:54 下午 org.apache.coyote.AbstractProtocol init
信息: Initializing ProtocolHandler ["ajp-nio-8009"]
十月 23, 2014 12:56:54 下午 org.apache.tomcat.util.net.NioSelectorPool getSharedSelector
信息: Using a shared selector for servlet write/read
十月 23, 2014 12:56:54 下午 org.apache.catalina.startup.Catalina load
信息: Initialization processed in 2062 ms
十月 23, 2014 12:56:54 下午 org.apache.catalina.core.StandardService startInternal
信息: Starting service Catalina
十月 23, 2014 12:56:54 下午 org.apache.catalina.core.StandardEngine startInternal
信息: Starting Servlet Engine: Apache Tomcat/8.0.14
十月 23, 2014 12:56:55 下午 org.apache.catalina.util.SessionIdGeneratorBase createSecureRandom
信息: Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [376] milliseconds.
十月 23, 2014 12:56:55 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["http-nio-8080"]
十月 23, 2014 12:56:55 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-nio-8009"]
十月 23, 2014 12:56:55 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 1005 ms
```

浏览器访问 <http://localhost:8080/TomcatTest/test.jsp>, 即可输出正常结果：



Servlet 实例创建

我们也可以使用以上环境创建 Servlet 文件，选择 "File-->New-->Servlet":




位于 TomcatTest项目的 /TomcatTest/src 目录下创建 "HelloServlet" 类，包为 "com.runoob.test":

Create Servlet



Specify class file destination.

Project: 

Source folder:

Java package:

Class name:

Superclass:

☐ Use an existing Servlet class or JSP

Class name:



HelloServlet.java 代码如下所示：

```
package com.runoob.test;

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class HelloServlet
 */
@WebServlet("/HelloServlet")
```

```
public class HelloServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public HelloServlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // 使用 GBK 设置中文正常显示
        response.setCharacterEncoding("GBK");
        response.getWriter().write("菜鸟教程: http://www.runoob.com");
    }

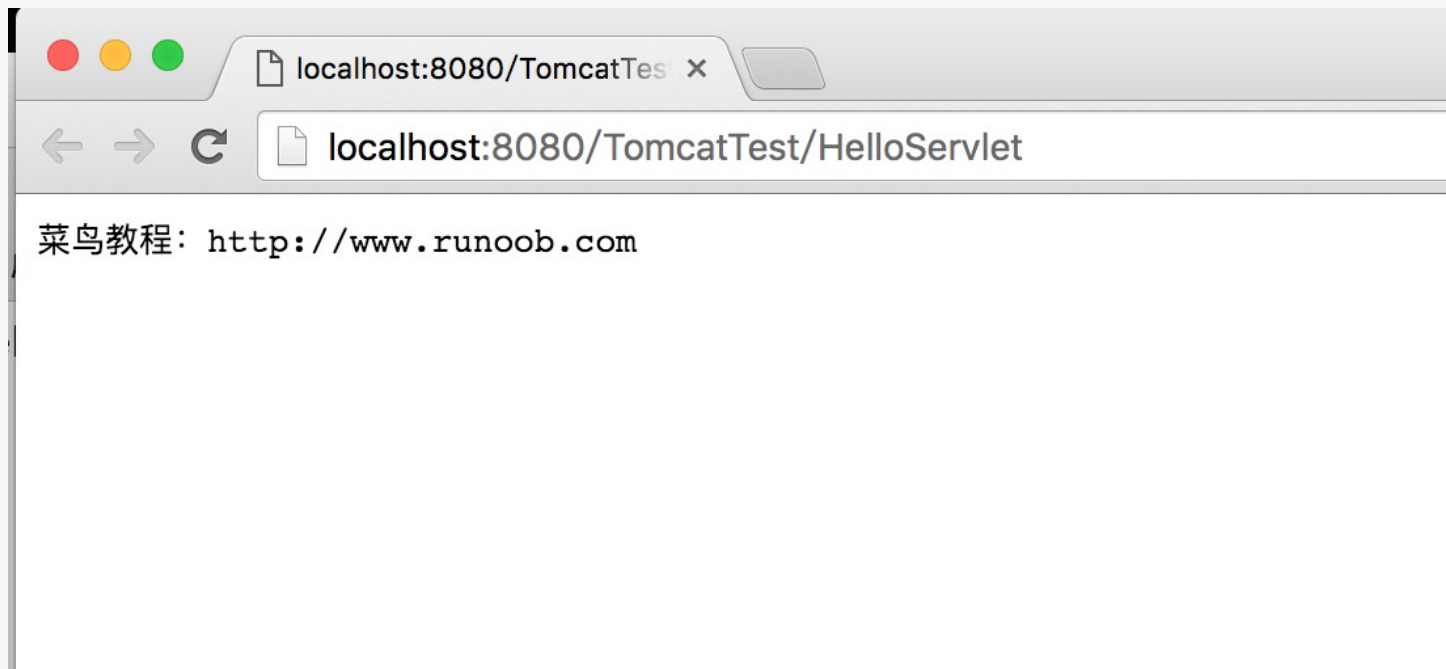
    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}
```

创建 /TomcatTest/WebContent/WEB-INF/web.xml 文件（如果没有），代码如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
        http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
    <servlet>
        <!-- 类名 -->
        <servlet-name>HelloServlet</servlet-name>
        <!-- 所在的包 -->
        <servlet-class>com.runoob.test.HelloServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>HelloServlet</servlet-name>
```

```
<!-- 访问的网址 -->
<url-pattern>/TomcatTest/HelloServlet</url-pattern>
</servlet-mapping>
</web-app>
```

接着重启 Tomcat，浏览器访问 <http://localhost:8080/TomcatTest/HelloServlet>：



参考文章：<http://www.cnblogs.com/smyhvae/p/4046862.html>

← JSP 标准标签库 (JSTL)



2 篇笔记

写笔记



Servlet3.0新特性(得Tomcat7.0版本及以上)，@WebServlet 用于将一个类声明为 Servlet，该注解将会在部署时被容器处理，容器将根据具体的属性配置将相应的类部署为 Servlet。

风不会停息 2年前 [2017-06-29]



按照教程安装的时候发现，我的eclipse中没有“服务区”选项。在网上找到了相关的解决方法，方法如下。

1. ● 在软件eclipse下的Help->InstallNew Software->中，在Work with中输入download。
2. ● 然后安装Web,XML,Java EE and OSGI这个选框。
3. ● 然后一直点击“下一步”，即可完成安装

ludans 5个月前 [10-22]

