

王大东 编著

高等院校程序设计规划教材

# JSP 程序设计

清华大学出版社



王大东 编著

高等院校程序设计规划教材

# JSP 程序设计

清华大学出版社  
北京

## 内 容 简 介

本书介绍在 Eclipse 环境下开发 JSP 应用程序的原理与技术。全书共 11 章,内容包括 JSP 原理及开发环境的搭建、JSP 语法、JSP 内置对象、EL 基本用法、数据库访问、JavaBean、Servlet、MVC 架构、标准标签库及自定义标签、Spring MVC 框架、文件上传下载组件和 Excel 组件的使用方法等。本书将知识点分解、循序渐进将理论知识测试的设计与实现步骤按照由浅入深的方式拆分到各章,有助于读者掌握 JSP 的基础知识、编程技巧及加深对完整的开发体系的理解。

本书结构紧凑,深入浅出,贴近实践,便于教学,可作为高等学校计算机及相关专业本科生教材或教学参考书,也可供网站开发人员参考使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

### 图书在版编目(CIP)数据

JSP 程序设计/王大东编著. —北京:清华大学出版社,2017

(高等院校程序设计规划教材)

ISBN 978-7-302-46005-3

I. ①J… II. ①王… III. ①JAVA 语言—网页制作工具—高等学校—教材 IV. ①TP312.8  
②TP393.092.2

中国版本图书馆 CIP 数据核字(2016)第 316427 号

责任编辑:袁勤勇 战晓雷

封面设计:常雪影

责任校对:白蕾

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质量反馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:18.75

字 数:431千字

版 次:2017年2月第1版

印 次:2017年2月第1次印刷

印 数:1~2000

定 价:39.00元

产品编号:072884-01

清华大学出版社

01

Java 是目前最流行的程序开发语言,在服务器程序设计、手机程序设计等方面应用广泛。Java 作为一种完全的面向对象语言,吸取了其他语言的优点,设计简洁优美,使用起来方便高效。Java 是一个完整的技术平台,在这个平台上不断涌现各种新技术,其中的很多技术是免费的,因此对于中小企业应用来说选择 Java 具有非常大的吸引力。在 Java 发展过程中,不断有过时的技术被弃用。只有那些自身具有强大生命力的技术才会存活下来,并且被不断注入新的活力。JSP(Java Server Pages)就是一种自身具有强大的生命力,一直在快速发展的技术。JSP 是基于 JavaServlet 以及整个 Java 体系的 Web 开发技术,利用这一技术可以建立安全、跨平台的动态网站。JSP 是由 Sun 公司倡导、许多公司参与一起建立的一种动态页面技术标准。自 JSP 推出后,很多公司都推出了支持 JSP 技术的服务器,如 IBM、Oracle 公司等,所以 JSP 迅速成为商业应用的服务器端语言。JSP 具备 Java 技术的简单易用、完全的面向对象、具有平台无关性且安全可靠、面向 Internet 等特点。JSP 从诞生至今,内容越来越丰富,页面设计越来越简洁,是 Internet 上的主流开发工具之一。

全书分为 11 章,内容包括:

第 1 章 概述 包括 Web 应用模型介绍、Tomcat 的配置、Eclipse 开发环境配置、JSP 运行原理、利用表单提交数据、在服务器端处理用户数据。

第 2 章 JSP 的基本语法 包括 JSP 页面的组成、脚本元素,常用的指令和动作的运用。

第 3 章 JSP 内置对象 包括 request、response、cookie、session、application、out 等对象工作机制及应用,结合一个简单的购物车实例讲解各个内置对象的作用范围。

第 4 章 EL 包括表达式语言基础语法和表达式语言的内置对象。

第 5 章 数据库访问 包括创建 MySQL 数据库、JDBC 应用、使用预编译语句和 ResultSet 对象等内容。

第 6 章 JavaBean 包括在 JSP 中使用 JavaBean、利用表单设置 JavaBean 属性、JavaBeans 的 Scope 属性及利用 JavaBean 实现数据库表分页显示。

第 7 章 Servlet 包括 Servlet 基础知识、工作原理、生命周期、Servlet 与客户端通信、Servlet 过滤器和利用过滤器做身份验证。

第 8 章 MVC 架构 包括 MVC 架构的基本原理、用 RequestDispatcher 实现 MVC、MVC 应用实例。

第 9 章 标签库 包括标准标签库和自定义标签。

第 10 章 Spring MVC 包括 Spring 基本特性、Spring 框架、Spring MVC 入门、基于注解的控制器和文件上传。

第 11 章 JSP 实用组件 包括文件上传和下载组件 SmartUpload、处理 Excel 文件组件 JXL。

本书面向具有一定静态页面设计基础的读者,如果未学习过静态页面设计,需要预先学习附录所列内容。附录内容包括 JSP 开发环境安装与配置、常用字符集、HTTP 协议、HTML、CSS、JavaScript。

本书围绕着一个地理知识测试项目,以教程的形式深入浅出、由易到难地介绍 JSP 的常用开发技术。为了便于读者阅读和理解,本书对某些特定内容采用专门的字体。新增的代码或关键代码用加粗字体表示,应该删除的代码打上了删除线。例如,在下面的代码里,将原来的 out.print()方法删除,替换为粗体显示代码。

#### 代码清单 2-3 修改页面内容显示来源 (index.jsp)

```
<del%>out.print("北京是中华人民共和国首都,简称京。");</del%>  
<%  
    i = (i+1) % mQuestions.length;  
    out.print(mQuestions[i].getQuestion());  
%>
```

本书各章内容前后联系比较紧密,例如,在前面章节定义的工具类,在后面会直接调用。读者在阅读时需要按照章节顺序阅读和调试程序,建议不要在章节间跳读。

对于比较简单的项目,实现时可以采用不同的 JSP 技术。为了养成良好的编程习惯,希望读者最后能够使用 MVC 模式设计 Web 应用程序。为了让初学者有一个由浅入深、渐进的学习过程,不同章节的代码设计规范会存在一定的差异。例如,后面章节尽量避免将大量 Java 代码嵌入到 JSP 页面中,而前面的章节并没有遵从这一思想;前几章将 JSP 页面放在网站根目录下,后面又将 JSP 页面改为放在 WEB-INF 目录下。这些差异实际上都与 JSP 程序设计中存在的不同软件设计模式之间存在的差异有关。

作为初学者,没有必要也不可能掌握 JSP 的全部,需要掌握的是软件系统的开发设计思路与语言的核心知识技能。本书在内容选取上没有特别注重知识的完备性。另外,为了突出 JSP 的核心知识,在页面设计上没有大量引入复杂的样式和 JavaScript 脚本。由于篇幅有限,地理知识测试项目部分内容未能在本书中详述,读者可在清华大学出版社网站下载相关电子文档及代码。

白文秀、侯锟、李淑梅、吕凯参与了本书的编写工作。

本书可以作为高等学校计算机及相关专业的 JSP 程序设计教材,也可以作为 JSP 爱好者和网站开发人员的参考书。由于作者水平有限,书中难免存在错漏之处,敬请读者批评指正。

作者

2016 年 10 月

<b>第 1 章 概述</b>	<b>1</b>
1.1 Web 应用基础 .....	1
1.1.1 Web 模型 .....	1
1.1.2 配置 Tomcat .....	2
1.2 JSP 页面 .....	8
1.2.1 创建 JSP 文件 .....	8
1.2.2 运行 JSP 文件 .....	10
1.2.3 JSP 运行机制 .....	11
1.3 用户数据提交与处理 .....	13
1.3.1 利用表单提交数据 .....	13
1.3.2 处理用户提交数据 .....	14
1.3.3 完善输入界面 .....	16
1.3.4 设置首页文件 .....	19
实验 1 .....	20
习题 1 .....	21
<b>第 2 章 JSP 基本语法</b>	<b>23</b>
2.1 脚本元素 .....	23
2.1.1 声明 .....	23
2.1.2 脚本小程序 .....	25
2.1.3 表达式 .....	26
2.2 指令元素 .....	29
2.2.1 page 指令 .....	29
2.2.2 include 指令 .....	34
2.3 动作元素 .....	35
2.3.1 <jsp:param> 动作 .....	36
2.3.2 <jsp:include> 动作 .....	36
2.3.3 <jsp:forward> 动作 .....	37
2.3.4 <jsp:plugin>、<jsp:fallback> 动作 .....	42
2.4 注释 .....	43

实验 2 .....	44
习题 2 .....	44
<b>第 3 章 JSP 内置对象</b>	<b>47</b>
3.1 request 对象 .....	47
3.1.1 request 封装的数据 .....	47
3.1.2 request 对象的主要方法 .....	49
3.2 response 对象 .....	54
3.2.1 response 封装的数据 .....	54
3.2.2 response 对象的主要方法 .....	55
3.2.3 操作 Cookie .....	57
3.3 session 对象 .....	60
3.3.1 session 工作机制 .....	61
3.3.2 session 对象的主要方法 .....	62
3.4 out 对象 .....	66
3.5 application 对象 .....	68
3.6 其他内置对象 .....	70
3.7 简单购物车 .....	71
实验 3 .....	78
习题 3 .....	80
<b>第 4 章 EL</b>	<b>85</b>
4.1 EL 表达式基础 .....	85
4.1.1 EL 语法 .....	85
4.1.2 “[ ]”和“.”操作符 .....	85
4.1.3 运算符 .....	86
4.2 EL 内置对象 .....	86
实验 4 .....	89
习题 4 .....	90
<b>第 5 章 数据库访问</b>	<b>92</b>
5.1 创建 MySQL 数据库 .....	92
5.1.1 创建数据库 .....	92
5.1.2 常用 DML 语句 .....	94
5.2 JDBC 应用概述 .....	95
5.2.1 载入 JDBC 驱动程序 .....	96
5.2.2 定义连接 URL .....	97
5.2.3 建立连接 .....	97
5.2.4 创建 Statement 对象 .....	98

5.2.5	执行查询或更新	98
5.2.6	结果处理	99
5.2.7	关闭连接	100
5.2.8	数据库连接工具类	101
5.3	使用预编译语句	104
5.4	ResultSet 对象	106
5.4.1	读取数据	106
5.4.2	ResultSet 类型	106
5.4.3	元数据	108
实验 5		112
习题 5		114

## 第 6 章 JavaBean 120

6.1	JavaBean 简介	120
6.2	在 JSP 中使用 JavaBean	122
6.3	利用表单设置 JavaBean 属性	127
6.4	JavaBean 的 Scope 属性	133
6.5	JavaBean 应用实例	137
实验 6		141
习题 6		142

## 第 7 章 Servlet 148

7.1	什么是 Servlet	148
7.1.1	编写第一个 Servlet	148
7.1.2	Servlet 工作原理	151
7.1.3	Servlet 生命周期	152
7.2	Servlet 与客户端通信	154
7.2.1	Servlet 生成纯文本	154
7.2.2	Servlet 生成 HTML	155
7.2.3	接收客户提交参数	157
7.2.4	session 对象	159
7.2.5	Servlet 上下文	160
7.2.6	Servlet 的请求转发	161
7.3	Servlet 过滤器	162
7.3.1	创建简单的过滤器	162
7.3.2	Filter 接口	166
7.3.3	Filter 生命周期	167
7.3.4	应用过滤器做身份验证	168
7.4	Servlet 应用实例	169



实验 7 .....	175
习题 7 .....	176
<b>第 8 章 MVC 架构</b> .....	<b>182</b>
8.1 什么是 MVC .....	182
8.2 用 RequestDispatcher 实现 MVC .....	183
8.2.1 定义 JavaBean 来表示数据 .....	184
8.2.2 编写 Servlet 处理请求 .....	184
8.2.3 填写 JavaBean .....	185
8.2.4 结果的存储 .....	185
8.2.5 转发请求到 JSP 页面 .....	186
8.2.6 从 JavaBean 中提取数据 .....	186
8.2.7 目的页面中相对 URL 的解释 .....	187
8.2.8 控制器示例 .....	187
实验 8 .....	194
习题 8 .....	196
<b>第 9 章 标签库</b> .....	<b>200</b>
9.1 标准标签库 .....	200
9.1.1 输出标签 c:out .....	201
9.1.2 迭代标签 .....	202
9.1.3 条件标签 .....	204
9.1.4 变量操作标签 .....	206
9.1.5 URL 相关标签 .....	207
9.2 自定义标签 .....	210
9.2.1 标签处理类 .....	210
9.2.2 标签库描述文件 .....	211
9.2.3 在 JSP 文件使用自定义标签 .....	212
9.2.4 标签属性 .....	212
9.2.5 标签体 .....	215
9.2.6 定制标签应用示例 .....	217
实验 9 .....	220
习题 9 .....	220
<b>第 10 章 Spring MVC</b> .....	<b>223</b>
10.1 Spring 简介 .....	223
10.1.1 基于 POJO .....	223
10.1.2 依赖注入 .....	224
10.1.3 面向切面编程 .....	226

10.1.4	利用模板消除重复代码	226
10.1.5	Bean 容器	226
10.1.6	Spring 框架	230
10.2	Spring MVC 入门	231
10.2.1	搭建 Spring MVC	232
10.2.2	Spring MVC 示例	234
10.3	基于注解的控制器	238
10.3.1	@Controller	238
10.3.2	@RequestMapping	239
10.3.3	利用控制器类实现 QuestionEdit	241
10.3.4	利用注解实现依赖注入	244
10.4	文件上传	247
10.4.1	客户端编程	248
10.4.2	接收上传的文件	248
实验 10		253
习题 10		254

## 第 11 章 JSP 实用组件 255

11.1	文件上传和下载组件	255
11.1.1	SmartUpload 常用类	255
11.1.2	使用 jspSmartUpload 上传	257
11.2	处理 Excel 文件组件	259
11.2.1	jxl 对象	260
11.2.2	基本操作	260
11.2.3	Excel 批量导入数据	263
实验 11		265
习题 11		266

## 习题参考答案 267

## 附录 A JSP 开发环境安装与配置 270

## 附录 B 常用字符集 272

## 附录 C HTTP 协议 274

## 附录 D HTML、CSS、JavaScript 简介 277

## 参考文献 287

# CHAPTER 第 1 章

## 概 述

JSP(JavaServer Pages)是一种基于 Java 的服务器端 Web 页面设计技术,用于创建含有动态生成内容的 Web 页面。JSP 技术功能强大,使用灵活,平台独立,为创建显示动态 Web 内容的页面提供了一个简捷而快速的方法。

### 1.1 Web 应用基础

本节介绍利用 JSP 编写 Web 应用需要掌握的一些基本概念。

#### 1.1.1 Web 模型

万维网(World Wide Web,WWW,简称 Web)是一个结构性的框架,通过万维网可以访问整个 Internet 上数以百万计的计算机中的相互链接的文档。

从用户的观点来看,Web 是由全球范围的文档或 Web 页面(Web page)集合组成的,Web 页面通常简称为页面。Web 页面是用一种称为 HTML(超文本标识语言)的标准语言编写的。每个页面可以包含指向任何其他页面的链接。通过单击一个链接,用户可以跟随这个链接来到它所指定的页面,这个过程可以无限重复下去。Web 一词形象、准确地描述了所有页面之间通过链接形成的这种网状结构。

Web 页面存储在 Internet 中遍布全球的 Web 服务器上。Web 服务器装有能够提供 Web 服务的软件,如 IIS、Apache、Tomcat、JBoss、WebSphere 等服务器软件。与普通的未安装服务器软件的计算机相比,Web 服务器能够对用户的请求做出响应。

用户利用浏览器浏览某台服务器上的某个页面,浏览器发出对该页面的请求,服务器上运行的服务器程序收到该请求后,从磁盘上取到该页面文件(静态页面)或动态生成该页面(动态页面),通过网络发回给用户浏览器,浏览器收到所请求的页面,对它上面的文本和格式命令进行解释,并在屏幕上按正确的格式显示出来。浏览器和服务器(进程)之间发送的数据必须按照双方事先约定好的格式编写,接收方才能知道发送方的数据含义,如请求页面是通过在 GET 后接请求页面标识来实现的,这些约定是在超文本传输协议(HTTP)中规定的。

Web 页面是用 URL(Uniform Resource Locator,统一资源定位符)来标识的,典型的 URL 如下所示:

`http://www.abcd.com:80/index.html`

URL 由协议名称、主机地址、端口号、页面文件名组成。

- (1) 由于 Web 服务中客户机与服务器之间使用 HTTP 传输数据,所以协议名称为 http。
- (2) 主机地址可以是域名,也可以是 IP 地址形式,主机地址指出页面在哪一台机器上。
- (3) 一个端口号对应服务器上正在运行的一个程序,端口号为 80(Web 服务默认端口)时可省略。
- (4) 文件名可以包含路径,路径为相对路径。

例如,在 www.abcd.com 主机上同时装有 IIS 和 Tomcat,IIS 对应的端口号设置为 80, Tomcat 对应的端口号设置为 8080。IIS 服务器网站根目录对应的路径为 c:\website, Tomcat 服务器网站根目录的路径为 d:\tomcat\webapps。标识 c:\website\index.html 和 d:\tomcat\webapps\a\index.html 的 URL 分别为

http://www.abcd.com:80/index.html 或 http://www.abcd.com/index.html

http://www.abcd.com:8080/a/index.html

图 1.1 为利用浏览器通过 URL 访问远程服务器上静态页面的过程。浏览器向服务器发送一个文件名(URL),然后服务器返回这个文件。

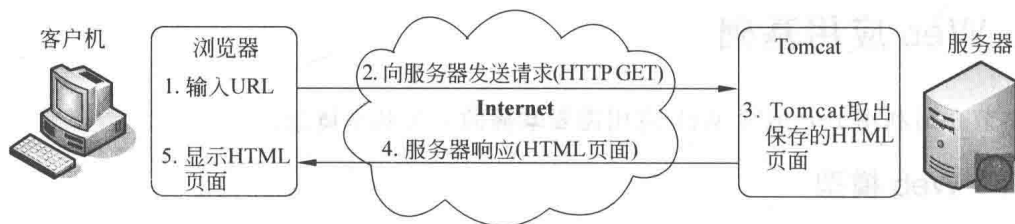


图 1.1 Web 浏览器访问静态页面

现在,越来越多的 Web 内容变成动态的,也就是说,它们不是被存储在磁盘上,而是根据需要生成。例如,在使用 JSP 设计的电子商务网站查找内容时,浏览器将要查找的内容交给服务器,服务器将查找内容交给程序处理,在服务器磁盘的数据库中找到满足条件的记录,然后生成一个定制的 HTML 页面返回给客户。如图 1.2 所示。



图 1.2 Web 浏览器访问动态页面

### 1.1.2 配置 Tomcat

开发 Web 应用,首先需要安装和配置服务器。Tomcat 安装后(参见附录 A),其目录结构如图 1.3 所示。将 Web 应用文件部署到 Tomcat 上,通常要经过如下步骤:

- (1) Web 应用发布(存储)。将 Web 应用文件发布到 Tomcat,可选择这种方法:将 Web 应用复制到 webapps 目录下;将要部署的 Web 应用发布文件(.war)复制到 webapps 目录下,在 Tomcat 启动时自动发布;将 Web 应用存放在其他位置,在 server.xml 中做额外的配置。

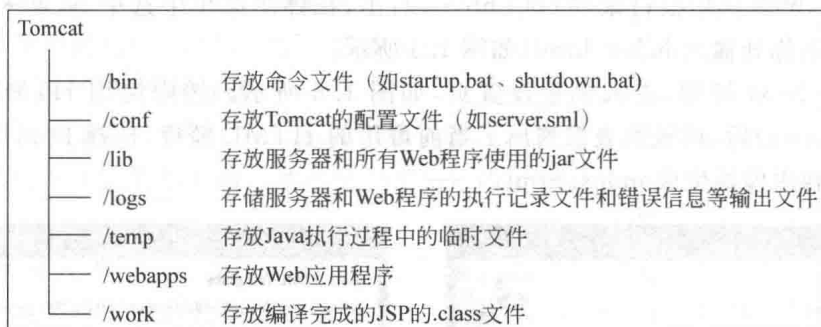


图 1.3 Tomcat 目录结构

(2) 修改服务器配置文件 server.xml, 如更改端口号、配置虚拟网站根目录、允许外部主机访问等。每次修改后, 需要重新启动 Tomcat。

(3) 运行 bin 目录下的 startup.bat, 启动 Tomcat, 测试 Web 应用。

Web 应用的发布可以手工布置, 也可以在 Eclipse 中配置服务器后自动发布。

### 1. 在 Eclipse 环境下建立 Web 项目

本章编写的页面用于测试用户的地理知识。Web 应用包括页面、样式、脚本等各种类型的文件, Eclipse 以项目文件方式管理这些文件。首先创建一个 Web 项目。

(1) 启动 Eclipse 程序, 工作区设置为 d:\workspaces。

(2) 选择 File→New→Dynamic Web Project 菜单项, 打开新建动态 Web 项目窗口创建一个新的项目。在项目名称(Project Name)处输入 GeoTest, Dynamic web module version 选择 3.0, 其他项取默认设置, 单击 Next 按钮。在 Java 代码设置窗口保持默认设置 (Java 代码目录设为 src) 不变, 单击 Next 按钮。在 Web module 窗口选择 Generate web.xml deployment descriptor, 单击 Finish 按钮, Eclipse 完成创建并打开新的项目。在项目资源管理器中展开项目, 如图 1.4 所示。

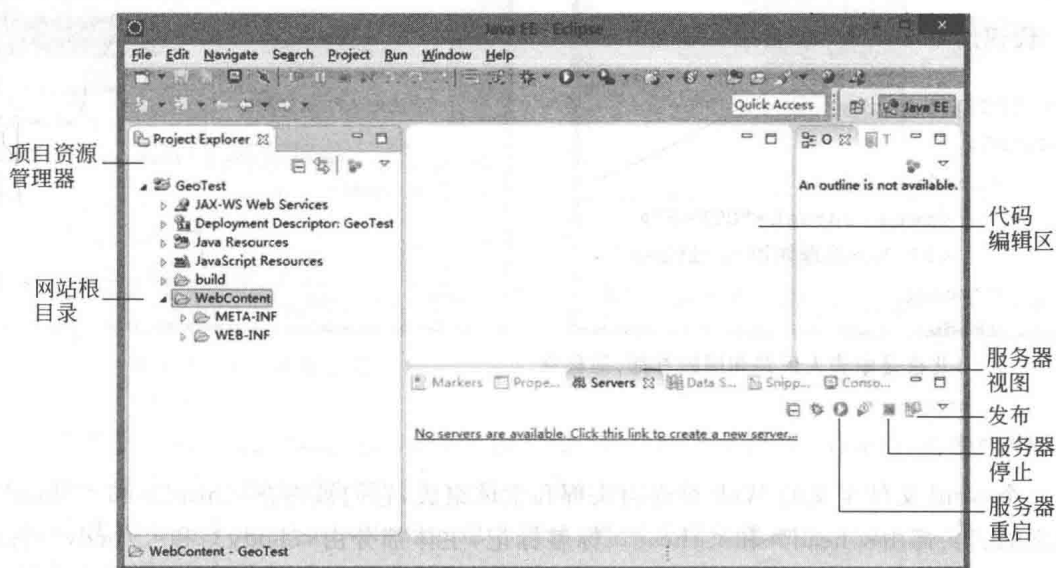


图 1.4 Web 项目目录结构

(3) 选中 Web 应用根目录 WebContent, 右击, 在弹出菜单中选中 New→HTML File 命令, 在文件名称处输入 index.html, 如图 1.5 所示。

(4) 单击 Next 按钮, 进入模板设置页, 如图 1.6 所示。选中使用 HTML 模板 (Use HTML Template) 后, 模板列表框列出了当前可用的 HTML 模板, 选择 html 5 模板, 单击 Finish 按钮, 利用模板生成 index.html。

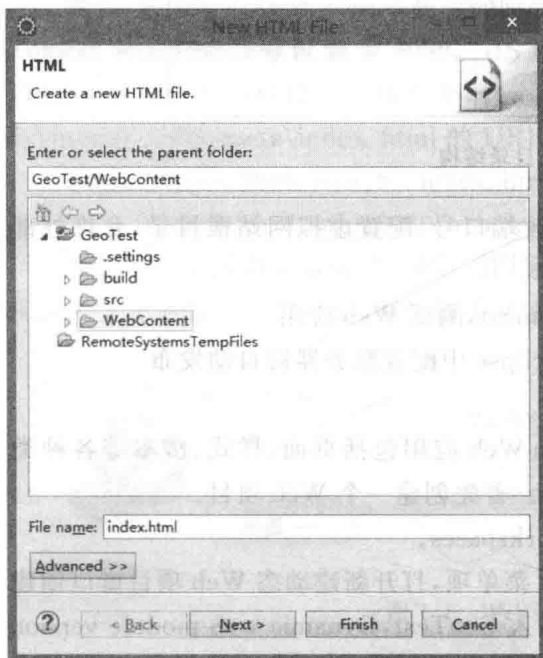


图 1.5 新建 HTML 文件对话框

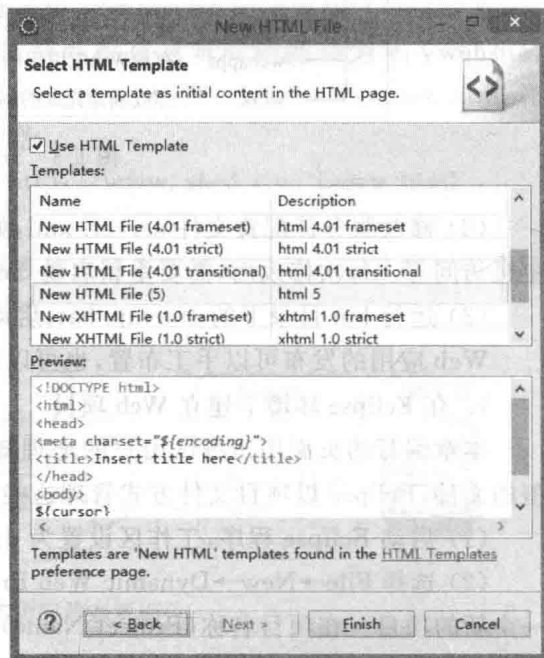


图 1.6 新建 HTML 文件模板选择对话框

(5) 打开 index.html, 将 title 标签内容修改为“地理知识”, 在 body 中添加文本, 结果如下:

#### 代码清单 1-1 静态页面 index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>地理知识</title>
  </head>
  <body>
    北京是中华人民共和国的首都,简称京。
  </body>
</html>
```

一个 html 文件定义的 Web 页面由头部和主体组成, 它们被夹在 <html> 和 </html> 标签之间。头部由 <head> 和 </head> 标签标记, 主体部分由 <body> 和 </body> 标签标记, 在浏览器窗口内看到的内容都在主体部分。标签可以大写也可以小写, 本书按 HTML 标准要求全部标签小写。

<!DOCTYPE> 指令告诉浏览器按照 HTML 的语法解释本页面,这是 HTML 5 的用法,在后面章节中使用的 HTML 4.01 版本中该指令格式会有所不同。

## 2. 在 Eclipse 中配置服务器

(1) 在服务器视图中(参见图 1.4,如果服务器视图未打开,选择 Window→Show View→Servers 菜单项打开服务器视图),单击链接 Click this link to create new Server,创建新的服务器。

(2) 在新服务器(New Server)窗口中,在服务器下拉列表选定 Apache→Tomcat v8.0 Server(或本机安装的其他版本服务器)。Server's host name 保持 localhost 不变,服务器名(Server name)保持不变。如图 1.7 所示。

localhost 为网络访问时的本机名,与实际主机名设置无关,此处也可以修改为 127.0.0.1 (两者都表示服务器运行在本机,127.0.0.1 为本机 IP 地址)。

(3) 单击 Next 按钮,进入新服务器运行环境窗口,在 Tomcat installation directory 处填写(或通过浏览方式填写)本机安装的 Tomcat 所在目录。单击 Next 按钮,进入资源发布选择窗口,如图 1.8 所示,双击当前项目名称 GeoTest,将其移入右侧下拉列表。单击 Finish 按钮,完成 Tomcat 服务器创建。服务器设置完成后,服务器视图中会出现以服务器名文本框中命名的服务器,此时,服务器处于停止状态。

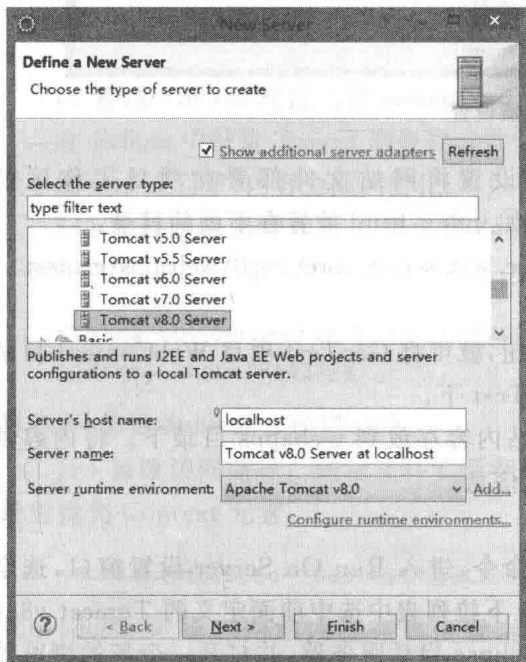


图 1.7 定义新服务器

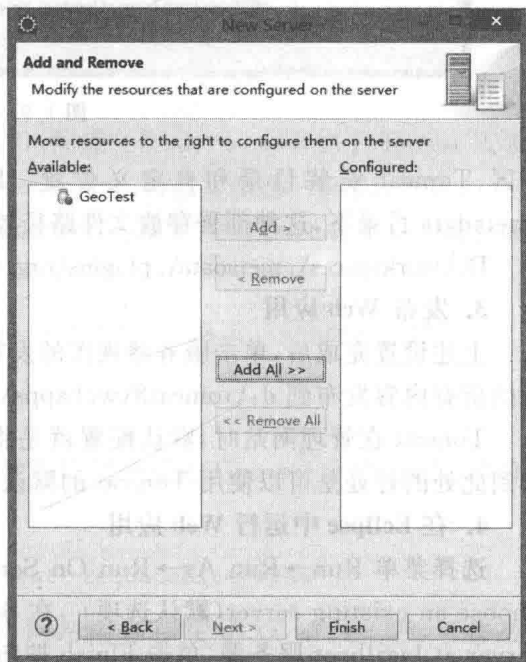


图 1.8 资源发布

(4) 在 Server 视图中选择服务器名(Tomcat v8.0 Server at localhost),右击,在菜单中选择 Open 命令,进入服务器配置信息设置页,如图 1.9 所示。

(5) 选择 Use Tomcat installation 单选按钮,将部署路径修改为 webapps,单击工具栏上的“保存”按钮保存修改后的服务器配置信息。

服务器位置(Server Locations)下的 3 个单选按钮分别对应将网站内容发布到项目工

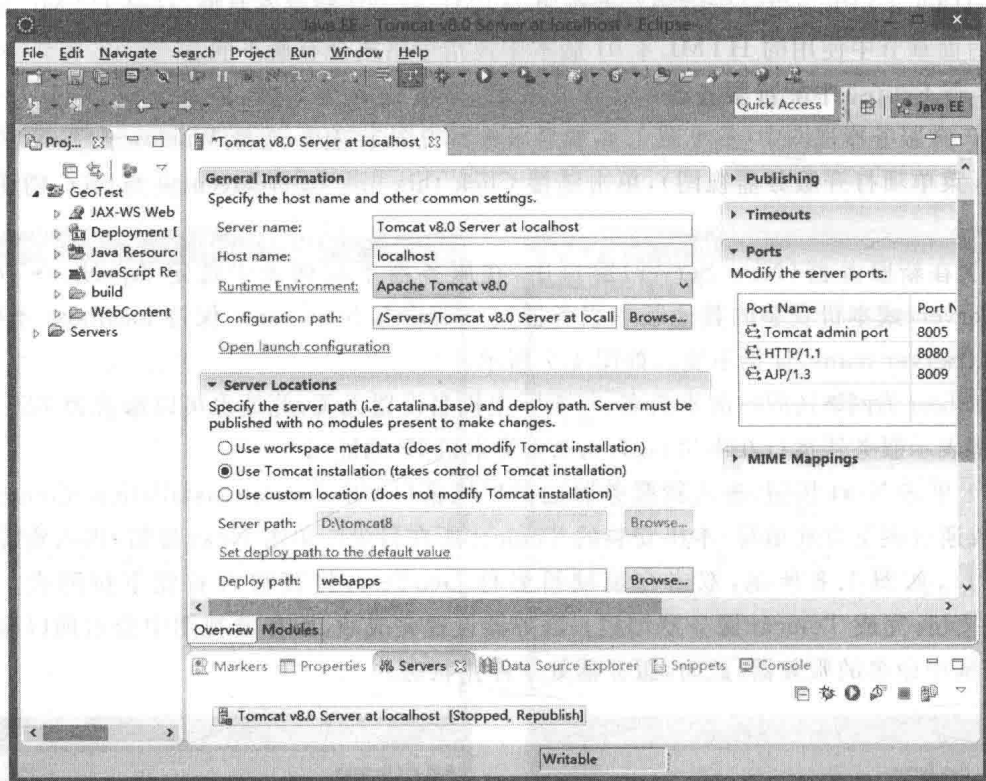


图 1.9 服务器设置

作区、Tomcat 安装目录和自定义位置。默认设置将网站文件部署在项目工作区的 .metadata 目录下,这种部署存放文件路径较长,如 index.html 放置在下面的目录:

D:\workspaces\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\webapps\geotest

### 3. 发布 Web 应用

上述设置完成后,单击服务器视图的发布按钮,就可将 GeoTest 项目 WebContent 目录下的所有内容发布到 d:\tomcat8\webapps\GeoTest 下。

Tomcat 在管理网站时,默认配置就是将网站内容存放到 webapps 目录下。将内容发布到此处的好处是可以使用 Tomcat 的默认配置信息。

### 4. 在 Eclipse 中运行 Web 应用

选择菜单 Run→Run As→Run On Server 命令,进入 Run On Server 设置窗口,选定 Choose an existing server(默认选项),在 Server 下拉列表中选中前面定义的 Tomcat v8.0 Server at localhost 服务器,单击 Finish 按钮。Eclipse 启动服务器,并打开一个新的浏览器窗口,显示 index.html 内容,如图 1.10 所示。

打开本机上的浏览器,在地址栏中输入地址 http://localhost:8080/GeoTest/index.html,可以看到显示结果与图 1.10 完全一致。在浏览器窗口任意位置右击,在弹出菜单中选择“查看源”命令,可以看到与 index.html 代码相同的内容。

### 5. 在 Tomcat 中运行 Web 应用

关闭 Eclipse,启动 Tomcat(运行 d:\tomcat8\bin\startup.bat)。打开浏览器,在地址栏中输入 http://localhost:8080/GeoTest/index.html,可以得到与图 1.10 相同的结果。



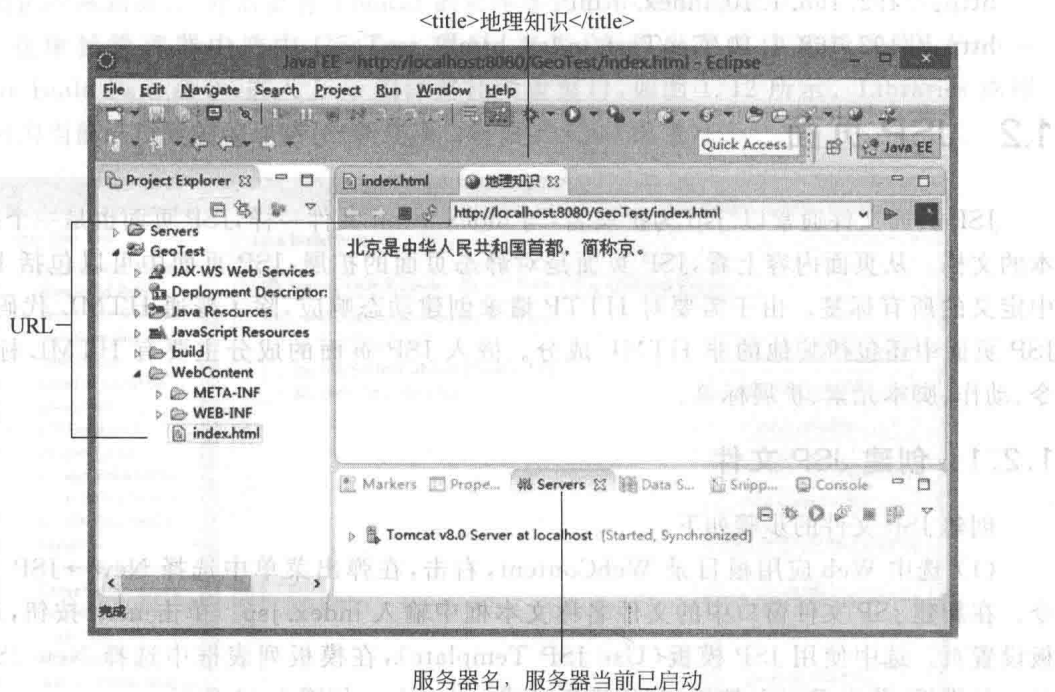


图 1.10 运行结果

## 6. 修改 Tomcat 配置文件 server.xml

在 Eclipse 中设置 Tomcat 服务器参数(图 1.8), 配置文件 server.xml(位于 Tomcat 安装目录下的 conf 子目录)内容会随同变化, 可以打开 server.xml, 根据需要进行进一步修改配置。

(1) 修改服务器端口号。服务器端口由 Connector 元素中的 port 指定, 找到下面的一行:

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1"
redirectPort="8443"/>
```

将 8080 修改为 80 或其他端口值。

(2) 修改访问路径。网站文件实际存储位置和访问路径由 Context 元素指定。在文件最后找到 Context 元素:

```
<Context docBase="GeoTest" path="/GeoTest" reloadable="true" />
```

其中, docBase 指定网站内容存放位置, 如果将网站内容存在 d:\GeoTest 下, 可修改为 docBase="d:\GeoTest"。path 指定相对路径, 如果不要该路径, 可修改为 path="/"。

(3) 允许其他主机访问。将

```
<Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true">
<Engine defaultHost="localhost" name="Catalina">
```

中的 localhost 替换成本机的 IP 地址, 例如, 本机地址如果为 192.168.1.10, 就将 localhost 用此地址替换。

上述配置修改完成后, 重启 Tomcat, 在本地局域网内其他的主机可以用下列地址访问: