

Java Web

应用开发

YINGYONG KAIFA

● 汤鸣红 主编

● 蒋卫祥 於志强 副主编



化学工业出版社

Java Web

应用开发

YINGYONG KAIFA

● 汤鸣红 主编
● 蒋卫祥 於志强 副主编



化学工业出版社

· 北京 ·

本书以 Servlet 2.4 和 JSP 2.0 规范为基础,详细介绍了应用 Java 技术实现 Web 应用的相关技术及编程方法,体现了应用 Java 技术开发 Web 应用的发展特性,涉及了当前应用广泛的开发规范,结构清晰,应用实例丰富,实现了理论学习和具体应用的充分结合。

主要包括:以 Servlet 为基础的 Web 应用的基础知识;以 JSP 2.0 规范为基础的 JSP 技术(JSP 基本语法、JSP 内置对象的使用、JavaBean 技术在 JSP 中的应用、标准标签库 JSTL 的使用),JSP 2.0 自定义标签的做法(Tag 接口、IteratingTag 接口、BodyTag 接口、SimpleTag 接口),JDBC 数据库访问技术(JDBC 处理数据库方法、使用数据源、使用连接池、实现分页显示),Web 应用分层与设计模式(JSP 设计模式、MVC 编程模式、持久层设计策略、应用 DAO 模式访问数据库。最后还介绍了一个 Java Web 综合实例——在线购书系统。

本书可作为计算机相关专业的教学用书,也可作为计算机工程技术人员的参考书。

图书在版编目(CIP)数据

Java Web 应用开发 / 汤鸣红主编. —北京:化学工业出版社, 2009.7

ISBN 978-7-122-05499-9

I. J… II. 汤… III. Java 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 068311 号

责任编辑:王听讲

文字编辑:吴开亮

责任校对:洪雅姝

装帧设计:刘丽华

出版发行:化学工业出版社(北京市东城区青年湖南街 13 号 邮政编码 100011)

印 装:三河市延风印装厂

787mm×1092mm 1/16 印张 18¼ 字数 502 千字 2009 年 7 月北京第 1 版第 1 次印刷

购书咨询:010-64518888(传真:010-64519686) 售后服务:010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价: 30.00 元

版权所有 违者必究

前 言

随着互联网的不断发展，目前的应用软件系统大多从原来的基于客户机/服务器模式，转向系统升级容易、维护方便、使用简单的基于浏览器/服务器的应用模式。

在这样的应用需求下，相应的 Web 开发技术也逐步成熟和完善。为了满足社会对软件人才的需求，让更多的人可以更快地学到实用的 Java Web 开发技术与方法，我们编写了《Java Web 应用开发》一书。

本书的主要特色如下。

1. 本书内容丰富，重点突出，覆盖当前主流 Java Web 各个关键技术，章节安排合理。
2. 展示 JSP 2.0、Servlet 2.4 的最新特性，并强调 Web 开发的综合实践。
3. 讲述了 JSP 语法和指令、JSP 内置对象、JavaBean、JSTL、标签库。
4. 涵盖了 Servlet 2.4 的 Servlet 基本结构、Servlet 会话跟踪、Servlet 相关接口和类、过滤器、数据库访问及 MVC 等方面的内容。
5. 重实践，给出很多实际案例，每个关键知识点都有一个案例。

本书以 Servlet 2.4 和 JSP 2.0 规范为基础，介绍了应用 Java 技术实现 Web 应用的相关技术及编程方法。本书内容分为三个部分。

基础部分：第 1 章～第 5 章主要讨论以 Servlet 为基础的 Web 应用的基础知识；第 6 章～第 9 章主要介绍 JSP 技术。

提高部分：第 10 章～第 12 章。第 10 章介绍了 JSP 2.0 自定义标签；第 11 章介绍了 JDBC 数据库访问技术；第 12 章介绍了 Web 应用分层与设计模式，介绍了 JSP 设计模式、MVC 编程模式、持久层设计策略、应用 DAO 模式访问数据库等。

综合应用部分：第 13 章详细分析并设计了一个在线购书系统。

本书将免费向读者提供 Java Web 应用开发相关的全部源代码，如有需要可与 wtjiang@sina.com 联系，或到化学工业出版社教育资源网站（<http://www.cipedu.com.cn>）免费下载使用。

参与此书编审的作者分工如下：

第 1、2、6、7 章由於志强编写；第 3、5、9、12、13 章由蒋卫祥编写；第 4、8、10、11 章由朱利华编写；汤鸣红组织和审阅了全书；蒋卫祥负责全书代码测试。

由于笔者水平有限，加之时间仓促，书中如有不妥之处，敬请读者不吝赐教。

编 者
2009 年 3 月

目 录

第 1 章	Web 应用体系架构	1
1.1	HTTP 请求响应模型	1
1.2	Web 应用程序发展过程	3
1.3	搭建 Java Web 开发环境	6
1.4	MyEclipse 开发工具	9
1.5	简单 Web 应用的开发样例	10
	练习思考题	11
第 2 章	Servlet 技术介绍	12
2.1	Servlet 介绍	12
2.2	基本 Servlet 结构	15
2.3	Servlet 的基本配置	20
	练习思考题	23
第 3 章	Servlet 相关接口和类	24
3.1	Servlet 包中的接口	24
3.2	Http 包中的接口和类	27
3.3	Servlet 中的表单处理	29
3.4	Servlet 请求转发	34
3.5	Servlet 异常	39
3.6	共享数据	40
	练习思考题	43
第 4 章	会话跟踪	45
4.1	会话跟踪的概念	45
4.2	Cookie	46
4.3	URL 重写	48
4.4	隐藏表单	50
4.5	HttpSession 会话跟踪	52
4.6	session 会话实例	56
	练习思考题	70
第 5 章	Servlet 过滤器和监听器	71
5.1	Servlet 过滤器	71

5.2	Servlet 事件监听	78
5.3	过滤器和监听器的应用场景	83
	练习思考题	97
第 6 章	JSP 基本语法	98
6.1	JSP 的构成	98
6.2	JSP 中的注释	99
6.3	指令元素	100
6.4	脚本元素	102
6.5	动作元素 (Action)	104
	练习思考题	108
第 7 章	JSP 内置对象	109
7.1	内置对象介绍	109
7.2	请求相应对象	109
7.3	会话对象 session	119
7.4	全局对象 application	121
7.5	其他对象	122
	练习思考题	126
第 8 章	在 JSP 中使用 JavaBean	127
8.1	JavaBean 的基本概念	127
8.2	在 JSP 中使用 JavaBean	129
8.3	JavaBean 应用实例	140
	练习思考题	148
第 9 章	标准标签库 JSTL	149
9.1	JSTL 介绍	149
9.2	一般用途的标签	151
9.3	条件标签	153
9.4	迭代标签	154
9.5	URL 相关的标签	156
9.6	函数标签	157
9.7	Database access 标签库	160
	练习思考题	161
第 10 章	JSP 自定义标签扩展	162
10.1	JSP 标签扩展	162
10.2	自定义标签示例	163
10.3	自定义标签接口	166
10.4	标签开发实例	178
	练习思考题	184

第 11 章 JDBC 数据库访问技术	185
11.1 JDBC 和驱动程序	185
11.2 JDBC 处理数据库的方法	186
11.3 使用 JDBC 操作数据库示例	189
11.4 连接不同的数据库	205
11.5 事务处理	206
11.6 使用数据源	207
11.7 使用连接池	210
11.8 使用 JTDS 访问数据库	220
11.9 实现分页显示	221
练习思考题	228
第 12 章 Web 应用分层与设计模式	230
12.1 Web 层设计目标	230
12.2 JSP 和 Servlet 结合的两模式	230
12.3 使用 MVC 编程模式	236
12.4 持久层设计策略	241
12.5 应用 DAO 模式访问数据库实例	243
练习思考题	248
第 13 章 在线购书系统	249
13.1 需求分析	249
13.2 系统设计	251
13.3 在线购书主页面设计	257
13.4 购物车管理模块	260
13.5 结账与订单管理模块	264
13.6 后台图书管理模块	269
13.7 代码发布与测试	275
练习思考题	276
附录	277
附录 1 Java Web 开发规范	277
附录 2 常见错误与解决方法	281
参考文献	285

第 1 章 Web 应用体系架构

Internet（因特网）是当今世界上最大的计算机信息网络，其实用性主要在于它的信息资源，Internet 在全球范围内提供了极为丰富的信息资源。从网络通信技术的观点来看，Internet 是一个以 TCP/IP 通信协议为基础，连接各个国家、企业和机构计算机网络的数据通信网；从信息资源的观点来看，Internet 是一个集各个领域、各个学科的各种信息资源为一体的、供网上用户共享的数据资源网。

WWW（World Wide Web）一般译为万维网，WWW 的创建是为了解决 Internet 上的信息传递问题，WWW 是由欧洲粒子物理实验室开发的，是采用超文本和多媒体技术，将不同文件通过关键字建立链接，提供一种交叉式查询方式（而不仅仅是传统的线性方式）。传输采用 HTTP（HyperText Transfer Protocol）超文本传输协议，而内容则采用超文本标记语言 HTML（Hyper Text Markup Language）。

Web 诞生于 20 世纪 90 年代早期，现在已经发展成为人们日常生活中重要的一部分了，它的出现立即改变了人们联系、交流、获取知识的方式。Web 应用已渗透到生产、生活的多个方面，对商业的发展影响很大，如新闻阅读、在线聊天、在线音乐、视频、电子商务、电子政务、企业信息管理、博客、论坛、搜索等。

随着网络和硬件设备的不断发展，通过各种手持设备和电脑，将可以随时随地接入到 Internet。Web 的应用能极大地提高生产、办事的效率。

1.1 HTTP 请求响应模型

HTTP 是一种让 Web 服务器与浏览器（客户端）通过 Internet 发送与接收数据的协议。HTTP 运用可靠的 TCP 连接，默认使用 80 端口。

超文本传输协议（HTTP）定义了简单事务处理程序，由以下 4 个步骤组成：

- ① 客户机与服务器建立连接；
- ② 客户机向服务器递交请求，在请求中指明所要求的特定文件；
- ③ 如果请求被接纳，那么服务器便发回一个应答，在应答中至少应当包括状态编号和该文件内容；
- ④ 客户机与服务器断开连接。

HTTP 是请求/响应协议，客户机通过浏览器发送请求，而服务器返回响应。下面是 HTTP 的工作步骤（访问 Java Web 网页）。

- ① 客户端（浏览器）发送 GET/POST 请求 连接到 Web Server。
- ② 服务器接收客户端发送的请求。
- ③ 采用 GET/POST 方法访问 HTML 网页、Servlet、JSP 等。
- ④ 网页、Servlet、JSP 等响应请求，运行解析后以 HTML 格式发送到服务器。
- ⑤ 服务器将响应请求通过网络发送到客户端。

⑥ 客户端得到响应内容后关闭连接，浏览器解析 HTML 内容显示。

1. HTTP 消息格式

格式包括三个部分：起始行、头信息、消息体。下面是一个简单的 HTTP 消息（前面粗体部分）：

```

POST /login HTTP/1.1           //起始行
Host: 192.168.1.1             /*
Content-Type: text/html       头信息
Content-Length:10            */

Hello.java                   //消息体

```

2. HTTP 请求消息

客户请求消息由以下三部分组成。

(1) 请求方法

请求方法（Request Method）确定数据如何发送给远程资源。有三种请求方法：GET、POST 和 HEAD。GET 方法是将数据作为 URL 的一部分发送；POST 方法是客户端与服务器建立连接后，数据在独自的数据流中发送；HEAD 请求不向服务器发送任何数据，只是描述关于远程资源的信息。

(2) 头信息

头信息包含多个字段信息（Head Field），客户端根据需要向服务器传递参数。常见的字段有 User-Agent、Content-Language、Content-Type、If-Modified-Since 和 Connection 等。

(3) 正文

正文（Body）是客户端发送给服务器的实际内容。如可以是 POST 方式提交时网页中相关组件的值。

3. HTTP 响应消息

当服务器端（Web Server）收到并处理了客户端请求后，就以 HTML 格式发送响应。服务器响应也由三部分组成。

(1) 状态行

服务器状态行是通过指定状态码通知客户端其请求的结果。主要分以下几类。

- ① 1XX 是提供信息。
- ② 2XX 是成功。
- ③ 3XX 是重定向。
- ④ 4XX 是客户端错误。
- ⑤ 5XX 是服务器端错误。

服务器端状态行包括在服务器上运行的协议版本号、状态码和状态码相关文字描述。如下面的例子。

- ① HTTP/1.1 200 OK
- ② HTTP/1.1 404 Not Found
- ③ HTTP/1.1 500 Internal Server Error

另外，有时错误状态码还会有子状态码，用来表示同类错误的不同状态，如“HTTP 500; 12 Internal Server Error-Application”。

(2) 头信息

与客户端类似，服务器可以通过头字段获取发送信息，可以返回多个头字段。

(3) 正文

服务器处理客户端请求后将处理结果发送给客户端，正文中为响应内容（HTML 文本）。

1.2 Web 应用程序发展过程

最初，所有 Web 页面都是静态的，用户请求一个资源，服务器再返回该资源。在浏览器发展的最初阶段，Web 页面的这种静态性不成问题。科学家只是使用因特网来交换研究论文，大学院校也只是通过因特网在线发布课程信息，企业界也还没有发现这个新“渠道”会提供什么商机。实际上，以前公司主页显示的信息通常很少，无非是一些联系信息或者只是一些文档。后来，Web 用户的期望越来越高，静态网页已经不能满足要求，用户希望得到更动态的网上体验，如动态的图片、视频、动态的数据，实现更多的交互等。

实现动态 Web 页面，除了静态的 HTML 技术，还要有能实现动态效果的客户端和服务端端的多种技术。

1.2.1 Web 客户端技术

1. Applet

1995 年 5 月，Sun 公司的 John Gage 和 Andreessen 宣布一种新的编程语言——Java 诞生。Netscape Navigator 为这种新语言提供了支持，最初是为了支持机顶盒。就像所有革命都机缘巧合一样，Java 和因特网的出现恰到好处，在适当的时间、适当的地点横空出世。Java 在 Web 上发布仅几个月，就已经有成千上万的人下载使用。由于 Netscape 的 Navigator 支持 Java，动态 Web 页面掀开了新的一页：Applet 时代到来了。

Applet 允许开发人员编写可嵌入在 Web 页面上的小应用程序。只要用户使用支持 Java 的浏览器，就可以在浏览器的 Java 虚拟机（Java Virtual Machine, JVM）中运行 Applet。尽管 Applet 可以做很多事情，但它也存在一些限制：通常不允许它读写文件系统，也不能加载本地库，而且可能无法启动客户端上的程序。除了这些限制外，Applet 还会在一个沙箱安全模型中运行，可以有助于防止用户运行恶意代码。

因为 Applet 在浏览器的虚拟机中运行，所以开发人员必须确保客户端安装了适当版本的 Java。尽管这个问题也可以解决，但它也妨碍了 Applet 技术的进一步推广。而且若 Applet 写得不好，很可能对客户主机造成影响，随着 Flash 的流行，Applet 应用也越来越少。

2. DHTML

动态 HTML（DHTML）支持 JavaScript 和 Java 等多项技术，但其最引人注目的特性是层叠样式表（Cascading Style Sheets, CSS）。层叠样式表可以帮助页面开发人员将显示元素从内容元素中分离出来。例如，与图书、杂志的页面布局相似的纯粹像素布局就需要级联样式表。层叠样式表还支持颜色、字体规范、显示图层和页边空白等页面元素特征。

动态 HTML（DHTML）文档对象模型（Document Object Model, DOM）使网页制作者可以直接地以可编程的方式访问 Web 文档上每个独立的部分，而不论被访问的是元素还是容器。这种访问方式包括了事件模型。事件模型令浏览器可对用户输入作出反应，通过执行脚本，无需从服务器下载一个新的页面就可以根据用户输入显示新的内容。动态 HTML 文档对象模型（DHTML DOM）以一种便捷的方式为广大普通网页制作者提供了丰富的网页交互性。

3. VBScript/JavaScript

JavaScript 是用于浏览器的第一种具有通用目的、动态的客户端脚本语言。Netscape 于 1995 年首先提出了 JavaScript，Java 的开发商 Sun 与他们在同一年发表了一项声明，声明中指出 Java

和 JavaScript 将互相补充，但它们是截然不同的技术，这样才纠正了很多人对这两项技术的错误理解。

JavaScript 为创建用户界面控件提供了一种脚本语言。事实上，JavaScript 在浏览器中插入了代码逻辑。它可以支持这样的效果：当光标在 Web 页的某个位置移动时验证用户输入或者变换图像。

Microsoft 也编写出了自己的 JavaScript 版本并将其称为 JScript。Microsoft 和 Netscape 都支持一种围绕 JavaScript 和 JScript 的核心特性并有（European Computer Manufacturers Association, ECMA）标准组织控制的脚本语言标准。ECMA 将其脚本语言命名为 ECMA Script。

Microsoft 还开发了 VBScript 作为其 Visual Basic 程序设计语言的解释子集。Microsoft 专门开发了 VBScript，用于在 Microsoft 的 Internet Explorer 上支持 Active X 控件，其他浏览器不支持 VBScript。

1.2.2 Web 服务器端技术

1. CGI

要让 Web 更为动态，最早的一个办法是使用公共网关接口（Common Gateway Interface, CGI）。与静态的 Web 获取不同，使用 CGI 可以创建程序，当用户发出请求时就会执行该程序。假设要在 Web 网站上显示销售的商品，可以利用 CGI 脚本来访问商品数据库，并显示结果。通过使用简单的 HTML 表单和 CGI 脚本，可以创建简单的网上商店，用户就可以通过浏览器来购买商品。编写 CGI 脚本可以用多种语言，从 Perl 到 Visual Basic 都可以，这使得掌握不同编程语言的人都能编写 CGI 脚本。

不过，要创建动态的 Web 页面，CGI 并不是最安全的方法，而且 CGI 程序基于进程，运行效率不高。尽管它存在一些缺陷，但到如今 CGI 仍在使用。

2. ASP

ASP（Active Server Page）是一种可以动态产生网页内容的技术。它可以在 HTML 程序代码中内嵌一些脚本语言（Scripting Language），如 JavaScript 和 VBScript。只要服务器端安装了适当的编译程序引擎，服务器便可以调用此编译程序来执行脚本语言，然后将结果传送到客户端的浏览器上。ASP 向用户提供制作网页的功能与 SSI（Server Side Include）或 CGI 应用程序非常相似。ASP 虽然功能非常优越，但只能在微软公司的服务器平台 IIS（Internet Information Server）中支持；ASP 是解释型执行语言，不进行编译，每次访问都从头到尾按代码解释执行，效率没有 JSP 编译型高；而且 ASP 不能跨平台。

另外微软的 Asp.Net 已经是编译型的，并且支持多种语言开发，运行效率非常高，而且支持可视化开发，但也有局限，只能应用于微软的 Windows 平台。

3. PHP

PHP（Personal Home Page 或 Hypertext Preprocessor）作为一个小开放源码，随着越来越多的人意识到它的实用性逐渐发展起来。Rasmus Lerdorf 在 1995 年发布了 PHP 的第一个版本，经过无数的改进和完善现在已经发展到版本 5.2.8。

PHP 是一种嵌入在 HTML 并由服务器解释的脚本语言。它能管理动态内容、支持数据库、处理会话跟踪，甚至构建整个电子商务站点。它支持许多流行的数据库，包括 MySQL、PostgreSQL、Oracle、Sybase、Informix 和 Microsoft SQL Server。

PHP 主要优点有：开发简便、易于上手、容错性好、多平台运用、函数较多。PHP 主要缺点有：运行时消耗资源较大、运行速度相对较慢、扩展性比较差、结构层次比较单一。

4. Servlet

尽管 Applet 是基于 Web 的，但客户端应用存在的许多问题在 Applet 上也有所体现。在大量

使用拨号连接的年代（就算是今天，拨号连接也很普遍），要下载一个复杂 Applet 的完整代码，要花很多时间，用户不能承受。开发人员还要考虑客户端上的 Java 版本，有些虚拟机还有更多的要求。理想情况下只需提供静态的 Web 页面就够了。当然，尽管静态页面是静态的，但是如果能在服务器上动态地生成内容，再把静态的内容返回，这就太好了。

Java 问世一年左右，Sun 引入了 Servlet，Java 代码不用再像 Applet 那样在客户端浏览器中运行了，它可以在控制的一个应用服务器上运行。这样，开发人员就能充分利用现有的业务应用，同时，如果需要升级为最新的 Java 版本，只需要考虑服务器就行了。Java 推崇“一次编写，到处运行”，这一点使得开发人员可以选择最先进的应用服务器和服务器环境，这也是这种新技术的另一个优点。另外 Servlet 还可以取代 CGI 脚本。

Servlet 向前迈出了很大一步。Servlet 提供了对整个 Java 应用编程接口（API）的完全访问，而且提供了一个完备的库可以处理 HTTP。不过，Servlet 不是十全十美的。使用 Servlet 设计界面可能比较困难，在典型的 Servlet 交互中，先要从用户那里得到一些信息，完成某种业务逻辑，然后使用一些“打印行”创建 HTML，为用户显示结果。

5. JSP

Servlet 不仅容易出错，很难生成可视化显示，而且还无法让开发者尽展其才。一般地，编写服务器端代码的人往往是软件开发人员，他们只是对算法和编译器很精通，但往往不能设计精美的网站外观。业务开发人员不仅要编写业务逻辑，还必须考虑怎么创建一致的设计。因此，很有必要将表示与业务逻辑分离。因此 JSP（Java Server Pages）出现了。JSP 的设计目的是为了将业务处理与页面外观相分离，这样 Web 设计人员能够专心设计页面外观，而软件开发人员可以专心开发业务逻辑。

6. EJB

1998 年，Sun 发布了 EJB 1.0 标准。EJB（Enterprise Java Bean）为企业级应用中必不可少的数据封装、事务处理、交易控制等功能提供了良好的技术基础。

J2EE 平台的三大核心技术为 Servlet、JSP 和 EJB。1999 年，Sun 正式发布了 J2EE 的第一个版本。紧接着，遵循 J2EE 标准，为企业级应用提供支撑平台的各类应用服务软件争先恐后地涌现了出来。IBM 的 WebSphere、BEA 的 WebLogic 都是这一领域里最为成功的商业软件平台。J2EE 体系及相关的软件产品已经成为了 Web 服务端开发的一个强有力的支撑环境。

1.2.3 基于 Java 的 JSP 技术特点

基于 Java 语言的 JSP 技术具有很多其他动态网页技术所没有的特点，具体表现在如下方面。

1. 简便性和有效性

JSP 动态网页的编写与一般的静态 HTML 的网页编写是十分相似的，只是在原来的 HTML 网页中加入一些 JSP 专有的标签，或是一些脚本程序（不是必需的）。这样，一个熟悉 HTML 网页编写的设计人员可以很容易进行 JSP 网页的开发。而且开发人员完全可以不用自己编写脚本程序，而只是通过 JSP 独有的标签使用别人已写好的部件来实现动态网页的编写。这样，一个不熟悉脚本语言的网页开发者，完全可以利用 JSP 做出漂亮的动态网页，而这在其他的动态网页开发中是不可实现的。

2. 程序的独立性

JSP 是 Java API 家族的一部分，它拥有一般的 Java 程序的跨平台的特性。换句话说，就是拥有程序的对平台的独立性，即“一次编写，到处运行”。

3. 程序的兼容性

JSP 中的动态内容可以以各种形式显示，所以它可以为各种客户提供服务。从使用 HTML/DHTML 的浏览器，到使用 WML 的各种手提无线设备（例如，移动电话和个人数字设备

PDA), 再到使用 XML 的 B2B 应用, 都可以使用 JSP 的动态页面。

4. 程序的可重用性

在 JSP 页面中可以不直接将脚本程序嵌入, 而只是将动态的交互部分作为一个部件加以引用。一旦这样的一个部件写好, 它可以为多个程序重复引用, 实现了程序的可重用性。现在, 大量的标准 JavaBeans 程序库就是一个很好的例证。

1.3 搭建 Java Web 开发环境

从最开始的 JSDK 到现在的 Tomcat、WebLogic、WebSphere 等, JSP 的运行环境也有了很大的变化, 出现了很多优秀的 JSP 容器, 如 Tomcat、JBoss、WebLogic、WebSphere 等, 这些都是应用服务器, 也可以称为中间件。下面简单介绍几种常用的 JSP 容器及其特点。

1. Tomcat

Tomcat 是 Apache Jakarta 软件组织的一个子项目, 它是一个 JSP/Servlet 的容器, 是在 Sun 公司的 JSDK (Java Server Web Development Kit) 基础上发展起来的一个 JSP 和 Servlet 规范的标准实现。使用 Tomcat 可以体验 JSP 和 Servlet 的最新规范。经过多年的发展, Tomcat 不仅是 JSP 和 Servlet 规范的标准实现, 而且具备了很多商业 Java Servlet 容器的特性, 使得它被广泛用于商业。

2. BEA WebLogic

BEA WebLogic Server 是 BEA Web 应用服务器的重要产品, 是用于集成、开发、部署和管理大型分布式 Web 应用、网络数据库应用的 Java 应用服务器。它将 Java 的动态功能和 Java Enterprise 标准的高效性和安全性引入大型 Web 应用的集成和开发之中, BEA WebLogic Server 是第一个提供 EJB 组件、Java 消息传递和事件服务、微软 COM 集成以及零管理客户机的 Web 应用服务器, 代表了新一代 Web 应用服务器的发展方向。2008 年 1 月, BEA 已经被 Oracle 公司收购。

3. IBM WebSphere

IBM WebSphere 是一种领先的因特网基础设施软件, 适用于跨多种平台创建、运行和集成各种业务的应用。它的优点是能将烦琐的 IT 流程进行整合, 并使其框架清晰, 使用简便, 能节省很多的人力和时间, 从而提高效率。

由于 Tomcat 是免费的, 也非常容易管理和使用, 所以本书以 Tomcat 作为 JSP/Servlet 的容器。

1.3.1 安装和配置 JDK

开发和运行 Java Web 应用离不开 JDK 和 JVM, 所以必须首先安装 JDK, 这里安装的是 JDK1.6, 设置好相应环境变量后, 再安装配置 Tomcat (带 JSP/Servlet 容器的开源 Java Web Server-Apache)。

JDK 是 Java 语言的编译环境, 可以从 Sun 公司的网站上免费下载。其下载地址是 “<http://java.sun.com/javase/downloads/index.jsp>”。

把 JDK 下载后执行安装程序, 默认安装目录是 “C:\Program Files\Java\jdk1.6.0_05”, 把这个目录设定为 JAVA_HOME。

安装完成后, 需要做些配置工作, JDK 才能开始正常工作, 可以按照下面介绍的步骤配置 JDK。

① 在桌面上右击 “我的电脑”, 选择 “属性” 命令, 在出现的对话框中选择 “高级” 选项卡, 然后单击 “环境变量” 按钮, 出现如图 1-1 所示的窗口。

② 检查在系统变量部分是否有 Path 变量, 如果没有, 则新建一个名为 “Path” 的变量, 并添加路径 “C:\Program Files\Java\jdk1.6.0_05\bin;”, 如果有, 则在原有路径的末尾添加 “C:\Program Files\Java\jdk1.6.0_05\bin;”, 效果如图 1-2 所示。

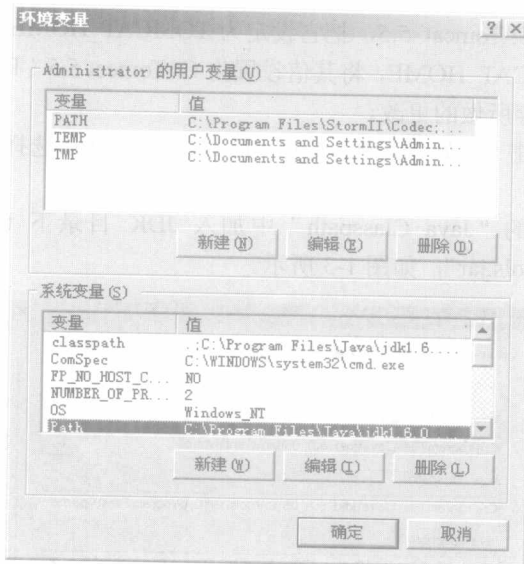


图 1-1 Java 环境变量设置

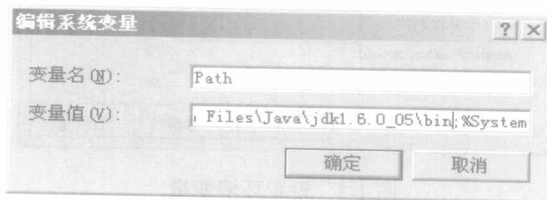


图 1-2 设置 Path 变量

③ 单击“确定”按钮，保存所做的修改。

④ 新建一个系统变量，名为 JAVA_HOME，值为“C:\Program Files\Java\jdk1.6.0_05”。

⑤ 新建一个系统变量 classpath，值为“.;C:\Program Files\Java\jdk1.6.0_05\lib\tools.jar; \ProgramFiles\Java\jdk1.6.0_05\jre\lib\rt.jar”。

注意 classpath 变量的值必须带“.”，表示当前目录，这样执行当前目录下的“.class”文件才不会出错。

下面编写一个简单的 Java 程序测试 JDK 的安装。HelloWorld.java 的代码如下。

【例 1-1】 一个简单的例子。

```
public class HelloWorld{
    public static void main(String args[]){
        System.out.println("Java 桌面版本的应用程序");
    }
}
```

如果编译运行输出了字符串，就表明 JDK 安装配置成功了，如果提示错误或者类找不到，请检查配置的过程是否与上面介绍的一样。

1.3.2 安装和配置 Tomcat

只有在确保 JDK 安装正确的情况下才可以安装 Tomcat。Tomcat 提供了可执行程序的安装程序，可以从其官方网站免费下载。下载地址为“<http://tomcat.apache.org/index.html>”。

目前 Tomcat 最新的版本是 6.0，本书使用的版本是 Tomcat 5.5，功能基本一致。执行 Tomcat 5.5 安装程序，使用默认设置并同意就可以了。

假定安装的主目录是 C:\Tomcat 5.5, 把它设定为 TOMCAT_HOME, 按照前面介绍的步骤添加一个新的环境变量 TOMCAT_HOME, 将其值设置为 C:\Tomcat 5.5 (Tomcat 安装的主目录), 然后单击“确定”按钮, 保存所做的更改。

在 Windows 系统中单击“开始”菜单, 选择“所有程序”, 然后选择 Apache Tomcat 5.5 命令, 并选择 Configure Tomcat。

在“Java”属性页中的“Java Classpath”中加入 JDK 目录下 tools.jar, 如“C:\Program Files\Java\jdk1.6.0_05\lib\tools.jar”, 如图 1-3 所示。

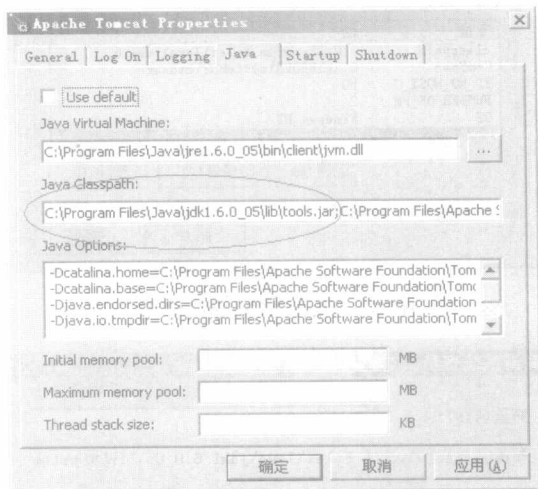


图 1-3 设置环境变量

在“General”属性页单击 start 按钮, 就可以启动 Tomcat 了。Tomcat 启动完成后, 在浏览器地址栏中输入地址“http://localhost:8080/”, 可以看到如图 1-4 所示的 Tomcat 的欢迎页面, 则表明 Tomcat 安装成功了。

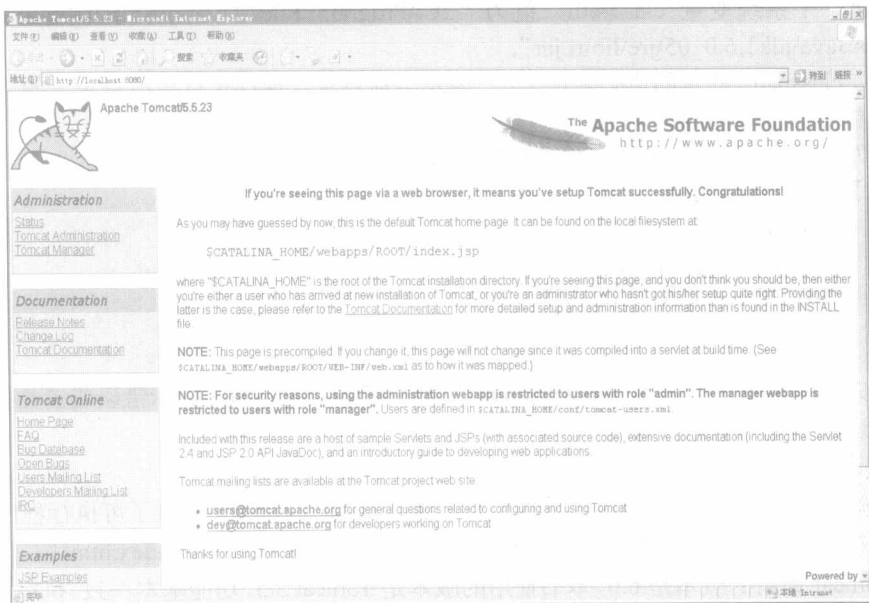


图 1-4 Tomcat 的欢迎页面

1.4 MyEclipse 开发工具

1.4.1 MyEclipse 工具简介

Eclipse 是一个开放源代码的软件开发项目,专注于为高度集成的企业开发提供一个全功能的具有商业品质的工业平台。MyEclipse 是 Eclipse 的插件,也是一款功能强大的 J2EE 集成开发环境,支持代码编写、配置、测试以及除错。

Genuitec 发布了 MyEclipse Enterprise Workbench 5.0,更智能、更快捷、更简单、更便宜的 J2EE 工具的新版本。它的价格对于个人和企业开发人员来说都是非常有吸引力的。它增加 UML 双向建模工具、WYSIWYG 的 JSP/Strutsdesigner、可视化的 Hibernate/ORM 工具、Spring 和 Web services 支持以及新的 Oracle 数据库开发。MyEclipse 5.0 继续为用户提供全面的产品,目前最新的版本是为 6.0。

1.4.2 MyEclipse 开发 Web 应用程序步骤

1. 准备

从下列网站上分别下载软件,并安装。本书以 MyEclipse5.5 版本为例说明 MyEclipse 使用过程。

Eclipse	http://www.eclipse.org/downloads
MyEclipse5.5	http://www.myeclipseide.com
Tomcat5.5	http://jakarta.apache.org/tomcat/index.html

2. 配置

安装完 MyEclipse 后在 Eclipse 中会出现一个 MyEclipse 的菜单,如图 1-5 所示。

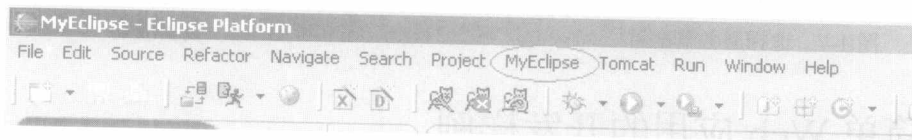


图 1-5 MyEclipse 的菜单

设置 Web 服务器: (Tomcat 5)。打开“Windows”→“Preferences”窗口,在“MyEclipse”→“Application Server”下选择“Tomcat 5”,在配置中允许使用 Tomcat (选择 Enable),并设置 Tomcat 的安装目录 (Tomcat Home Directory),最后点“Apply”使配置生效,然后选择“Tomcat 5”→“JDK”,增加一个 Tomcat JRE 配置,并保存配置。

3. 建立工程

在 New Project Wizard 中选择 J2EE 下的 Web Project,然后输入“Project Name”后点“Finish”按钮。新增一个 JSP 文件测试一下,用 MyEclipse 的 JSP Wizard 可以很快生成一个,可以自动配置 web.xml。

然后设置该 Web Project,具体设置如图 1-6~图 1-8 所示。

最后发布,在工具栏中选择 Deploy J2EE Project to Server,然后选择将要发布到的服务器 (这里选 To .lcat 5),如果这里没有要发布的服务器,请参见之前的配置中设置服务器,确认后出现发布成功的信息。最后启动服务器 (Tomcat),用浏览器运行测试 JSP。

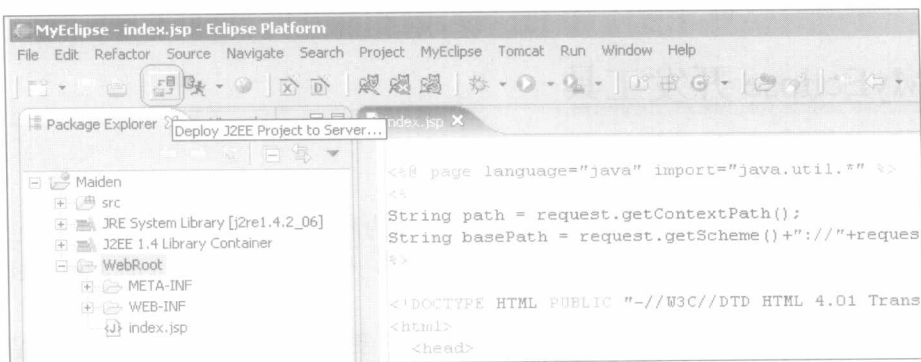


图 1-6 MyEclipse 的设置菜单

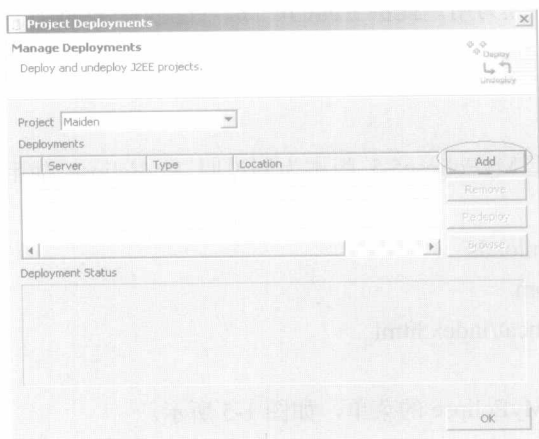


图 1-7 选择服务器配置

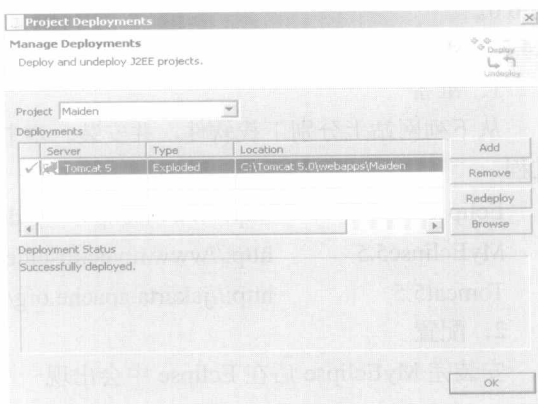


图 1-8 设置成功

1.5 简单 Web 应用的开发样例

1.5.1 Java Web 应用的目录结构

Tomcat 下的 Web 应用有固定的结构。首先在“webapps”目录下创建一个目录“helloapp”。则这个 Web 应用的目录结构如下。

- ① helloapp: Web 应用的根目录, 所有的 JSP 和 HTML 文件都存放在该目录下。
- ② helloapp/WEB-INF: 存放 Web 应用的描述文件 web.xml。
- ③ helloapp/WEB-INF/classes: 存放各种 class 文件, Servlet 类文件也存放于该目录下。
- ④ helloapp/WEB-INF/lib: 存放 Web 应用所需的各种 JAR 文件。如可以存放 JDBC 驱动程序的 JAR 文件。

1.5.2 简单 HTML 和 JSP 测试

下面编写一个简单的 JSP 程序, 内容跟普通 HTML 文件是一样的, 唯一的区别就是它的文件名后缀是 JSP, 而不是 HTML。

【例 1-2】简单的 JSP 程序。

```
<html>
<head><title>Welcome</title></head>
```