

计算机系列教材

Java Web项目开发 案例教程

王娜 编著

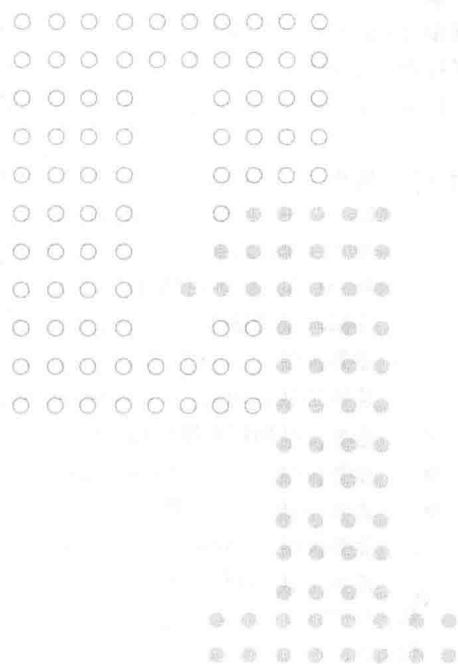


清华大学出版社

计算机系列教材

王娜 编著

Java Web项目开发 案例教程



清华大学出版社
北京

内 容 简 介

本书全面而又详细地介绍了 Java Web 应用项目开发需要的各种知识与技能,主要包括开发环境的安装与配置、Servlet、JSP、过滤器、监听器、JavaBean、JDBC 等应用。本书涵盖了 5 个小项目:登录验证器、网络购物车、编码过滤器、留言板、用户信息管理小系统,是一本以“项目驱动、案例教学、理论与实践相结合”教学方法为主的一体化教材。

本书结构严谨,层次分明,不仅适合作为计算机及相关专业 Java Web 开发的教材,也可作为专业技术人员的参考书。

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

Java Web 项目开发案例教程/王娜编著. —北京:清华大学出版社,2014

计算机系列教材

ISBN 978-7-302-37364-3

I. ①J… II. ①王… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 163170 号

责任编辑:张 玥

封面设计:常雪影

责任校对:时翠兰

责任印制:沈 露

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

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

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

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:12.75

字 数:292千字

版 次:2014年12月第1版

印 次:2014年12月第1次印刷

印 数:1~2000

定 价:27.00元

产品编号:060304-01

随着信息社会的发展,传统的教育模式已难以满足社会需要。一方面,大量毕业生无法找到满意的工作,另一方面,用人单位却在感叹无法招到符合职位要求的员工。因此,积极推进教学改革,根据市场需求调整课程和教学,已成为多数院校专业建设和课程改革的共识。

本书是根据教学改革要求,采用项目导向、任务驱动等新的教学思路编写的。它改变了以往遵循章节设计的教学思路,用项目和细化的小任务贯穿整个教材的知识体系,旨在训练学生的岗位工作技能,培养其解决实际问题的综合能力。全书由5个项目组成,每个项目又涵盖了若干实施该项目的工作任务,细化了项目学习目标,条理清晰,层次分明,方便读者一步步实现学习目标。

本书在介绍知识点的同时还列举了几十个实例。它们都源于教学、科研和企业、行业的最新典型项目,内容全面,贴近实际,具有可读性、趣味性和广泛性。

本书由王娜编写。编者在编写过程中参考了大量技术资料,吸取了许多同仁、企业专家的宝贵经验,建立了以“工作项目为导向,用工作任务为驱动,以行动体系为框架,以典型案例情境为引导”的教材体系。

由于编者水平有限,书中难免存在疏漏或不妥之处,敬请读者提出宝贵意见和建议,发送邮件到 wnbird@hotmail.com。

编者

2014年5月

- 项目 1 登录验证器 /1
 - 1.1 项目描述 /1
 - 1.2 学习目标 /1
 - 1.3 项目实施 /1
 - 任务 1.3.1 Web 应用入门 /1
 - 任务 1.3.2 Java Web 环境搭建 /6
 - 任务 1.3.3 Servlet 基础知识 /15
 - 任务 1.3.4 Servlet 的生命周期 /20
 - 任务 1.3.5 Servlet API /27
 - 任务 1.3.6 登录验证器的编写 /36
 - 1.4 学习总结 /43
 - 1.5 课后习题 /43

- 项目 2 网络购物车 /45
 - 2.1 项目描述 /45
 - 2.2 学习目标 /45
 - 2.3 项目实施 /45
 - 任务 2.3.1 服务器应用对象 /45
 - 任务 2.3.2 页面跳转与包含 /54
 - 任务 2.3.3 Servlet 间传递参数的方法 /60
 - 任务 2.3.4 购物车设计 /70
 - 2.4 学习总结 /80
 - 2.5 课后习题 /80

- 项目 3 编码过滤器 /81
 - 3.1 项目描述 /81
 - 3.2 学习目标 /81
 - 3.3 项目实施 /81
 - 任务 3.3.1 与 ServletContext 对象相关的
侦听器 and 事件 /81
 - 任务 3.3.2 与 HttpSession 对象相关的
侦听器 and 事件 /87

任务 3.3.3	与 ServletRequest 对象相关的 侦听器	/90
任务 3.3.4	过滤器基础	/91
任务 3.3.5	创建 Servlet 过滤器	/93
任务 3.3.6	编码过滤器	/98
3.4	学习总结	/105
3.5	课后习题	/105
项目 4	留言板	/106
4.1	项目描述	/106
4.2	学习目标	/106
4.3	项目实施	/106
任务 4.3.1	JSP 简介	/106
任务 4.3.2	JSP 页面基本结构	/113
任务 4.3.3	JSP 隐含对象	/129
任务 4.3.4	JSP 范围	/150
任务 4.3.5	留言板	/150
4.4	学习总结	/156
4.5	课后习题	/157
项目 5	用户信息管理小系统	/158
5.1	项目描述	/158
5.2	学习目标	/158
5.3	项目实施	/158
任务 5.3.1	JavaBean 简介	/158
任务 5.3.2	JSP 调用 JavaBean	/160
任务 5.3.3	JSP 与 Servlet 间传递参数的 三个作用范围	/166
任务 5.3.4	数据库访问	/166
任务 5.3.5	用户信息管理小系统	/177
5.4	学习总结	/194
5.5	课后习题	/194
	参考文献	/195

项目 1 登录验证器

1.1 项目描述

登录验证器是实现用户登录验证的,主要需求如下:用户在登录页面上输入用户名和密码,服务器端接收到客户端提交的请求,到数据库检验用户是否合法。如果是合法用户,转到欢迎页面,否则返回登录失败页面。同时,新用户还可以通过注册页面注册。

本项目完成的功能如下。

- (1) 可以注册新用户,注册的信息存入数据库中,注册的用户名必须是没注册过的。
- (2) 可以从数据库中查询用户名和密码,实现登录验证。

1.2 学习目标

学习目标:

- (1) 了解 Web 应用程序的基本知识。
- (2) 正确安装、配置 Tomcat,掌握 Web 环境的搭建。
- (3) 了解 Servlet 的基本知识。
- (4) 会使用 Servlet 的各个对象。
- (5) 能够编写 Servlet 的应用程序。

本项目通过完成登录验证器的编写,介绍开发 Web 应用程序基本概念、Web 环境搭建、Servlet 组件等相关知识。

1.3 项目实施

任务 1.3.1 Web 应用入门

1. 什么是 Web

Web 全称 World Wide Web,简称 WWW,译名万维网或全球信息网。Web 提供一个图形化的界面,用以浏览网上资源。它是一个在 Internet 上运行的全球性、分布式信息发布系统,通过 Internet 提供基于超媒体的数据信息服务。它把各种类型的信息(文本、图像、声音和影视)有机地集成起来,供用户使用。

2. URL 简介

统一资源定位符(Uniform Resource Locator,URL)是 Web 页的地址,该地址会在浏

览器顶部的 Location 或 URL 框内显示出来。鼠标指针移至某个超链接上方时,URL 也会在屏幕的底部显示出来。

URL 由两个主要部分构成: 协议(Protocol)和目的地(Destination)。

“协议”告诉人们面对的是何种类型的 Internet 资源。Web 中最常见的协议是 http, 它表示从 Web 中取回的是 HTML 文档。其他协议还有 Gopher、FTP 和 Telnet 等。

“目的地”可以是某个文件名、目录名或某台计算机的名称。例如 `http://www.bsa.edu.cn/index.html`, 这样的一个 URL 能让浏览器知道 HTML 文档的正确位置以及文件名是什么。如果 URL 是 `ftp://ftp.netabc.com/`, 浏览器就知道自己该登录进入一个 FTP 站点, 它位于名为 `netabc.com` 的一台网络计算机内。

Web 中有加亮、下划线或用不同颜色显示的某些文字——那些文字也许就是某个超链接(Hyperlink)。如果鼠标指针移至它们上方时改变了形状, 则确信它们就是超链接。这就意味着文字是“可单击”的。单击这种文字, 便会直接进入一个新网页。

那么计算机如何知道下面该传送哪个网页呢? 这就需要使用 URL 来实现。

URL 用来定义 Web 网页地址的格式如下。

URL 格式: 协议名://主机名[: 端口号]/[路径名/.../文件名]

例: ↑ ↑ ↑ ↑
 http ://www.asd.com /edu/ index.htm

注: 主机名 = 域名或 IP 地址。

3. Web 发展历程

Web 技术起源于 20 世纪 80 年代。1991 年, CERN(European Organization for Nuclear Research)正式发布了 Web 技术标准。目前, 与 Web 相关的各种技术标准都由著名的 W3C 组织(World Wide Web Consortium)管理和维护。

Web 是一种典型的分布式应用架构。Web 应用中的每一次信息交换都涉及客户端和服务端两个层面。因此, Web 开发技术大体也可以分为客户端技术和服务端技术两大类。Web 客户端的主要任务是展现信息内容, 最早、最常用的客户端信息显示语言为 HTML, 它是目前使用最广泛且被大多数浏览器支持的语言。和 Web 客户端相关的技术还有 JavaScript、VBScript、CSS 等。

Web 服务器端主要是响应客户的请求, 最早的 Web 服务器仅是简单响应浏览器发来的 HTML 请求, 并将存储在服务器上的 HTML 文件返回给客户端浏览器。这种静态的服务功能非常有限, 随着 Web 技术的发展, 逐渐出现了各种可以动态响应客户请求的技术, 从最早的 CGI 技术到目前的 PHP、ASP、JSP 等。通过这些动态的 Web 服务器端技术, 人们可以享受到信息检索、信息交换、信息处理等更为便捷的动态信息服务。

4. C/S 结构与 B/S 结构

Web 应用程序基于 B/S 结构, 和一般的 C/S 的应用程序不同。所谓的 C/S 结构, 即 Client/Server(客户机/服务器)结构, 也称桌面应用程序。通过将任务合理分配到 Client

端和 Server 端,降低了系统的通信开销,可以充分利用两端硬件环境的优势。典型的桌面应用程序有字处理程序、记事本、媒体播放器等。

最简单的 C/S 体系结构的数据库应用由两部分组成,即客户应用程序和数据库服务器程序。该结构的客户端主要处理数据,而服务器端主要存储数据,通过把应用数据的处理和数据存储合理地分配在客户机和服务器两端,可以有效降低网络通信量和服务器的运算量。但是,基于 C/S 模式的应用需要专门的客户端安装程序,分布功能弱,针对点多面广且不具备网络条件的用户群体,不能够实现快速部署安装和配置。所以这种结构的软件适于在用户数目不多的局域网内使用。另外,由于它的客户端比较庞大,常称为胖客户端,在较大范围内部署和应用维护成本较高。

C/S 结构的软件还存在一些问题,如伸缩性差、性能较差、重用性差、移植性差等。

C/S 结构的系统结构如图 1.1 所示。



图 1.1 C/S 系统结构图

B/S 软件体系结构,即 Browser/Server (浏览器/服务器)结构,是随着 Internet 技术的兴起,对 C/S 体系结构的一种变化或改进的结构。在 B/S 体系结构下,用户界面完全通过 WWW 浏览器实现,一部分事务逻辑在前端实现,但是主要事务逻辑在服务器端实现。数据库不是直接向每个客户提供服务,而是与 Web 服务器交互,实现了对客户请求服务的动态性、实时性和交互性。

B/S 结构的系统结构如图 1.2 所示。

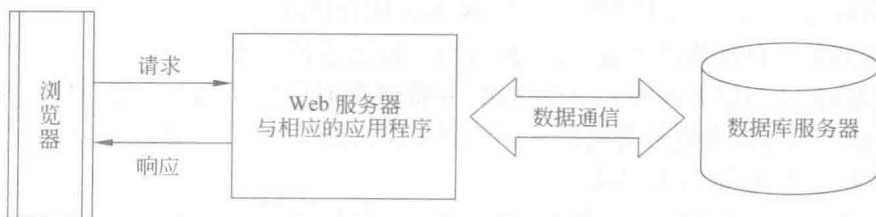


图 1.2 B/S 系统结构图

C/S 和 B/S 的差异主要是在支撑环境、安全控制、程序架构、软件重用、系统维护、用户接口、信息流等方面。

5. 静态网页与动态网页

静态网页的内容是固定的,不会根据浏览者的不同需求而改变,一般使用 HTML 语言编写。早期网页一般都是静态网页,通常以 .htm、.html、.shtml、.xml 等为后缀,如 <http://www.163.com/index.htm>。

静态网页具有如下特点。

(1) 每个网页都有一个固定的 URL,且以 .htm,.html,.shtml 等常见形式为后缀。

(2) 网页内容一发布到网站服务器上,无论是否有用户访问,每个静态网页的内容都保存在网站服务器上。

(3) 内容相对稳定,因此容易被搜索引擎检索。

(4) 没有数据库的支持,网站制作和维护的工作量较大。

(5) 交互性差,在功能方面有较大限制。

判断一个网页是否是静态网页,可以先看后缀名,再看是否能与服务器发生交互行为。静态网页有很大的局限性,仅由 HTML 页面构成的 Web 应用程序的内容是静止的,不会对用户的动作做出动态响应。

动态网页是指在接到用户访问要求后动态生成的页面,页面内容会随着访问时间和访问者发生变化。动态网页通常以 .asp,.jsp,.php,.perl,.cgi 等为后缀。

动态网页具有如下特点。

(1) 以数据库技术为基础,极大地降低了网站维护的工作量。

(2) 采用动态网页技术的网站可以实现更多的功能,如用户注册、用户登录、在线调查、用户管理、订单管理等。

(3) 动态网页实际上并不是独立存在于服务器上的网页文件,只有当用户请求时,服务器才给出一个完整的网页。

(4) 动态网页中的“”对搜索引擎检索存在一定问题。搜索引擎一般不可能从一个网站的数据库中访问全部网页,或者出于技术方面的考虑,搜索蜘蛛不去抓取网址中“”后面的内容。因此,采用动态网页的网站推广时,需要做一定的技术处理,才能适应搜索引擎的要求。

6. HTTP 请求

在前面的 B/S 结构中,用户的请求和 Web 应用程序的响应需要通过 Internet 从一台计算机发送到另一台计算机或服务器,使用的是超文本传输协议(HTTP)。HTTP 客户端(如浏览器)需要与服务器建立一个连接,并将请求消息发送到 HTTP 服务器,以请求相应的资源,然后服务器返回带有请求资源的相应消息。HTTP 协议的主要特点如下。

(1) 支持客户机/服务器模式。

(2) 简单快速:客户向服务器请求服务时,只需传送请求方法和路径。请求方法常用的有 GET、POST、HEAD 等。由于 HTTP 协议简单,因此 HTTP 服务器的程序规模小,通信速度很快。

(3) 灵活:HTTP 允许传输任意类型的数据对象。

(4) 无连接:限制每次连接只处理一个请求。服务器处理完客户的请求,并收到客户的应答后,即断开连接。采用这种方式可以节省传输时间。

(5) 无状态:HTTP 协议是无状态协议。无状态是指协议对于事务处理没有记忆能力,如果后续处理需要前面的信息,则必须重传,这可能导致每次连接传送的数据量增大。另一方面,服务器不需要先前信息时,它的应答就较快。

HTTP 请求消息使用 GET 或 POST 方法,以便在 Web 上传输请求。

检索信息时使用 GET 方法,如检索文档、图表或数据库查询结果。要检索的信息作为字符参数传递,称为查询字符串。因此传递的数据对客户端是可见的。根据服务器的不同,查询字符串的长度限制在 240~255 个字符。例如,要使用 GET 方法,在网站 www.163.com 中查询 name 为 a 的用户信息,那么查询字符串的表示如下。

```
www.163.com/user?name=a
```

HTTP 定义的另一请求方法是 POST 方法。使用 POST 发送的数据对客户端是不可见的,且对发送数据的量没有限制。POST 方法多用来传输敏感数据,如信用卡号或用户的密码等。

7. Web 应用程序体系结构

大多数应用程序由以下 3 个组件组成。

- (1) 表示逻辑:由用户界面和用于生成界面的代码组成。
- (2) 业务逻辑:包含系统的业务和功能代码。
- (3) 数据存取逻辑:负责完成存取数据库的数据。

这 3 个组件中使用的单词“逻辑”通常被替换为“层”。因此,这 3 个组件被称为表示层、业务层和数据层或数据存取层。应用程序的体系结构定义如何将这些组件组合在一起,并交互完成软件的功能。以下是 3 种应用程序体系结构。

1) 一层体系结构

在这种体系结构中,所有与表示逻辑、业务逻辑和数据存取逻辑相关的代码都耦合在一起。如图 1.3 所示。

2) 二层体系结构

在这种体系结构中,数据存取逻辑的代码与业务逻辑和表示逻辑分开,如图 1.4 所示。而且任何与数据存取层的交互都通过业务层完成,但表示逻辑和业务逻辑的代码仍然耦合在一起。

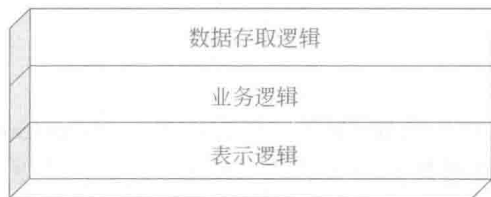


图 1.3 一层体系结构

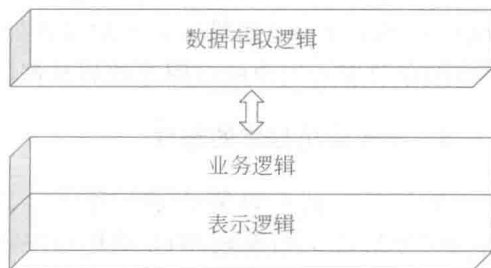


图 1.4 二层体系结构

3) 三层体系结构

在这种体系结构中,与 3 个组件相关的代码相互之间保持独立。但是,现在是业务层充当数据存取层和表示层之间的接口,通常表示层不能直接与数据存取层进行通信。三层体系结构如图 1.5 所示。

设计良好的 Web 应用程序通常基于三层体系结构,它的优点如下。

(1) 降低了各组件之间的耦合性,即一个组件的更改不会影响其他两个组件。例如,如果用户需要更换数据库,那么只有数据存取逻辑组件需要修改代码。同样,如果更改了用户界面设计,那么只有表示逻辑组件需要修改。

(2) 由于表示逻辑和数据存取逻辑相对独立,因而可以方便地扩充表示逻辑,使系统具有良好的可扩展性。

(3) 代码重复最少,因为在 3 个组件之间尽可能共享代码。

(4) 良好的分工与协作,这将使不同的小组能够独立开发应用程序的不同部分,并充分发挥各自的长处和优势。

图 1.6 展示了应用于 Web 应用程序的三层体系结构。此处,表示层(通常为 HTML 或 JSP 页面)由在客户端系统显示用户接口的代码组成。

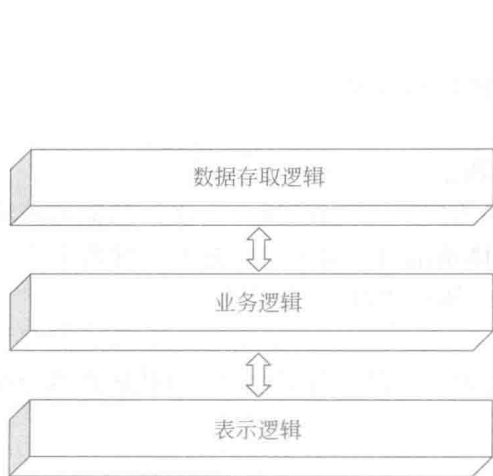


图 1.5 三层体系结构

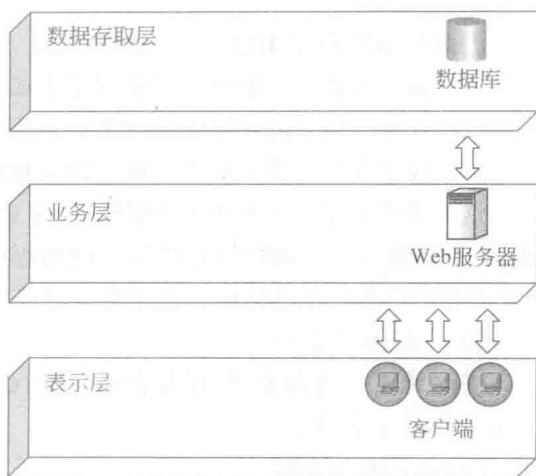


图 1.6 基于三层体系结构的典型 Web 应用程序

例如,用户界面可以是包含用户订阅的时事通讯列表的 HTML 窗体。一旦用户选择一个或多个时事通讯并单击【提交】按钮,Web 服务器就会将此信息转发到业务层中相应的 Servlet 或 JSP 组件。业务层组件处理了用户输入后,将进一步与数据存取层交互,由数据存取层将用户的订阅存储到数据库中。

8. Web 应用程序的组件

基于 Java 技术的 Web 应用程序由 Servlet、JSP 页面、图像、HTML 文件、JavaBean、Applet 等组成。如果该 Web 应用程序将被移植到其他服务器或系统上,开发人员则必须将所有这些文件都复制转移到新系统上。一种简便的方法是将所有与 Web 应用程序关联的文件打包成一个 .war 文件,并将该文件部署到新服务器或系统上。与 Servlet 规范兼容的所有 Web 容器都支持 .war 文件。

任务 1.3.2 Java Web 环境搭建

对于 Java Web 应用程序的开发环境,需要安装 JDK、Web 服务器、数据库服务器以

及集成的 Java 开发平台。其中, JDK、Web 服务器是开发所有 Java Web 应用程序必须安装的。

1. JDK 安装

JDK 安装包可以在 <http://java.sun.com> 网站下载, 双击安装文件, 进入安装流程。

(1) 首先进入自定义安装界面, 如图 1.7 所示。



图 1.7 JDK 自定义安装界面

(2) 要在自定义安装界面中修改 JDK 的安装路径, 可单击【更改】按钮更改, 如图 1.8 所示。



图 1.8 更改 JDK 安装路径

(3) 更改后, 先后单击【确定】、【下一步】按钮开始安装, 直至安装完成, 如图 1.9 所示。

(4) JDK 安装完成后, 需要设置两个环境变量(不区分大小写): PATH、CLASSPATH。在桌面上右击【我的电脑】, 在弹出的快捷菜单中选择【属性】选项, 然后选择【高级系统设

置】选项,在弹出的对话框中选择【高级】选项卡,如图 1.10 所示。



图 1.9 JDK 安装完成



图 1.10 【系统属性】对话框中的【高级】选项卡

(5) 单击【环境变量】按钮,打开【环境变量】对话框,如图 1.11 所示。【环境变量】对话框中有用户变量和系统变量,两者的区别是系统变量对所有用户都生效,用户变量只对指定的用户生效。

(6) 在系统变量中选择 Path 变量,单击【编辑】按钮,在变量值中添加“JDK 的安装路径\bin”,其中变量值中原有的值用英文输入法下的分号进行分隔,如图 1.12 所示。

(7) 单击【环境变量】对话框中的【新建】按钮,弹出【新建用户变量】对话框,在变量名中输入 classpath,变量值为“.;%JAVA_HOME%\lib”。其中“.”表示当前目录,分号后面的“%JAVA_HOME%\lib”表示 JDK 安装目录中的 lib 子目录,如图 1.13 所示。



图 1.11 【环境变量】对话框

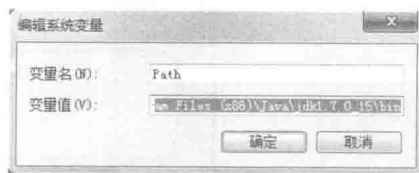


图 1.12 修改 Path 环境变量

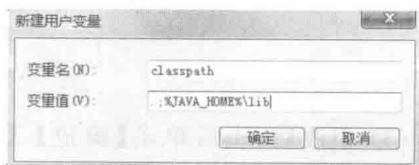


图 1.13 新建 classpath 环境变量

2. Web 服务器安装

1) Web 服务器简介

Web 服务器也称为 WWW 服务器。WWW 采用的是客户机/服务器结构,其作用是整理和储存各种 WWW 资源,并响应客户端软件的请求。

Web 服务器不仅能够存储信息,还能在用户通过 Web 浏览器提供的信息基础上运行脚本和程序。在 Windows 操作系统中,如果采用 ASP 或 ASP.NET 进行 Web 程序开发,则需要通过 IIS 搭建 Web 服务器;如果采用 PHP 进行 Web 程序开发,则需要通过 Apache 搭建 Web 服务器;如果采用 JSP 进行 Web 程序开发,则需要通过 J2SDK 和 J2EESDK 以及相关的应用服务器,如 Tomcat、WebLogic 等搭建 Web 服务器。本书的所有示例都将基于 Tomcat 服务器。

2) Tomcat 的下载与安装

Tomcat 是 Apache 基金会 Jakarta 项目中的一个核心项目,由 Apache、Sun 和其他一些公司及个人共同开发而成。它是一个集成了 Servlet 容器的免费开源 Web 服务器,既能解析 JSP/Servlet,也能提供 Web 服务。

Tomcat 可以在 <http://tomcat.apache.org/> 网站免费下载。本书采用的是 Tomcat 8.0.5 版。直接解压压缩包即可安装 Tomcat,解压的位置可以随意选择,这里解压到 D:\apache-tomcat-8.0.5。

安装完成后,仍需要设置一个环境变量(不区分大小写): java_home,同 JDK 的 classpath 环境变量的配置方法。将变量名设置为 java_home,变量值设为“JDK 的安装路径”,如图 1.14 所示。

打开 D:\apache-tomcat-8.0.5\bin 目录,双击 startup 文件,启动 Tomcat,如图 1.15 所示。



图 1.14 新建 java_home 环境变量

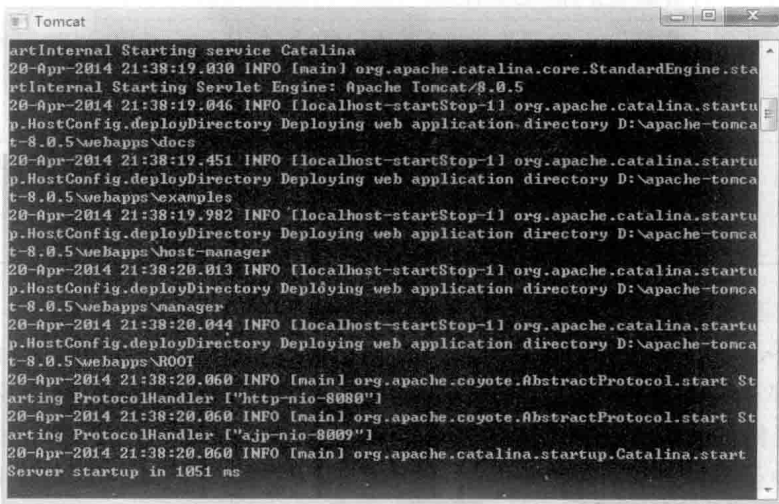


图 1.15 Tomcat 启动窗口

启动浏览器,在地址栏中输入 `http://localhost:8080`,如果出现图 1.16 所示的窗口,则 Tomcat 安装成功。

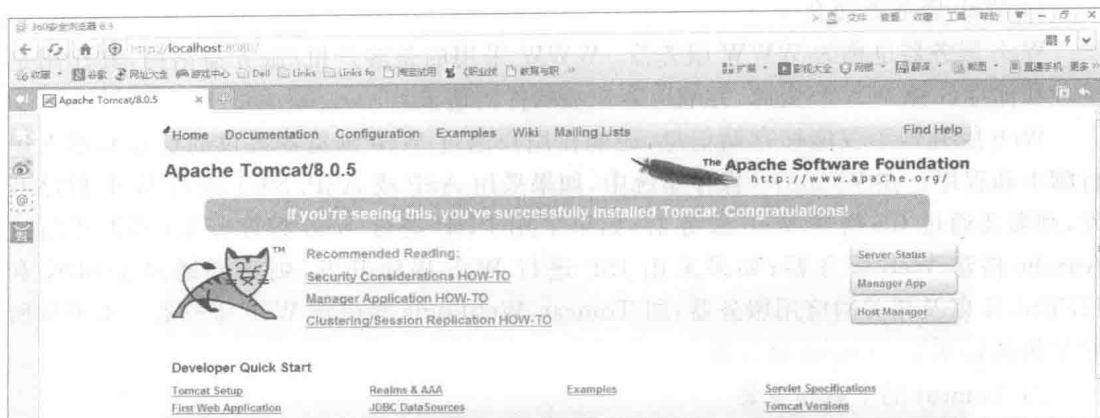


图 1.16 测试 Tomcat

3. Tomcat 的配置

1) Tomcat 的目录结构

编写 Web 应用程序前,首先应了解 Tomcat 的目录结构和作用。Tomcat 的目录结构如表 1.1 所示。

表 1.1 Tomcat 的目录结构

目录名	作用
\bin	存放启动和关闭 Tomcat 服务器的文件
\conf	存放服务器的各种配置文件,包括 server.xml、web.xml
\logs	存放服务器日志文件
\src	存放 Tomcat 服务器相关的源代码,包括 jakarta-servletapi-5、jakarta-tomcat-5、jakarta-tomcat-catalina 等
\temp	存放 Tomcat 服务器的各种临时文件
\webapps	存放 Web 应用程序文件。如 JSP 应用程序、Servlet 应用程序和默认 Web 服务目录 ROOT
\work	存放 JSP 页面所转换成的 Servlet 文件和字节码文件

2) Web 应用程序目录结构

Tomcat 服务器的默认 Web 服务目录是 `\apache-tomcat-8.0.5\webapps\ROOT`。ROOT 即是 Web 应用程序的一个顶层目录,用来标识 Web 应用程序,该顶层目录可以自己命名创建。例如,如果顶层目录的名称为 hello,则此 Web 应用程序可以用 `http://localhost:8080/hello/` 来访问。该顶层目录结构成为文档根目录,由以下几部分组成。

- (1) 静态文件:包括所有的 HTML 网页和图像文件等。
- (2) JSP 页面文件:利用 JSP 页面技术可以很方便地在页面中生成动态内容。

(3) WEB-INF: 该目录存在于 Web 应用程序根目录下。主要由以下部分组成。

① classes 目录: 存储 Servlet 类、JavaBean 类和 Web 应用程序需要的其他类。

② lib 目录: 包含 Web 应用程序所需的各种 .jar 文件。

③ web.xml 文件: Web 应用程序的部署描述文件, 该文件包含有关 Web 应用程序的元数据信息。根元素为 <web-app>。

3) 配置 Tomcat 服务器

默认情况下, Tomcat 服务器的服务端口号是 8080。实际应用中, 有时需要更改这个配置。下面讲述如何修改 Tomcat 服务器端口号。

假设希望将服务器端口号 8080 改为 8000。使用记事本或其他文本编辑器打开 \apache-tomcat-8.0.5\conf\server.xml 文件, 定位到 94 行的 port=8080 处, 将此处修改为 port=8000。修改完毕, 保存该文件, 然后重启 Tomcat 服务器, Tomcat 就使用 8000 端口提供服务了。

4) 示例: 建立简单的 Web 程序并运行

步骤一: 在 Tomcat 下的 webapps 目录下建立 Web 应用程序主目录 hello。

步骤二: 在 hello 目录下建立 WEB-INF 目录和简单的 HTML 文件。代码如下。

```
//index.html
<HTML>
<HEAD>
<TITLE>Hello</TITLE>
</HEAD>
<BODY>
<H1>Hello World!</H1>
</BODY>
</HTML>
```

步骤三: 在 WEB-INF 目录下建立 classes 目录、lib 目录以及 web.xml 文件。在 web.xml 文件中写入根元素 <web-app></web-app>, 代码如下。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
</web-app>
```

步骤四: 启动 Tomcat 服务。

步骤五: 通过 http://localhost:8080/hello/index.html 进行访问。访问结果如图 1.17 所示。

4. 数据库服务器的安装

1) MySQL 简介

本书数据库服务器采用开源的 MySQL 5.0 数据库。MySQL 是一个小型关系型数据库管理系统, 开发者为瑞典的 MySQL AB 公司。MySQL 被广泛应用在 Internet 的中小型网站中。由于其体积小、速度快、总体拥有成本低, 尤其具有开放源码这一特点, 因此