



北京工业职业技术学院
国家示范性高职院校建设项目成果

计算机应用专业

J2EE轻量级应用开发教程

郑 睿 主 编
张建军 姜言芳 唐玉媛 副主编



高等教育出版社
Higher Education Press

国家示范性高职院校建设项目成果

第1版

J2EE 轻量级应用开发教程

J2EE Qingliangji Yingyong Kaifa Jiaocheng

郑 睿 主 编

张建军 姜言芳 唐玉媛 副主编



高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

内容提要

本书是国家示范性高职院校建设项目成果教材。

本书介绍了 J2EE 开发中 Struts、Hibernate、Spring 等目前主流框架的具体使用方法。本书共包括 15 章, 分为三个部分。第一部分讲述 Struts 起源、体系结构、Struts 工作原理、应用程序国际化、表单验证以及 Struts 标记等内容。第二部分讲述 Hibernate 基础知识、实体映射、复杂实体映射、对象关联映射、HQL 数据查询等内容。第三部分讲述 Spring 基础知识、Spring 核心模块、Spring IOC 概念、控制反转、依赖注入、Spring 容器实例化以及 Bean 实例化等 Spring 框架的实用技术。

本书适合作为高职高专院校计算机相关专业的教材, 也可作为初学者的自学用书以及开发人员的参考资料。

图书在版编目(CIP)数据

J2EE 轻量级应用开发教程 / 郑睿主编. —北京: 高等教育出版社, 2010. 4

ISBN 978-7-04-028845-2

I. ①J… II. ①郑… III. ①JAVA 语言—程序设计—高等学校: 技术学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 039053 号

策划编辑	杜冰	责任编辑	许可	封面设计	张志奇
版式设计	王艳红	责任校对	王效珍	责任印制	韩刚

出版发行 高等教育出版社

社址 北京市西城区德外大街 4 号

邮政编码 100120

总机 010-58581000

购书热线 010-58581118

咨询电话 400-810-0598

网 址 <http://www.hep.edu.cn>

<http://www.hep.com.cn>

网上订购 <http://www.landaco.com>

<http://www.landaco.com.cn>

畅想教育 <http://www.widedu.com>

经 销 蓝色畅想图书发行有限公司

印 刷 北京民族印务有限责任公司

开 本 787 × 1092 1/16

印 张 16.75

字 数 400 000

版 次 2010 年 4 月第 1 版

印 次 2010 年 4 月第 1 次印刷

定 价 34.80 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 28845-00

前 言

J2EE(Java 2 Platform Enterprise Edition)是一套全然不同于传统应用开发的技术架构,它包含许多组件,可以简化和规范应用系统的开发和部署,提高可移植性、安全性与重用价值。J2EE的核心是一组技术规范与指南,其包含的各类组件、服务架构及技术层次均有共通的标准及规格,因而各种遵循 J2EE 架构的平台具有良好的兼容性,可以解决企业后端使用的信息产品之间无法兼容、企业内部或外部难以互通的问题。J2EE 不仅继承了标准版中的许多优点,例如“编写一次、随处运行”的特性,方便存取数据库的 JDBC、API、CORBA 技术以及能够在 Internet 应用中保护数据的安全模式等,还提供了对 EJB(Enterprise JavaBeans)、Java Servlet API、JSP(Java Server Pages)以及 XML 技术的全面支持,其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。

轻量级 Java 是相对于一些重量级的容器(如 EJB 容器)来说的,指目前主流的如 Struts、Spring、Hibernate 等框架技术。轻量级 Java 旨在消除与传统 J2EE API 有关的、不必要的复杂性和限制。

本书共 15 章,详细介绍了 Struts、Hibernate、Spring 等目前主流框架的原理及具体使用方法。

第 1 章主要介绍 Struts 起源、架构概念及 Struts 体系结构等知识。

第 2 章主要介绍 Struts 的安装和开发简单的 Struts 应用程序等知识。

第 3 章主要介绍 Struts 工作原理, Model、View、Controller 等核心组件,应用程序国际化,表单验证和错误处理等知识。

第 4 章主要介绍应用程序部署文件和 Struts 配置文件的内容。

第 5 章主要介绍动态表单集动态验证表单、Struts JDBC 连接池等 Struts 高级特性。

第 6 章主要介绍 HTML 标记、Bean 标记、逻辑标记、模板标记等 Struts 标记库的使用方法。

第 7 章主要介绍 Hibernate 基础知识、实体映射基础、复杂实体映射、对象关联映射、HQL 数据查询等内容。

第 8 章主要介绍 Spring 基础知识、Spring 核心模块、简单 Spring 应用等内容。

第 9 章主要介绍 Spring IOC 概念、控制反转、依赖注入等内容。

第 10 章主要介绍 Spring 容器实例化及 Bean 实例化等内容。

第 11 章主要介绍 Spring 对 DAO 的支持、Spring 中使用 JDBC、Spring 中使用 Hibernate 等内容。

第 12 章主要介绍 Spring 对 Hibernate 的支持,整合 Hibernate 和 Spring,SessionFactory 注入,使用 HibernateTemplate、HibernateDaoSupport 等内容。

第 13 章主要介绍 Spring 的事务处理,主要内容有事务概念、JDBC 编程事务处理、Hiber-

nate 编程事务处理等内容。

第 14 章主要介绍 Spring MVC，主要内容有 Spring Web MVC 框架、分发器、WebApplicationContext 等内容。

第 15 章主要介绍 Spring 对 Struts 的支持。

本书内容翔实、结构紧凑、条理清晰、知识点全面、语言通俗易懂，示例丰富，并给出了相应运行效果图。

本书由北京工业职业技术学院的郑睿担任主编，张建军、姜言芳、唐玉媛担任副主编，其中郑睿编写第 1、2、8~15 章，张建军编写第 3~5 章，唐玉媛编写第 6 章，姜言芳编写第 7 章，全书由郑睿负责统稿。由于时间仓促，加之水平所限，书中不足之处在所难免，敬请读者批评指正。

编者

2010 年 2 月

目 录

第 1 章 Struts 概述	1
本章要点	1
1.1 Struts 起源	1
1.2 架构的概念	2
1.3 Struts 体系结构	3
1.3.1 经典 MVC 设计模式	3
1.3.2 JSP 计模式	3
1.3.3 Struts 1.1MVC 体系结构	5
本章小结	6
第 2 章 Struts 1.1 安装和使用	
初步	7
本章要点	7
2.1 Struts 安装	7
2.1.1 需要提前安装的软件	7
2.1.2 通过 Struts 的二进制发布	
包安装 Struts	8
2.1.3 通过源码构造 Struts	8
2.1.4 Struts 的基本组件包简介	8
2.2 开发第一个 Struts 应用程序	9
2.2.1 应用程序功能及 UI 设计	9
2.2.2 实现步骤	10
本章小结	15
第 3 章 Struts 核心机制及组件	16
本章要点	16
3.1 Struts 工作原理	16
3.2 Struts 核心组件	19
3.2.1 Model 组件	19
3.2.2 View 组件	29
3.2.3 Controller 组件	29
3.3 应用程序国际化	38
3.3.1 Resource Bundle	39
3.3.2 Struts 标记使用初步	40
3.3.3 开发第一个 Struts 国际化	
应用程序	41
3.4 表单验证和错误管理	44
3.4.1 ActionError 和 ActionErrors	44
3.4.2 表单验证	45
3.4.3 错误管理	47
本章小结	50
第 4 章 Struts 配置	51
本章要点	51
4.1 应用程序部署文件 web.xml	51
4.2 Struts 配置文件 struts-config.xml	52
本章小结	55
第 5 章 Struts 高级特性	56
本章要点	56
5.1 动态表单及动态验证表单	56
5.1.1 动态表单	56
5.1.2 动态验证表单	57
5.2 Struts JDBC 连接池	65
本章小结	69
第 6 章 Struts 标记库	70
本章要点	70
6.1 HTML 标记	70
6.1.1 基本标记	71
6.1.2 表单及输入组件标记	72
6.1.3 显示错误信息的标记	75
6.2 Bean 标记	75
6.2.1 Bean 定义标记	76
6.2.2 变量定义标记	78
6.2.3 显示 Bean 属性标记	82
6.2.4 消息标记和国际化	82
6.3 逻辑标记	82
6.3.1 条件逻辑标记	82

6.3.2 重复逻辑标记	85
6.3.3 流程控制标记	88
6.4 模板标记	89
本章小结	91
第7章 Hibernate 框架	92
本章要点	92
7.1 概述	92
7.1.1 Hibernate 产生背景	92
7.1.2 Hibernate 的相关概念	94
7.1.3 Hibernate 的下载及设置	94
7.1.4 第一个 Hibernate 示例	95
7.1.5 第二个 Hibernate 示例	101
7.2 实体映射基础	105
7.2.1 Hibernate 基本配置	105
7.2.2 Hibernate 架构	111
7.2.3 Hibernate 核心 API	112
7.2.4 持久化对象的生命周期	118
7.3 复杂实体映射	119
7.3.1 联合主键	120
7.3.2 组合映射	127
7.3.3 Blob/Clob 数据类型的映射 关系	132
7.4 对象关联映射	137
7.4.1 集合类映射	137
7.4.2 关联映射：一对多/多对一	141
7.4.3 关联映射：一对一（外键关联）	148
7.4.4 关联映射：一对一（主键关联）	154
7.4.5 关联映射：多对多	157
7.4.6 继承关系映射	164
7.5 数据查询	171
7.5.1 HQL 基本查询	171
7.5.2 HQL 条件查询	173
7.5.3 其他查询方式	174
本章小结	176
第8章 Spring	177
本章要点	177
8.1 Spring 介绍	177
8.2 Spring 核心模块	177

8.3 Spring 实例	180
本章小结	187
第9章 Spring IOC	188
本章要点	188
9.1 IOC 介绍	188
9.2 控制反转	188
9.3 依赖注入	191
本章小结	192
第10章 Spring 容器	193
本章要点	193
10.1 容器实例化	193
10.1.1 从 classpath 路径下加载 appli- cationContext.xml 文件	193
10.1.2 从文件系统中加载 appli- cationContext.xml 文件	194
10.1.3 从输入流中加载 appli- cationContext.xml 文件	194
10.1.4 基于多配置文件的加载	194
10.2 Bean 实例化	195
10.2.1 使用构造器创建实例	195
10.2.2 使用静态工厂方法创建实例	196
10.2.3 使用实例工厂方法创建实例	197
本章小结	199
第11章 Spring 数据访问	200
本章要点	200
11.1 Spring 对 DAO 的支持	200
11.1.1 DAO	200
11.1.2 一致的异常层次	200
11.1.3 一致的 DAO 抽象类	201
11.2 在 Spring 中使用 JDBC	201
11.2.1 使用 JdbcTemplate	201
11.2.2 JDBC 操作的 Java 对象化	207
11.3 在 Spring 中使用 Hibernate	209
11.4 自增长序列 DataFieldMax- ValueIncrementer	209
本章小结	210
第12章 Spring 对 Hibenate 的支持	211

本章要点	211	13.3 JDBC 声明式事务管理	236
12.1 在 MyEclipse 开发配置 Hibernate 和 Spring	211	13.4 Hibernate 编程事务处理	238
12.2 配置数据浏览视图	211	13.5 Hibernate 声明式事务管理	241
12.2.1 创建新项目	212	13.6 编程事务管理与声明式事务 管理的选择	242
12.2.2 安装 Spring 框架	212	13.7 事务属性介绍	243
12.2.3 安装 Hibernate 3.1 框架	212	本章小结	243
12.3 注入 SessionFactory	218	第 14 章 Spring MVC	244
12.4 使用 HibernateTemplate	225	本章要点	244
12.5 使用 HibernateDaoSupport	227	14.1 Spring Web MVC 框架	244
12.6 使用 HibernateTemplate 进行 分页	228	14.2 分发器	247
本章小结	229	14.3 WebApplicationContext	248
第 13 章 Spring 事务处理	230	14.3.1 使用 ContextLoadListener	248
本章要点	230	14.3.2 使用 ContextLoadServlet	249
13.1 事务概念	230	本章小结	249
13.1.1 编程事务管理	231	第 15 章 Spring 对 Struts 的 支持	250
13.1.2 声明式事务管理	231	本章要点	250
13.2 JDBC 编程事务处理	233	15.1 使用 DelegatingActionProxy 类	250
13.2.1 使用 TransactionTemplate	233	15.2 使用 ActionSupport	254
13.2.2 使用 PlatformTransaction Manager	235	本章小结	256

第 1 章

Struts 概述



【本章要点】

作为学习 Struts 的开始，本章首先介绍一些有关的通用性理论和基本概念，包括应用架构的概念、MVC 模式、JSP 和 Struts 1.1 的 MVC 体系结构。

1.1 Struts 起源

随着软件技术的发展，Web 应用系统越来越复杂。因此，在代码重用、代码移植以及可扩展性等方面面临着诸如重复开发、维护困难等许多问题。Struts 在当前的一些相关标准的开发技术 (JSP、Servlet、XML) 基础之上，提供了一个基于 MVC 模式的开发架构。

作为 Apache 基金会 Jakarta 项目的一个子项目，Struts 的前身是 Craig R. McClanahan 编写的 JSP Model 2 架构中的 Application Framework。Jakarta 项目是 Apache 基金会下属的主要项目之一，其创立的目的是：通过该项目的研究工作，持续改进和提高 JSP、Servlet、标签库以及面向对象的技术水准。除 Struts 之外，Jakarta 项目还推出了包括目前流行的 Tomcat、Ant 等其他的开放源码子项目，开源软件的一大优点是使开发者能更深入地了解其内部实现机制。

Struts 在英文中是“支架，支撑”的意思，这也从侧面表明了 Struts 在开发 Web 应用程序过程中所发挥的重要作用。采用 Struts 可以简化遵循 MVC 设计模式开发 Web 应用程序的工作，使程序员从一些烦琐的工作中解脱出来，快速开发出能够充分发挥 JSP/Servlet 优点并具有强可扩展性的 Web 应用程序。

Struts 1.0 版本于 2001 年 6 月发布，目前其最新的版本为 Struts 1.1。作为 Struts 的主要架构师和开发者，Craig R. McClanahan 曾参与 JSP 规范制定和 Tomcat 4 的开发，同时还领导制定了 J2EE 平台的 Web 层架构的规范。受此影响，Struts 架构一经推出即受到各 Java 开发群体的广泛重视。Borland 公司自 2002 年底开始在其 Java 集成开发工具 JBuilder 中支持 Struts 架构。

1.2 架构的概念

初学者常会出现的一个困惑是：专业书籍和文章中总是使用 Struts Framework 这一称谓，人们说 Struts 是一种 Web 应用架构(Framework)，那么到底什么是架构呢？

架构(也有人翻译为“框架”，为避免译名的不统一导致歧义，本书在后续章节中有时也会直接使用 Framework 一词)的概念很早就有了，经历过面向过程和面向对象的软件开发之后，人们又提出了基于组件的开发技术。在大型，通常是多层结构的软件开发项目中，人们常常开发和使用具有良好的可重用性、可扩展性、经过良好测试的软件组件。这样做的目的是使开发者从烦琐的代码工程中解脱出来，专注于分析设计和业务逻辑的实现。人们逐渐地将被验证为有效的、相同类型问题的解决方案进行抽象，提炼成一个应用程序框架，这就是人们所说的 Framework，例如人们最常接触到的 Microsoft 的 .Net Framework。

多种 Framework 各自有自己的一套明确的实现机制，通常在 Framework 结构中都会包含一个“命令和控制”组件(“command and control” component)，类似于生活中各种机构的“控制中心”，开发人员通过它可以很容易地控制、扩充和实现该 Framework 上的开发结构。

图 1 - 1 给出了一个基于请求 - 响应(Request-Response)模式的应用架构的体系结构。

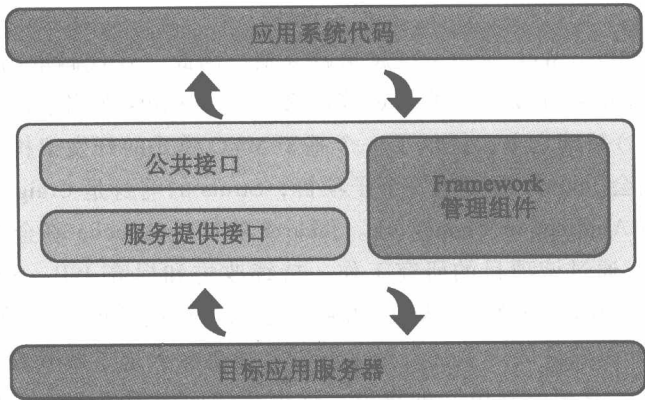


图 1 - 1 Framework 体系结构

从图 1 - 1 中可以看出，Framework 应用服务器通过其控制组件来调用事先配置好的基础组件(公共接口 API)，而开发人员通过事先对这些公共接口进行扩展来实现真正的业务逻辑。

对于初次接触 Framework 的学习者，此时可能很难有深刻的理解，建议在学完第 4 章(那时基本掌握了 Struts Framework)之后再来回顾本节的内容。

1.3 Struts 体系结构

1.3.1 经典 MVC 设计模式

MVC(Model View Controller)模式的核心思想是将程序代码分成相对独立而又能协同工作的三个组成部分。

- ① 模型(Model): 业务逻辑层, 主要用来实现具体的业务逻辑和状态管理。
- ② 视图(View): 显示层, 主要用来实现信息输入和展示功能。
- ③ 控制器(Controller): 控制层, 主要起到控制业务流程控制(Flow Control)的作用, 实现视图和模型部分的协同工作。

MVC 模式起初是针对需要为同样的数据提供多个不同视图的应用程序而设计的, 其结构及各组成部分间的通信方式如图 1-2 所示。

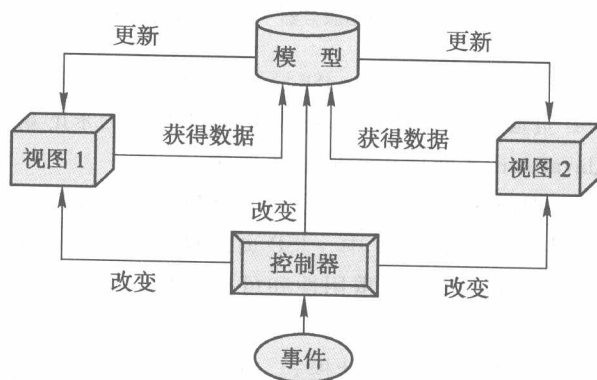


图 1-2 MVC 结构

在这种经典的 MVC 模式中, 事件由控制器处理, 控制器根据事件的类型来改变模型、一个或多个视图, 反之亦然。具体说来, 模型维护一个视图列表, 这些视图为获得模型变化通知已经向模型登记过。当模型发生变化时该模型将通知已向该模型登记的每个视图。视图通常从该视图对应的模型中获得信息, 然后更新自己。

1.3.2 JSP 计模式

在对 MVC 设计模式有了初步认识后, 遵循 MVC 模式开发的应用程序在实现预期的功能外, 其模型部分程序代码将更容易在更大范围内被重用, 设计者只要根据需要配以新的视图和控制器部分, 就得到了一个新的应用程序。

随着 JSP 与 Servlet 技术发展成熟和大量应用基于 Web 的 Java 应用程序, 人们开始尝试在 Java 应用开发中采用 MVC 设计模式, 以使 Java 应用程序具有更好的可维护性和可重用性。AWT、JFC/Swing、JSP 都可用来实现 MVC 结构中视图的部分; JavaBean 和 EJB(Enterprise JavaBean)可对应模型的部分, 即在这部分程序代码中包含具体的业务逻辑; 而 Java Servlet 则兼

具两者的功能，同时还可以作为控制器起到控制程序流程(页面转向)的功能。

早期的 JSP 规格书中曾列举两种可行的 JSP 应用架构，分别为 JSP 设计模式 1(JSP Model1) 与 JSP 设计模式 2(JSP Model2)。下面分别做简单介绍。

1. JSP 设计模式 1

在 JSP 技术的发展初期，由于其便于掌握和可实现快速开发的优点，它很快就成了创建 Web 应用的热门技术。人们在 JSP 页面中可以同时显示业务逻辑和流程控制，从而快速地完成应用开发。最初很多的 Web 应用甚至全部由 JSP 页面构成，这种以 JSP 为中心的开发模型被称之为 Model1。其具体实现方式如图 1-3 所示。

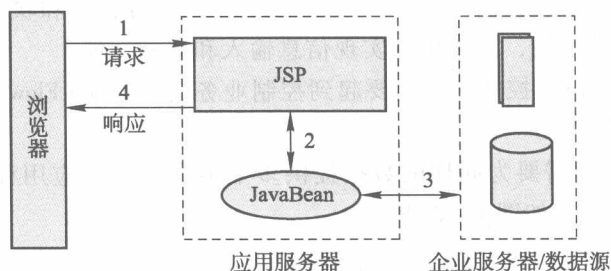


图 1-3 JSP 设计模式 1

在设计模式 1 中，JSP 页面接收处理客户端 Web 浏览器发送来的请求，并在处理后直接作出响应，其间可以辅以 JavaBean 来处理相关的业务逻辑。

设计模式 1 这种模式实现起来比较简单，适合快速开发小规模的项目，但是从工程化的角度看，它的局限性也非常明显。在 Model1 中，JSP 页面身兼 View 和 Controller 两种角色，控制逻辑和表现层逻辑混杂在一起导致代码的重用性和可扩展性下降、维护困难。

2. JSP 设计模式 2

设计模式 2 是基于 MVC 架构的一种设计模式。在设计模式 2 中采用 Servlet 作为前端控制器(Controller)，负责接收客户端 Web 浏览器发送来的所有请求，Servlet 中通常只包含控制逻辑和一些简单的前置处理逻辑(如不同语言字符集的编码转换处理)，再转发到对应的 JSP 页面处理显示逻辑；期间再辅以 JavaBean 实现具体的业务逻辑。其具体实现方式如图 1-4 所示。

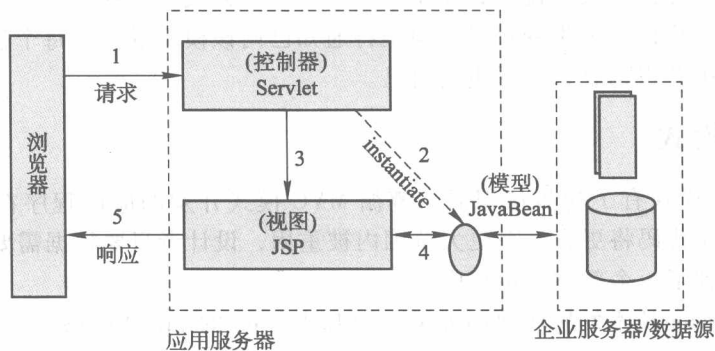


图 1-4 JSP 设计模式 2

由于引入了 MVC 模式，Model 2 具有组件化的优点，更适合用于大规模系统的开发和管

理,但也增加了应用开发的复杂程度。原来建立一个简单的 JSP 页面就能实现的应用现在被分解成多个协同工作的部分,开发者需要花更多时间才能真正掌握其设计和实现过程。

1.3.3 Struts1.1 MVC 体系结构

作为 MVC 模式的一种典型体现,Struts 架构对 Model、View 和 Controller 都提供了对应的实现组件,其实现方式如图 1-5 所示。

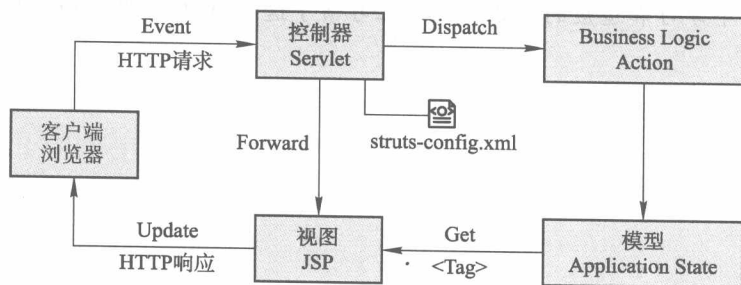


图 1-5 Struts MVC 结构概览

要真正理解和掌握 Struts,必须耐心学完后续章节,这里只是对 Struts 架构的工作原理做一个简要介绍,以使学习者建立一个初步的概念。现在结合图 1-5 分析 Struts 中的 MVC 模式。

1. Controller 部分

Struts 中的 Controller 部分是通过一个专门的 Servlet 来实现的,该 Servlet 是 Struts API 所提供的 ActionServlet 类的一个实例,ActionServlet 类继承了 javax.servlet.http.HttpServlet 类,因此该类的实例可以像普通的 Servlet 一样工作,其作用是接收客户端请求,然后选择执行相应的业务逻辑,再把响应结果送回到客户端。

2. Model 部分

MVC 系统中的 Model 部分从概念上可以分为两类——系统的内部状态和改变系统状态的动作。

Struts 中的 Model 部分由 Action、ActionForm 和 JavaBean 对象组成,其中的 Action 对象是开发者定义的类(该类必须是 Struts API 所提供的 Action 类的子类)的实例。在派生该类的过程中开发者加入具体业务逻辑或调用业务逻辑模块。Action 对象在处理业务逻辑后会将应用程序流程转到合适的 View 组件,最终将响应回送到客户端。ActionForm 对象可以用来自动接收客户端表单数据,这将在第 3 章做详细介绍。Struts 中提倡 Action 对象只用来决定“做什么”,相当于一个低层的控制器,因此也有一种观点将 Action 归纳为 Controller 的组成部分。而使用 JavaBean 来提供具体的业务逻辑,即“怎么做”,对于复杂的系统也可以使用 EJB 等组件来实现系统状态维护。业务逻辑的细分可以增强代码的可重用性。

3. View 部分

Struts 中的 View 部分采用 JSP 来实现。Struts 还提供了丰富的自定义标记库,View 部分使用这些自定义标记可以实现与 Model 部分的有效交互并能增强显示功能。

本章小结

和 JSP 设计模式 2 类似，Struts 架构采用了 MVC 模式，可使 Web 应用具有更加清晰的应用结构，更好的可维护性、重用性和可扩展性，同时也增加了系统的复杂度。对于初学者而言，掌握 Struts 需要一个学习和适应的过程，但一旦习惯了这种 Web 应用架构，你会觉得这些付出是非常值得的。

第 2 章

Struts 1.1 安装和使用初步



【本章要点】

本章主要介绍安装和使用 Struts 前需要做的一些准备工作，并展示了开发 Struts 应用程序的基本步骤。

2.1 Struts 安装

2.1.1 需要提前安装的软件

1. Java 2 SDK SE

Struts 是用 Java 编写的，因此其运行需要 Java 2 SDK SE(简称 JDK)。要求 JDK 1.2 或以上版本，推荐使用 1.4 及以后版本，下载地址：<http://java.sun.com/j2se>。

2. Servlet Container(Web Application Server)

Struts 可以运行在任何一个支持 JSP 1.2 和 Servlet 2.3 的 Web Container 中，推荐使用 Apache Tomcat 4.0 以上版本，下载地址：<http://jakarta.apache.org/tomcat>。

3. Servlet API Classes

编译和运行 Servlet 时需要用到 Servlet 和 JSP API，以将包含了相应 API 的 `servlet.jar` 文件加入到环境变量 `CLASSPATH` 中去。Java EE SDK 和大多数 Servlet Container(例如 Tomcat)中都已自带了这个文件。也可以自己下载，下载地址：<http://jakarta.apache.org/builds>。

4. Ant Build System

如果要通过 Struts 源码发布包来安装 Struts，我们还需要事先下载和安装 Ant 1.1 或更高版本，下载地址：<http://jakarta.apache.org/ant>。

5. XML Parser

Struts 要求一个与 JAXP(Java API for XML Parser)规范相兼容的 XML 解析器，以将 XML 解析库文件(`jaxp.jar`、`parser.jar` 等)加入到环境变量 `CLASSPATH` 中去，下载地址：<http://xml.apache.org/xerces-j>。

6. JDBC 2.0 Optional Package Binary

如果使用数据库，则需要安装 JDBC 2.0 Optional Package Binary，下载地址：<http://java.sun.com/products/jdbc>。

说明：如果使用的是 JDK1.4，那么 XML 解析器和 JDBC 2.0 Optional Package Binary 已自动包含在里面，不必另行下载和安装。

2.1.2 通过 Struts 的二进制发布包安装 Struts

操作步骤如下：

- ① 下载 Struts 的二进制发布版，下载地址：<http://jakarta.apache.org/site/binindex.html>。
- ② 将下载的 Struts 二进制发布版解包。

也可以按照 2.1.3 节中的介绍，通过源码来构造 Struts，实现同样的目的。

2.1.3 通过源码构造 Struts

操作步骤如下：

- ① 下载 Struts 的源码发布包，下载地址：<http://jakarta.apache.org/>。
- ② 设置 ANT_HOME 环境变量，指向 Ant 目录。
- ③ 设置 Java_HOME 环境变量，指向 JDK 目录。
- ④ 将 servlet.jar 文件添加到环境变量 CLASSPATH 中。
- ⑤ 将 Struts 的源码发布包解包
- ⑥ 进入 Struts 目录，运行以下命令：

```
./build.sh dist
```

该命令将创建 Struts 的二进制发布包，目录在 ../dist/struts(相对于编译目录)。

2.1.4 Struts 的基本组件包简介

Struts 1.1 发布版提供的 API 共包括 26 个包(包括子包)、284 个类(包括抽象类和接口)。这里仅对其中几个主要的做简要介绍。

1. org.apache.struts.action

org.apache.struts.action 包是整个 Struts Framework 的核心，该包提供了 Struts MVC 模式中 Controller 的成分，包括上面提到过的控制器类 ActionServlet，以及 Action、ActionForm、ActionMapping、ActionError 等多个非常有用的类。在 Struts 1.1 版本中还新增一个 DynaActionForm 类，可以提供动态扩展生成 FormBean 的功能。

2. org.apache.struts.actions

该包中的 API 提供了作为输入的客户 HTTP 请求和作为响应的业务逻辑处理之间的转换适配器的功能。

3. org.apache.struts.config

该包中定义了一些配置类，这些类的对象是对 Struts 模块配置文件 struts-config.xml 中定义的一些元素的映射，是 Struts 1.1 中新增的功能。

4. org.apache.struts.util

org.apache.struts.util 包提供了多个支持类，如 ErrorMessage，用于解决构建 Web 应用过程中通常会遇到的问题。

5. org.apache.struts.taglib

该包下还含多个子包，分别提供用于构建用户界面的多种订制标记类，包括 Bean Tags、HTML Tags、Logic Tags、Nested Tags、Template Tags 等。

6. org.apache.struts.validator

org.apache.struts.validator 包提供了多个用于验证输入表单数据有效性的类，是 Struts 1.1 中新增的功能。

从 <http://jakarta.apache.org/struts/userGuide/index.html> 可以获得更多关于 Struts 安装和相关配置的说明信息。由于 Struts 是一个正在进行的项目，在未来的新版本中，一些组织和实现方式仍会有所变化，感兴趣的读者可以经常访问 <http://jakarta.apache.org/struts> 以获取最新信息。

2.2 开发第一个 Struts 应用程序

本节将向读者展示一个简单但完整的 Struts 应用程序，对于初学者来说，直接的使用体验将更有帮助。

示例 2-1 开发第一个 Struts 应用程序

具体实现如下所示。

2.2.1 应用程序功能及 UI 设计

这里要开发的是一个简单的身份验证应用模块，用户在登录时必须进行身份验证，系统只允许已经注册的用户进入，该登录页面如图 2-1 所示。

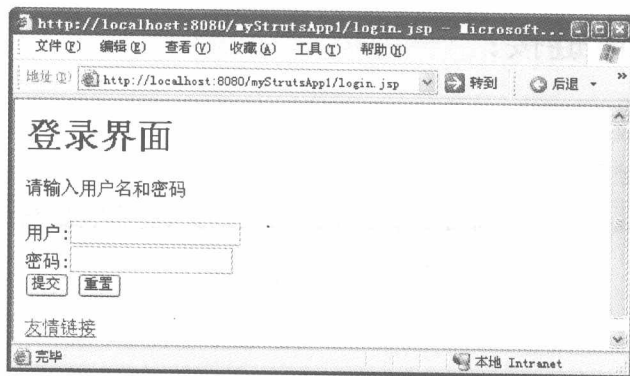


图 2-1 登录页面

在对用户名和密码验证通过后系统将显示登录成功页面，如图 2-2 所示，否则显示登录出错页面，如图 2-3 所示。登录成功或失败后均可选择重新登录，在上述各页面中还添加了一个超链接指向“友情链接”页面，该页面中包含了一些相关的学习网址，页面效果如图 2-4