

在线服务：视频库、源代码库、专业论坛、专家实时支持

# Struts 2<sup>+</sup> Hibernate<sup>+</sup>Spring 整合开发 技术详解

蒲子明 许勇 王黎 等编著



61段全程配音语音教学视频

全书实例源代码，使学习、分析、调试程序更方便

## 在线服务方式

在线服务网站：[www.itzcn.com](http://www.itzcn.com)

QQ群在线服务：45368980、33925615、107423140



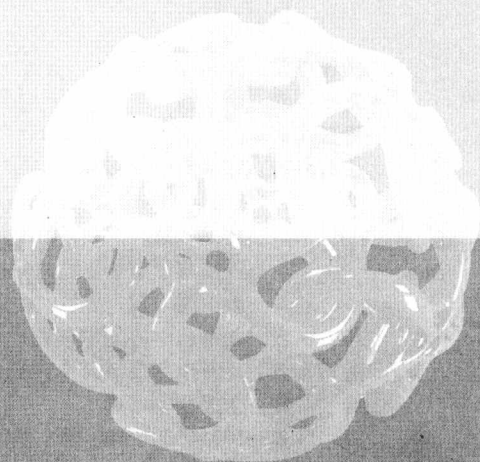
清华大学出版社

在线服务：视频库、源代码库、专业论坛、专家实时支持

# Struts 2<sup>+</sup> Hibernate<sup>+</sup>Spring 整合开发 技术详解

Struts 2<sup>+</sup>  
Hibernate<sup>+</sup>Spring

蒲子明 许勇 王黎 等编著



清华大学出版社

北京

## 内 容 简 介

本书通过理论与实践相结合的方式讲述 Struts 2+Hibernate+Spring 整合开发知识。本书共分 4 篇 19 章,第 1 篇为 1~7 章,内容包括 Struts 2 运行机制,常用配置,拦截器,Struts 2 标签优势及分类,Struts 2 国际化、异常处理、OGNL 和文件上传与下载。第 2 篇为 8~11 章,内容包括 Hibernate 框架、基础配置和 Hibernate 核心接口,基本映射,Hibernate 一级缓存和二缓存及性能优化。第 3 篇为 12~16 章,内容包括 Spring 体系结构,Bean 容器, Spring AOP, Spring MVC 框架, DispatcherServlet 配置、处理器映射与拦截器、视图解析器和控制器, Struts 2 与 Spring 的整合原理、方式和流程。第 4 篇通过 3 个综合应用实例介绍了 Struts 2+ Hibernate + Spring 整合开发知识。

本书适合的读者对象包括 Java Web 开发和 J2EE 开发人员,具备一定基础的 JSP 和 Servlet 开发人员,正在自学 J2EE 知识的读者,本书还可以作为在校师生的教学参考资料。

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

Struts 2+Hibernate+Spring 整合开发技术详解/蒲子明,许勇,王黎等编著. —北京:清华大学出版社,2010.10

ISBN 978-7-302-21103-7

I. S… II. ①蒲… ②许… ③王… III. ①软件工具—程序设计 ②JAVA 语言—程序设计  
IV. TP311.56 TP312

中国版本图书馆 CIP 数据核字(2009)第 168937 号

责任编辑:夏兆彦

责任校对:徐俊伟

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954,jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015,zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京市世界知识印刷厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:190×260 印 张:37.75 字 数:943 千字

附光盘 1 张

版 次:2010 年 10 月第 1 版

印 次:2010 年 10 月第 1 次印刷

印 数:1~5000

定 价:69.50 元

# FOREWORD

## 前言

在众多的软件开发语言中,Java 语言以其简单易学、适用范围广泛等诸多优点,成为近几年最为流行的一种编程语言。尤其是 Web 应用的普及化,更推动了 Java 语言的发展。根据调查结果显示,全国各大 IT 企业,招聘职位最多的就是 Java 开发工程师。对于那些想要加入 IT 队伍的人来说,Java 自然成了首选,越来越多的人选择了学习 Java 语言。

Java 2 Enterprise Edition (J2EE) 技术自从被推出以来就得到了广泛认可和应用,随着多年的技术演变和发展,J2EE 技术平台已经日趋成熟,成为当今电子商务的最佳解决方案。相对于微软推出的 .NET 平台,J2EE 继承了 Java 平台无关性的优点,成为金融、保险、电信等大型应用系统的首选平台方案。

而在 J2EE 技术中,使用 Struts 2+Hibernate+Spring 进行整合开发是最为流行和最受欢迎的框架搭配,正被越来越多的开发者使用,本书就如何将这些框架整合起来应用到 J2EE 开发中去从理论到实践给出了实际的解决方案,引导读者快速进入最流行的 J2EE 开发框架应用实践中去。

### 1. 本书内容

本书共分 4 篇 19 章,通过理论与实践相结合的方式讲述 Struts 2 + Hibernate + Spring 整合开发。

第 1 篇包括第 1~7 章。第 1 章简单介绍 MVC 设计,概述 Struts 2,然后通过一个简单的实例介绍 Struts 2 的使用及运行机制;最后讲述 Struts 2 常用配置。第 2 章讲述 Struts 2 拦截器,包括拦截器原理、配置和自定义拦截器。第 3 章讲述 Struts 2 的 Action 类的实现,讲述 ActionContext 和 Servlet API 的访问、Action 配置、动态方法调用。

第 4 章首先介绍 Struts 2 标签优势及分类,然后以实例形式讲解 Struts 2 的常用标签。第 5 章介绍 Struts 2 输入校验必要性、种类以及与类型的关系,然后以实例形式讲述了两种方式的输入校验:重写 validate() 方法和使用 Struts 2 内置校验器。第 6 章讲述 Struts 2 国际化、异常处理、OGNL 和文件上传与下载。第 7 章以用户注册系统为实例,讲述 Struts 2 应用的使用方法。

第 2 篇包括第 8~11 章。第 8 章通过对象持久化与持久层引出 ORM,概述了 Hibernate 框架,最后讲述了 Hibernate 配置、日志配置和 Hibernate 核心接口。第 9 章讲述 Hibernate 基本映射,介绍集合映射和实体关联关系映射,最后讲述了 Hibernate 常用检索方式: HQL 查询和 QBC 查询。第 10 章介绍事务,概念、并发控制和 Hibernate 悲



观锁与乐观锁, 以及 Hibernate 一级缓存和二级缓存, 最后讲解如何对 Hibernate 进行性能优化。第 11 章通过 BBS 论坛实例来讲述 Struts 2 与 Hibernate 的整合、应用。

第 3 篇包括第 12~16 章。第 12 章主要讲述 Spring 的下载和安装、体系结构、单态与工厂模式的实现、控制反转和与其他产品的比较。第 13 章介绍 Bean 容器、生命周期、基本 Bean 装配、自动装配和 Spring 特殊 Bean 的使用。第 14 章首先概述了 Spring AOP, 然后以实例讲解 5 种通知类型, 最后讲述了切点, 定义与使用, 以及 ProxyFactoryBean 和自动代理。

第 15 章首先概述了 Spring MVC, 通过入门实例来讲解其机制, 介绍了 DispatcherServlet 配置、映射处理器与拦截器、视图解析器和控制器的使用, 最后讲述了中文乱码的处理、信息国际化、文件上传和 Tiles 布局。第 16 章通过相册系统来讲述 Struts 2 与 Spring 的整合原理、方式和流程, 以及如何应用 Struts 2 与 Spring。

第 4 篇包括第 17~19 章。第 17 章通过通讯录实例来讲解 Spring 与 Hibernate 的整合与应用方法。第 18 章通过列车查询系统实例讲述 Struts 2、Spring 与 Hibernate 的整合过程、原理及应用方法。第 19 章通过网上书店项目实例讲述 Struts 2、Spring 与 Hibernate 的整合过程、原理及应用方法。

## 2. 本书特色

本书中采用大量的实例进行讲解, 力求通过实例使读者更形象地理解面向对象的思想, 快速掌握 Struts 2、Spring 和 Hibernate 理论及实际应用。本书难度适中, 内容由浅入深, 实用性强, 覆盖面广, 条理清晰。

- ❑ **示例典型, 应用广泛** 作者精心挑选了大量的示例程序, 它们都是根据作者在实际开发中的经验总结而来的, 涵盖了在实际开发中可能出现的各种问题, 而且有些程序能够直接在项目中使用, 避免读者进行二次开发。
- ❑ **基于理论, 注重实践** 在讲述过程中, 不仅仅只介绍理论知识, 而且在合适位置安排综合应用实例, 或者小型应用程序, 将理论应用到实践当中, 来帮助读者提高实际应用能力, 巩固 Struts 2、Spring 和 Hibernate 开发基础和知识。
- ❑ **语言简洁, 版式活泼** 本书力求以最简洁、活泼的语言和生活中的经典例子来阐述各种知识, 达到易于阅读、理解和掌握的目的。在阐述过程中穿插了各种提示、注意、技巧等体例, 同时能突出重点, 层次分明。
- ❑ **随书配备光盘** 本书为实例配备了视频教学文件, 读者可以通过视频文件更加直观地学习 Struts 2、Spring 和 Hibernate 的操作知识。

## 3. 读者对象

本书具有知识全面、实例精彩、指导性强的特点, 力求以全面的知识性及丰富的实例来指导读者透彻地学习 Struts 2、Spring 和 Hibernate 各方面的知识。本书可以作为 Struts 2、Spring 和 Hibernate 开发的入门书籍, 也可以帮助中级读者提高技能, 对高级读者也有一定的启发意义。

本书适合以下人员阅读学习。

- ❑ Java Web 开发和 J2EE 开发人员。
- ❑ 有一定的 JSP 和 Servlet 基础的读者。
- ❑ 正在接受 J2EE 培训的读者。

- ❑ 在校师生、参加工作的读者以及自学编程的读者。

#### 4. 本书案例开发环境

- ❑ 操作系统 Windows XP。
- ❑ Web 服务器 Tomcat 6.x。
- ❑ 数据库服务器 MySQL 5.0。
- ❑ 开发工具 MyEclipse 7.0。

除了封面署名人员之外，参与本书编写的人员还有于永军、张秋香、李乃文、张仕禹、夏小军、赵振江、李振山、李文才、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林、郭新志、牛丽平、唐守国等。在编写过程中难免会有漏洞，欢迎读者批评指正，以进一步改进和提高本书的质量。

# CONTENTS

## 目 录

### 第 1 篇 Struts 2 篇

|                            |    |
|----------------------------|----|
| 第 1 章 Struts 2 入门          | 2  |
| 1.1 Struts 2 发展历程          | 2  |
| 1.1.1 MVC 概述               | 2  |
| 1.1.2 Struts 1 概述          | 4  |
| 1.1.3 Struts 2 概述及优势       | 6  |
| 1.2 配置 Struts 2 运行环境       | 7  |
| 1.3 Struts 2 第一个例子         | 8  |
| 1.4 Struts 2 各个部分的作用       | 11 |
| 1.5 配置 Struts 2            | 14 |
| 1.5.1 配置 web.xml           | 14 |
| 1.5.2 配置 struts.properties | 15 |
| 1.6 配置 struts.xml          | 18 |
| 1.6.1 文件结构                 | 18 |
| 1.6.2 Bean 配置              | 19 |
| 1.6.3 常量配置                 | 19 |
| 1.6.4 包配置                  | 20 |
| 1.6.5 命名空间配置               | 21 |
| 1.6.6 包含配置                 | 23 |
| 第 2 章 拦截器                  | 25 |
| 2.1 拦截器简介                  | 25 |
| 2.1.1 拦截器工作原理              | 25 |
| 2.1.2 拦截器意义                | 26 |
| 2.1.3 拦截器角色                | 28 |
| 2.2 拦截器配置                  | 29 |
| 2.2.1 配置拦截器                | 29 |
| 2.2.2 使用拦截器                | 32 |
| 2.2.3 默认拦截器                | 34 |
| 2.3 自定义拦截器                 | 35 |
| 2.3.1 实现拦截器类               | 35 |
| 2.3.2 使用自定义拦截器             | 37 |
| 2.3.3 文字过滤拦截器实例            | 40 |
| 2.4 深入拦截器                  | 43 |
| 2.4.1 拦截器方法过滤              | 43 |
| 2.4.2 拦截器的拦截顺序             | 46 |
| 2.4.3 拦截结果监听器              | 47 |

|                                                                |     |                                                 |     |
|----------------------------------------------------------------|-----|-------------------------------------------------|-----|
| 2.4.4 覆盖拦截器中的参数 .....                                          | 49  | 4.5.7 doubleselect 标签 .....                     | 102 |
| 2.5 内建拦截器 .....                                                | 50  | 4.5.8 optgroup 标签 .....                         | 103 |
| 2.5.1 内建拦截器简介 .....                                            | 50  | 4.5.9 datetimetypepicker 标签 .....               | 104 |
| 2.5.2 耗时拦截器实例 .....                                            | 55  | 4.5.10 token 标签 .....                           | 105 |
| <b>第 3 章 Action 和类型转换</b> .....                                | 57  | 4.5.11 updownselect 标签 .....                    | 107 |
| 3.1 实现 Action 控制类 .....                                        | 57  | 4.5.12 optiontransfersselect 标签 .....           | 108 |
| 3.2 访问 ActionContext .....                                     | 60  | 4.6 使用非表单 UI 标签 .....                           | 109 |
| 3.3 直接访问 Servlet API .....                                     | 63  | 4.6.1 actionerror 标签和 actionmessage<br>标签 ..... | 110 |
| 3.4 配置 Action .....                                            | 65  | 4.6.2 component 标签 .....                        | 111 |
| 3.5 动态方法调用 .....                                               | 67  | 4.6.3 tree 标签和 treenode 标签 .....                | 112 |
| 3.6 使用通配符 .....                                                | 68  | <b>第 5 章 Struts 2 输入校验</b> .....                | 114 |
| 3.7 类型转换 .....                                                 | 68  | 5.1 输入校验概述 .....                                | 114 |
| 3.7.1 使用 Struts 2 内建类型转换器 .....                                | 68  | 5.1.1 进行输入校验的必要性 .....                          | 114 |
| 3.7.2 类型转换中的异常处理 .....                                         | 72  | 5.1.2 客户端校验与服务器端校验 .....                        | 115 |
| <b>第 4 章 Struts 2 标签库</b> .....                                | 75  | 5.1.3 类型转换与输入校验的关系 .....                        | 117 |
| 4.1 Struts 2 标签库概述 .....                                       | 75  | 5.2 Struts 2 手动完成输入校验 .....                     | 117 |
| 4.1.1 使用标签的优势 .....                                            | 75  | 5.2.1 重写 validate() 方法 .....                    | 117 |
| 4.1.2 Struts 2 标签库分类 .....                                     | 76  | 5.2.2 重写 validateXxx() 方法 .....                 | 119 |
| 4.2 使用控制标签 .....                                               | 77  | 5.2.3 Struts 2 输入校验流程 .....                     | 122 |
| 4.2.1 if/elseif/else 标签 .....                                  | 77  | 5.3 使用 Struts 2 内置校验器 .....                     | 123 |
| 4.2.2 iterator/append/merge/generator/<br>subset/sort 标签 ..... | 78  | 5.3.1 登录内置校验器实例 .....                           | 123 |
| 4.3 使用数据标签 .....                                               | 84  | 5.3.2 校验器配置风格 .....                             | 125 |
| 4.3.1 action 标签 .....                                          | 84  | 5.3.3 常用内置校验器 .....                             | 126 |
| 4.3.2 bean 标签 .....                                            | 87  | 5.3.4 服务器端校验转换为客户端<br>校验 .....                  | 136 |
| 4.3.3 debug 标签 .....                                           | 88  | 5.4 自定义校验器 .....                                | 137 |
| 4.3.4 include 标签 .....                                         | 88  | <b>第 6 章 Struts 2 扩展与高级技巧</b> .....             | 140 |
| 4.3.5 param 标签 .....                                           | 89  | 6.1 Struts 国际化 .....                            | 140 |
| 4.3.6 property 标签 .....                                        | 90  | 6.1.1 Struts 2 实现国际化的机制 .....                   | 140 |
| 4.3.7 set 标签 .....                                             | 91  | 6.1.2 加载国际化资源文件的方式 .....                        | 144 |
| 4.3.8 url 标签 .....                                             | 92  | 6.1.3 带占位符的国际化消息 .....                          | 145 |
| 4.3.9 date 标签 .....                                            | 93  | 6.1.4 实现自由选择语言环境 .....                          | 147 |
| 4.4 使用主题模板 .....                                               | 94  | 6.2 应用中的异常处理 .....                              | 149 |
| 4.5 使用表单 UI 标签 .....                                           | 96  | 6.2.1 Struts 2 异常处理机制 .....                     | 149 |
| 4.5.1 表单标签的通用属性 .....                                          | 97  | 6.2.2 除法运算异常实例 .....                            | 150 |
| 4.5.2 简单表单标签 .....                                             | 97  | 6.3 OGNL .....                                  | 152 |
| 4.5.3 checkboxlist 标签 .....                                    | 98  | 6.4 文件上传与下载 .....                               | 155 |
| 4.5.4 radio 标签 .....                                           | 100 | 6.4.1 文件上传 .....                                | 155 |
| 4.5.5 combobox 标签 .....                                        | 101 | 6.4.2 文件下载 .....                                | 158 |
| 4.5.6 select 标签 .....                                          | 101 |                                                 |     |



|                 |     |
|-----------------|-----|
| 第 7 章 用户注册系统    | 160 |
| 7.1 系统概述        | 160 |
| 7.1.1 需求分析      | 160 |
| 7.1.2 系统用例图     | 161 |
| 7.1.3 系统设计      | 163 |
| 7.2 数据库设计       | 164 |
| 7.3 通用模块实现      | 165 |
| 7.3.1 实现通用数据库连接 | 165 |
| 7.3.2 配置文件      | 166 |

|              |     |
|--------------|-----|
| 7.3.3 国际化    | 169 |
| 7.4 用户模块实现   | 170 |
| 7.4.1 用户注册   | 170 |
| 7.4.2 用户登录   | 175 |
| 7.4.3 查看所有用户 | 177 |
| 7.4.4 修改个人信息 | 180 |
| 7.5 管理员模块实现  | 184 |
| 7.5.1 管理员登录  | 184 |
| 7.5.2 删除管理员  | 186 |

## 第 2 篇 Hibernate 篇

|                                          |     |
|------------------------------------------|-----|
| 第 8 章 Hibernate 简介                       | 192 |
| 8.1 持久化概述                                | 192 |
| 8.1.1 对象持久化                              | 192 |
| 8.1.2 持久化层                               | 193 |
| 8.2 ORM 简介                               | 195 |
| 8.2.1 ORM 的概念                            | 195 |
| 8.2.2 ORM 面临的问题                          | 197 |
| 8.2.3 ORM 的优点                            | 199 |
| 8.3 Hibernate 框架                         | 200 |
| 8.3.1 Hibernate 的发展                      | 200 |
| 8.3.2 Hibernate、EJB3 和 JPA               | 200 |
| 8.3.3 Hibernate 的下载与安装                   | 202 |
| 8.4 第一个 Hibernate 程序                     | 203 |
| 8.4.1 创建数据库                              | 204 |
| 8.4.2 创建项目                               | 205 |
| 8.4.3 创建持久化类                             | 205 |
| 8.4.4 Hibernate 映射文件                     | 207 |
| 8.4.5 Hibernate 配置文件                     | 208 |
| 8.4.6 Hibernate Session 和<br>MemberDAO 类 | 209 |
| 8.4.7 运行第一个 Hibernate 程序                 | 211 |
| 8.5 Hibernate 基础配置                       | 213 |
| 8.5.1 两种格式配置文件                           | 213 |
| 8.5.2 配置属性                               | 214 |
| 8.5.3 日志配置                               | 218 |
| 8.6 Session 接口                           | 219 |
| 8.6.1 构建 SessionFactory                  | 219 |
| 8.6.2 Session 的创建与关闭                     | 222 |
| 8.6.3 Session 的使用方法                      | 222 |

|                           |     |
|---------------------------|-----|
| 第 9 章 Hibernate 映射与检索     | 227 |
| 9.1 映射文件配置                | 227 |
| 9.1.1 <hibernate-mapping> | 228 |
| 9.1.2 <class>元素           | 229 |
| 9.1.3 <id>元素              | 231 |
| 9.1.4 <property>元素        | 232 |
| 9.2 集合映射                  | 234 |
| 9.2.1 Java 集合类            | 234 |
| 9.2.2 无序 Set 映射           | 237 |
| 9.2.3 有序 Set 映射           | 239 |
| 9.2.4 映射 List             | 241 |
| 9.2.5 映射 Bag              | 242 |
| 9.2.6 映射 Map              | 243 |
| 9.3 实体对象关联关系映射            | 244 |
| 9.3.1 映射单向多对一关联           | 244 |
| 9.3.2 基于外键单向一对一关联         | 248 |
| 9.3.3 基于主键单向一对一关联         | 249 |
| 9.3.4 映射双向一对一关联           | 250 |
| 9.3.5 映射双向一对多关联           | 252 |
| 9.3.6 映射单向多对多关联           | 254 |
| 9.3.7 映射双向多对多关联           | 256 |
| 9.4 Hibernate 检索方式        | 257 |
| 9.4.1 HQL 基础              | 258 |
| 9.4.2 动态查询和动态实例查询         | 262 |
| 9.4.3 分页查询                | 263 |
| 9.4.4 HQL 嵌套子查询           | 264 |
| 9.4.5 多表查询                | 265 |
| 9.4.6 QBC 入门              | 270 |
| 9.4.7 举例查询                | 274 |

## 第 10 章 Hibernate 事务、缓存与性能优化

|                              |     |
|------------------------------|-----|
| 10.1 事务                      | 276 |
| 10.1.1 事务的概念                 | 277 |
| 10.1.2 ACID                  | 278 |
| 10.1.3 在 Hibernate 中使用事务     | 279 |
| 10.2 并发控制                    | 281 |
| 10.2.1 4 个并发问题               | 282 |
| 10.2.2 事务隔离                  | 284 |
| 10.2.3 在 Hibernate 中设置事务隔离级别 | 285 |
| 10.3 悲观锁和乐观锁                 | 285 |
| 10.3.1 Hibernate 锁定模式        | 286 |
| 10.3.2 悲观锁                   | 286 |
| 10.3.3 乐观锁                   | 287 |
| 10.4 Hibernate 缓存            | 291 |
| 10.4.1 缓存的概念                 | 291 |
| 10.4.2 Hibernate 中的两级缓存结构    | 292 |

|                          |     |
|--------------------------|-----|
| 10.4.3 Hibernate 中的第一级缓存 | 293 |
|--------------------------|-----|

|                        |     |
|------------------------|-----|
| 10.5 Hibernate 中的第二级缓存 | 293 |
| 10.6 Hibernate 查询缓存    | 297 |
| 10.7 Hibernate 性能优化    | 299 |
| 10.7.1 优化系统设计          | 300 |
| 10.7.2 批量数据操作优化        | 300 |
| 10.8 Hibernate 查询优化    | 302 |

## 第 11 章 BBS 论坛开发

|                                 |     |
|---------------------------------|-----|
| 11.1 系统设计                       | 307 |
| 11.1.1 需求分析                     | 307 |
| 11.1.2 功能设计                     | 308 |
| 11.2 数据库设计                      | 309 |
| 11.3 系统实现                       | 310 |
| 11.3.1 搭建 Struts 2+Hibernate 环境 | 310 |
| 11.3.2 建立业务实体对象                 | 312 |
| 11.3.3 用户注册模块                   | 314 |
| 11.3.4 论坛帖子模块                   | 320 |
| 11.3.5 管理员模块                    | 325 |
| 11.3.6 安全退出模块                   | 329 |

## 第 3 篇 Spring 篇

### 第 12 章 Spring 概述

|                             |     |
|-----------------------------|-----|
| 12.1 使用 Spring 的原因          | 332 |
| 12.2 Spring 的下载和安装          | 333 |
| 12.3 Spring 快速入门            | 334 |
| 12.3.1 Spring 体系简介          | 334 |
| 12.3.2 Spring 2.5 新特性       | 336 |
| 12.3.3 单态模式回顾               | 337 |
| 12.3.4 工厂模式回顾               | 338 |
| 12.3.5 单态模式与工厂模式的 Spring 实现 | 340 |
| 12.4 理解 IoC                 | 342 |
| 12.4.1 IoC 实施策略             | 342 |
| 12.4.2 依赖注入                 | 344 |
| 12.4.3 设值注入                 | 345 |
| 12.4.4 构造注入                 | 348 |
| 12.5 Spring 产品线横向比较         | 349 |
| 12.5.1 Spring 与 EJB 产品比较    | 349 |
| 12.5.2 与 Spring 相似的框架       | 351 |
| 12.5.3 其他 Web 框架            | 352 |

|               |     |
|---------------|-----|
| 12.5.4 其他持久框架 | 354 |
|---------------|-----|

### 第 13 章 装配 Bean

|                                        |     |
|----------------------------------------|-----|
| 13.1 Bean 容器                           | 357 |
| 13.1.1 Bean 工厂                         | 357 |
| 13.1.2 使用应用程序环境                        | 361 |
| 13.2 Bean 的生命周期                        | 364 |
| 13.2.1 BeanFactory 中 Bean 的生命周期        | 364 |
| 13.2.2 ApplicationContext 中 Bean 的生命周期 | 369 |
| 13.3 基本 Bean 装配                        | 370 |
| 13.3.1 使用 XML 进行装配                     | 371 |
| 13.3.2 添加 Bean                         | 372 |
| 13.3.3 Bean 命名                         | 373 |
| 13.3.4 使用 Setter Injection             | 373 |
| 13.3.5 使用 Constructor Injection        | 375 |
| 13.4 自动装配                              | 376 |
| 13.4.1 自动装配类型                          | 377 |
| 13.4.2 自动装配控制                          | 379 |

|                                       |     |                               |     |
|---------------------------------------|-----|-------------------------------|-----|
| 13.4.3 默认自动装配                         | 381 | 15.1.3 配置 DispatcherServlet   | 428 |
| 13.4.4 使用自动装配的前提                      | 381 | 15.1.4 处理器映射与拦截器              | 430 |
| 13.5 使用 Spring 特殊 Bean                | 382 | 15.1.5 视图解析器                  | 433 |
| 13.5.1 后处理 Bean                       | 382 | 15.1.6 异常解析器                  | 435 |
| 13.5.2 后处理 Bean 工厂                    | 386 | 15.1.7 控制器简介                  | 436 |
| 13.5.3 配置信息分离                         | 386 | 15.1.8 命令控制器                  | 437 |
| 13.5.4 解析文本信息                         | 388 | 15.1.9 表单控制器与验证器              | 439 |
| 13.5.5 事件处理                           | 389 | 15.1.10 多动作控制器                | 444 |
| <b>第 14 章 面向切面编程</b>                  | 392 | 15.1.11 向导控制器                 | 447 |
| 14.1 AOP 简介                           | 392 | 15.1.12 参数映射控制器和文件名<br>映射控制器  | 452 |
| 14.1.1 AOP 术语                         | 392 | 15.2 中文乱码问题                   | 453 |
| 14.1.2 Spring AOP 的实现者                | 394 | 15.3 国际化信息                    | 454 |
| 14.1.3 Spring AOP 实现                  | 395 | 15.4 文件上传                     | 457 |
| 14.2 创建 advice                        | 396 | 15.5 使用 Tiles 布局              | 459 |
| 14.2.1 Before Advice                  | 397 | 15.5.1 定义模板                   | 460 |
| 14.2.2 After Advice                   | 400 | 15.5.2 配置 Tiles               | 461 |
| 14.2.3 Around Advice                  | 401 | <b>第 16 章 相册系统</b>            | 464 |
| 14.2.4 Throws Advice                  | 402 | 16.1 系统概述                     | 464 |
| 14.2.5 Introduction Advice            | 403 | 16.1.1 需求分析                   | 464 |
| 14.3 定义 Pointcut                      | 404 | 16.1.2 系统用例图                  | 465 |
| 14.3.1 定义一个新 Pointcut 的方法             | 404 | 16.1.3 系统设计                   | 466 |
| 14.3.2 理解 Advisor                     | 405 | 16.1.4 数据库设计                  | 466 |
| 14.3.3 使用 Spring 静态 Pointcut          | 406 | 16.2 系统配置                     | 467 |
| 14.3.4 使用 Spring 动态 Pointcut          | 410 | 16.2.1 整合原理                   | 468 |
| 14.4 创建 Introduction                  | 412 | 16.2.2 整合方式                   | 468 |
| 14.4.1 实现 IntroductionInterceptor     | 412 | 16.2.3 整合流程                   | 471 |
| 14.4.2 创建 IntroductionAdvisor         | 414 | 16.2.4 applicationContext.xml | 472 |
| 14.5 使用 ProxyFactoryBean              | 415 | 16.2.5 struts.xml             | 474 |
| 14.6 自动代理                             | 416 | 16.3 系统模块开发                   | 475 |
| 14.6.1 实现类                            | 416 | 16.3.1 用户注册                   | 475 |
| 14.6.2 BeanNameAutoProxyCreator       | 417 | 16.3.2 用户登录                   | 479 |
| 14.6.3 DefaultAdvisorAutoProxyCreator | 418 | 16.3.3 创建相册                   | 481 |
| <b>第 15 章 Spring Web 框架</b>           | 420 | 16.3.4 上传图片                   | 484 |
| 15.1 Spring MVC 框架                    | 420 | 16.3.5 查看相册                   | 488 |
| 15.1.1 Spring MVC 概述                  | 420 | 16.3.6 管理相册                   | 491 |
| 15.1.2 Spring Web 入门实例                | 423 |                               |     |

## 第 4 篇 综合实例篇

|                   |     |              |     |
|-------------------|-----|--------------|-----|
| <b>第 17 章 通讯录</b> | 496 | 17.1.1 需求分析  | 496 |
| 17.1 系统概述         | 496 | 17.1.2 系统用例图 | 497 |

|                          |     |                                               |     |
|--------------------------|-----|-----------------------------------------------|-----|
| 17.1.3 系统设计              | 498 | 18.3.5 系统文件描述                                 | 540 |
| 17.1.4 数据库设计             | 498 | 18.4 Hibernate 持久层                            | 541 |
| 17.2 系统整合                | 499 | 18.4.1 连接数据库                                  | 541 |
| 17.2.1 整合流程              | 499 | 18.4.2 设计持久化对象                                | 543 |
| 17.2.2 配置数据库连接           | 502 | 18.4.3 创建持久化类                                 | 544 |
| 17.2.3 配置 SessionFactory | 502 | 18.4.4 映射持久化类                                 | 546 |
| 17.2.4 在 Spring 中配置 DAO  | 503 | 18.5 实现 DAO 层                                 | 548 |
| 17.3 联系组模块               | 504 | 18.5.1 实现 DAO 组件                              | 549 |
| 17.3.1 添加联系组             | 504 | 18.5.2 部署 DAO 组件                              | 551 |
| 17.3.2 删除联系组             | 508 | 18.6 按车次查询列车模块                                | 552 |
| 17.3.3 查看分组              | 512 | 18.6.1 设计按车次查询列车页面                            | 552 |
| 17.4 联系人模块               | 515 | 18.6.2 创建 Action                              | 553 |
| 17.4.1 添加联系人             | 515 | 18.6.3 配置 Action                              | 554 |
| 17.4.2 修改联系人             | 516 | 18.6.4 JSP 页面输出                               | 554 |
| 17.4.3 全部联系人             | 518 | 第 19 章 网上书店项目                                 | 558 |
| 17.4.4 查询联系人             | 522 | 19.1 系统设计                                     | 558 |
| 第 18 章 列车查询系统            | 524 | 19.1.1 需求分析                                   | 558 |
| 18.1 系统概述                | 524 | 19.1.2 功能设计                                   | 559 |
| 18.1.1 需求分析              | 524 | 19.2 数据库设计                                    | 560 |
| 18.1.2 系统用例图             | 525 | 19.3 系统实现                                     | 563 |
| 18.1.3 系统设计              | 526 | 19.3.1 搭建 Struts 2 + Spring + Hibernate<br>环境 | 563 |
| 18.2 数据库设计               | 527 | 19.3.2 建立业务实体对象                               | 566 |
| 18.3 框架整合                | 529 | 19.3.3 用户注册模块                                 | 568 |
| 18.3.1 整合过程              | 529 | 19.3.4 图书显示模块                                 | 573 |
| 18.3.2 管理员登录             | 535 | 19.3.5 购物车模块                                  | 577 |
| 18.3.3 整合原理              | 538 | 19.3.6 后台管理模块                                 | 585 |
| 18.3.4 系统结构              | 540 |                                               |     |

# 第 1 篇

# Struts 2 篇

第 1 章 Struts 2 入门

第 2 章 拦截器

第 3 章 Action 和类型转换

第 4 章 Struts 2 标签库

第 5 章 Struts 2 输入校验

第 6 章 Struts 2 扩展与高级技巧

第 7 章 用户注册系统



# 第 1 章

## Struts 2 入门



### 内容摘要 | Abstract

Struts 2 最早是 Apache Jakarta 项目的组成部分,项目创立者希望通过对该项目的研究,改进和提高 JSP、Servlet、标签库以及面向对象的技术水准。Struts 采用 MVC 模式,帮助 Java 开发者利用 J2EE 开发 Web 应用。和其他 Java 架构一样,Struts 也是面向对象的设计,充分发挥了 MVC 模式“分离显示逻辑和业务逻辑”的优势。

本章首先介绍 Struts 的发展历程,让读者对 Struts 2 框架的技术发展历程有所了解,接下来介绍 Struts 2,包括与其他框架相比 Struts 2 的优势、Struts 2 组成等基础知识。然后配置 Struts 2 的运行环境,通过一个非常简单的实例向读者介绍使用 Struts 2 进行 Web 开发的基本流程和步骤,最后讲解 Struts 2 各个组成部分的作用,以使读者对 Struts 2 有更清晰的认知。



### 学习目标 | Objective

- 了解 Struts 2 的发展历程
- 熟悉 Struts 2 的项目组成
- 掌握 Action 配置
- 掌握配置 Struts 2 运行环境的方法
- 熟悉 Struts 2 的开发流程
- 掌握 Struts 2 各个部分的作用
- 掌握 Struts 2 的基本配置

## 1.1 Struts 2 发展历程

进入 Struts 2 世界固然令人兴奋,但是 Struts 2 并不是一种完全独立的技术,而是建立在其他 Web 技术之上的一个 MVC 框架,如果脱离了这些技术,Struts 2 框架也就无从谈起。因此,本节首先介绍了 MVC,然后对 Struts 1 和 Struts 2 进行了概述。

### 1.1.1 MVC 概述

MVC 本来存在于桌面程序中,M 是指数据模型,V 是指用户界面,C 则是指控制器。MVC 的目的是将 M 和 V 的实现代码分离,从而使同一个程序可以使用不同的表现形式。例

如一批统计数据，可以分别用柱状图和饼图来表示。C 存在的目的则是确保 M 和 V 的同步，一旦 M 改变，V 应该同步更新，其工作模式如图 1-1 所示。

此种模式的 MVC 又称为经典 MVC，用户与视图（View）进行交互，写入数据并提交，控制器（Controller）接收到来自视图的事件并对模型（Model）进行操作，根据用户输入的数据来更新模型，将更新的结果显示给用户。这种模式有以下两个优点。

- 实现了视图与业务逻辑的分离，提高了代码重用率，减少了数据表达、数据描述和应用操作的耦合度。
- 使得软件可维护性、可重用性、可扩展性、灵活性以及封装性大大提高。

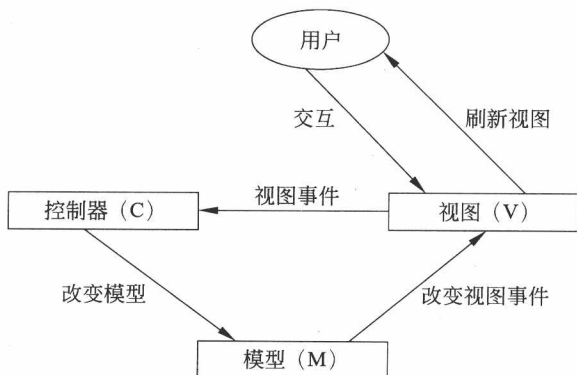


图 1-1 经典 MVC 模式图解



图 1-1 这种经典模式在单机且实时更新的应用程序中可以起到很好的作用。

在 Web 应用中，图 1-1 这种经典的 MVC 模式就不适合了，其缺点主要体现在以下几个方面。

- 因为在 Web 世界中，视图是位于客户端浏览器上的，而控制器和模型则是位于服务器端的，视图无法通过 HTTP 像经典 MVC 那样直接调用控制器。
- 视图不是一个可以更新的对象，而是在客户端发送一个新的请求，随之获得一个全新的 Web 页面。
- 模型不能将自身的改变通知视图，因为视图在另一台计算机的浏览器中。

由于存在以上几点劣势，人们不得不对 MVC 设计模式进行重新设计，因此就诞生了在 Web 应用程序中通过前端控制器来实现 MVC 应用程序的模型。

在这种模型中包含了一个“分发器”（在 Java 的 Web MVC 中分发器使用 Servlet 来实现），分发器可以将一个请求 URL 映射成一个需要被执行的命令，然后去执行相应的命令实例（例如 Struts 2 中的命令实例就是 Action，Action 与后台的服务程序进行交互，这个后台服务程序又叫模型（M））。当命令实例执行完相应的业务逻辑后就会返回一个码值，此码值又会被映射到相应的视图（通常是一个 Web 页面模板，例如 JSP）中。最后结合控制器和模型，映射到的视图将会被发送到客户端的浏览器呈现，此过程的图解如图 1-2 所示。

这种模式对于设计一个系统有很多令人惊喜的好处，如下所示。

- 多个视图可以对应一个模型，这样，可以减少代码的复制及代码的维护量，一旦模型

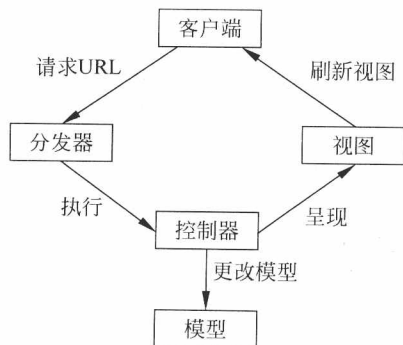


图 1-2 Web 应用中的 MVC 模型

发生改变,也易于维护。

- ☐ 模型返回的数据与显示逻辑分离,这样,模型数据就可以应用任何的显示技术,例如,使用 JSP 页面、Velocity 模板或者直接产生 Excel 文档等。
- ☐ 应用被分隔为层,降低了各层之间的耦合度,提供了应用的可扩展性。
- ☐ 控制层可以发挥更大的作用,由于它把不同的模型和不同的视图组合在一起,可以完成不同的请求,因此,控制层可以包含对用户请求权限的限制。
- ☐ 更符合软件工程化管理的精神。不同的层各司其职,每一层的组件具有相同的特征,有利于通过工程化和工具化管理程序代码。



提示

MVC 的缺点是没有明确的定义,所以完全理解 MVC 并不是很容易。使用 MVC 需要精心地去设计,由于它的内部原理比较复杂,所以需要花费一些时间去思考。

常见 MVC 组件有如下几个。

- ☐ **Struts** 最流行的 MVC 组件, Apache Jakarta 项目组的一个开源项目。
- ☐ **Struts 2** Apache 用 Struts 和 WebWork 组合出来的新产品,目前上升势头强劲。
- ☐ **WebWork** 这是老牌的 MVC 组件,后来组合成了 Struts 2。
- ☐ **Spring MVC** Spring Framework 自己整合自己 Spring 的优势推出的 MVC 组件。
- ☐ **JSF** 这是一个规范。用户量很大,大多数 IDE 都支持。

## 1.1.2 Struts 1 概述

在过去, Struts 1 是所有 MVC 框架中不容辩驳的胜利者,不管是市场占有率,还是所拥有的开发人群, Struts 1 都拥有其他 MVC 框架不可比拟的优势。Struts 1 的成功得益于它丰富的文档和活跃的开发群体,当然, Struts 1 是 Web 应用的第一个 MVC 框架,这一点也是它得到如此广泛拥戴的一个重要原因。

对于整个 Struts 1 框架而言,控制器就是它的核心(大部分 Web MVC 框架也都是如此), Struts 1 的控制器由两个部分组成:核心控制器和业务逻辑控制器。其中核心控制器就是 ActionServlet,由 Struts 1 框架提供;业务逻辑控制就是用户自定义的 Action,由应用开发者提供。



提示

当用户发送一个需要得到服务器处理的请求时,该请求被 ActionServlet 拦截到, ActionServlet 将该请求转发给对应的业务逻辑控制器,业务逻辑控制器调用模型来处理用户请求;如果用户请求只是希望得到某个 URL 资源,则由 ActionServlet 将被请求的资源转发给用户。

Struts 1 是一个 Web 的 MVC 框架(后面如果再提到 MVC 就默认是指 Web 的 MVC 模型),那么它的“MVC 三角色”又分别是什么呢?

### 1. Model 部分

Struts 1 的 Model 部分主要由底层的业务逻辑组件充当,这些业务逻辑组件封装了底层数

数据库访问、业务逻辑方法实现。实际上, 对于一个成熟的企业应用而言, Model 部分也不是一个简单的 JavaBean 所能完成的, 它可能是一个或多个 EJB 组件, 也可能是一个 WebService 服务。总之, Model 部分封装了整个应用的所有业务逻辑, 但整个部分并不是由 Struts 1 提供的, Struts 1 也没有为实现 Model 组件提供任何支持。

## 2. View 部分

Struts 1 的 View 部分采用 JSP 实现。Struts 1 提供了丰富的标签库, 通过这些标签库可以最大限度地减少脚本的使用。这些自定义的标签库可以输出控制器的处理结果。

虽然 Struts 1 提供了与 Tiles 框架的整合, 但 Struts 1 所支持的表现层技术非常单一: 既不支持 FreeMarker、Velocity 等模板技术, 也不支持 JasperReports 等报表技术。

## 3. Controller 部分

Struts 1 的 Controller 由两个部分组成: 核心控制器和业务逻辑控制器。

### □ 核心控制器

由 Struts 1 框架提供, 就是系统中的 ActionServlet。它继承自 HttpServlet 类, 因此可以配置成一个标准的 Servlet, 该控制器负责拦截所有 HTTP 请求, 然后根据用户请求决定是否调用业务逻辑控制器, 如果需要调用业务逻辑控制器, 则将请求转发给 Action 处理, 否则直接转向请求的 JSP 页面。

### □ 业务逻辑控制器

不是由 Struts 1 框架提供的, 而是用户自己实现的 Action 实例。业务逻辑控制器负责处理用户请求, 但业务逻辑控制器本身并不具有处理能力, 而是调用 Model 来完成处理。

Struts 1 不但提供了系统所需要的核心控制器, 也为实现业务逻辑控制器提供了许多支持。因此, 控制器部分就是 Struts 1 框架的核心。对于任何的 MVC 框架而言, 其实只实现了 C (控制器) 部分, 但它负责用控制器调用业务逻辑组件, 并负责控制器与视图技术 (JSP、FreeMarker 和 Velocity 等) 的整合。

当客户端向服务器发送请求时, 请求首先被 Struts 1 的核心控制器 ActionServlet 拦截, ActionServlet 根据请求决定需要调用哪个业务逻辑控制器 Action 来处理客户端请求。Action 本身并没有处理能力, 只是进行业务逻辑控制, 它还需要调用响应的模型来完成处理。当处理完客户端请求后 Action 就将处理结果通过 JSP 页面呈现给用户, 如图 1-3 所示。

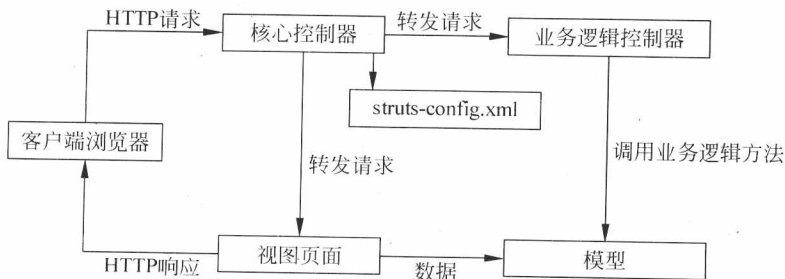


图 1-3 Struts 1 的程序运行流程